

SMAC03 - GRAFOS

COLETAS DE DADOS IMOBILIÁRIOS

2023005290 - Amon Lemes dos Santos

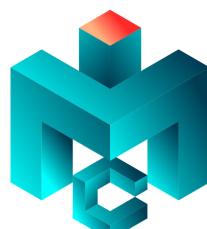
2023006878 - Lucas Salles Granado

2023004480 - Sammuel Gonçalves Reis

2024001849 - Ulisses Douglas de Paula Assis Junior

SMAC03 - GRAFOS

Prof. Rafael Frinhani



INSTITUTO DE
MATEMÁTICA E
COMPUTAÇÃO

UNIFEI - Itajubá



Coletas de Dados Imobiliários

1 Introdução

O Recadastramento Imobiliário Multifinalitário é um processo importante para atualizar informações geográficas e cadastrais sobre os imóveis de um município. Esses dados são utilizados em áreas como planejamento urbano, regularização fundiária, cálculo de tributos e apoio à gestão pública. Uma parte relevante desse processo envolve a coleta em campo, em que agentes percorrem quarteirões registrando características dos imóveis. Como essa atividade exige deslocamento físico, a eficiência das rotas influencia diretamente o tempo e o custo total da coleta.

Para estudar esse problema, utilizamos um recorte do mapa da cidade de Elói Mendes, obtido por meio da plataforma [OpenStreetMap \(2025\)](#). A partir desses dados criamos um dataset representando a malha viária da região. As vias foram convertidas em um grafo simples: cada cruzamento foi modelado como um vértice e cada rua como uma aresta. Inicialmente, os pesos das arestas representavam apenas a distância, mas o modelo foi ampliado para incluir também a quantidade de casas presentes em cada quarteirão, permitindo estimar melhor o esforço necessário para percorrer cada rua.

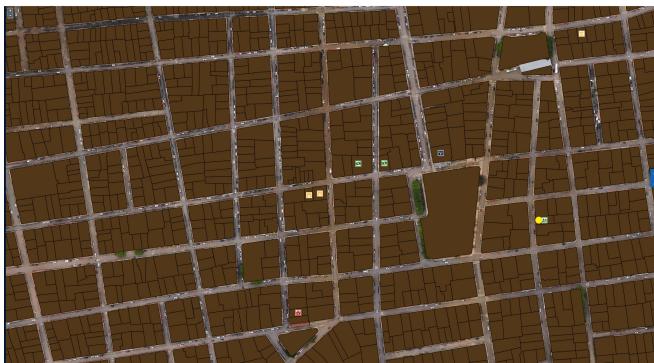


Figura 1: Lotes do recorte do Mapa de Elói Mendes

Com o grafo definido, aplicamos algoritmos da Teoria dos Grafos para encontrar rotas eficientes. O problema se aproxima do *Chinese Postman Problem*, conforme descrito em [Edmonds & Johnson \(1973\)](#), no qual se busca uma rota de menor custo que percorra todas as arestas ao menos uma vez. Também foram considerados algoritmos de caminho mínimo, como Dijkstra, e abordagens para dividir o grafo quando existe mais de um agente. Isso possibilitou avaliar tanto o caso em que um único agente realiza toda a coleta quanto cenários em que o trabalho é distribuído entre vários agentes.

O objetivo deste projeto é propor e analisar soluções de roteamento baseadas em grafos que possam apoiar o planejamento da coleta em campo. Para isso, estudamos diferentes

formas de modelar o peso das arestas, exploramos variações de algoritmos e desenvolvemos visualizações que facilitam a interpretação das rotas geradas. Os resultados obtidos permitem comparar abordagens e entender como diferentes escolhas de modelagem influenciam a eficiência final do processo.

1.1 Divisão de Tarefas entre os Integrantes

O desenvolvimento do projeto contou com contribuições complementares dos integrantes, que atuaram em diferentes frentes do sistema de coleta e otimização de rotas.

Lucas Salles Granado trabalhou tanto na solução para um agente quanto nas extensões para múltiplos agentes. Além de estruturar o dataset inicial, aplicar o algoritmo de Edmonds-Johnson e desenvolver visualizações estáticas e animadas, ele também implementou uma estratégia baseada em *route-first, split-second*, gerando uma rota global para então dividi-la entre vários agentes. Lucas também contribuiu de forma significativa para a criação dos materiais de apresentação.

Ulisses Douglas de Paula Assis Junior contribuiu na organização geral do projeto e na montagem do script principal, que integra e automatiza a execução das diferentes etapas do pipeline. Ele também apoiou melhorias estruturais no repositório e participou do andamento geral da implementação.

Amon Lemes dos Santos participou ativamente do desenvolvimento técnico e da consolidação do relatório em [L^AT_EX](#). Ele explorou alternativas de clusterização e rotas multiagente que, embora não tenham sido adotadas na solução final, foram importantes para ampliar o entendimento do grupo sobre diferentes abordagens possíveis. Amon também auxiliou na construção das visualizações usadas para interpretar o comportamento das rotas.

Sammuel Gonçalves Reis foi responsável pela elaboração e organização do relatório em [L^AT_EX](#), além de integrar partes distintas do pipeline ao longo do desenvolvimento. Também realizou testes manuais para validar o comportamento dos algoritmos e ajustou o dataset final, incluindo a contagem precisa do número de casas por quarteirão, incorporada manualmente após análise do mapa. Sammuel também está auxiliando na finalização dos materiais de apresentação.

Em conjunto, as contribuições dos integrantes permitiram explorar múltiplas abordagens, validar cenários diversos, aprimorar o modelo de pesos das arestas e estruturar uma solução completa para o roteamento dos agentes.

2 Referencial Teórico

Para o referencial teórico deste artigo, buscou-se os principais conceitos necessários para compreender o problema estudado,

além de trabalhos relacionados que tratam de roteamento em malhas urbanas, algoritmos clássicos da teoria dos grafos e técnicas de particionamento.

2.1 Modelagem de Redes Urbanas como Grafos

A representação de cidades como grafos é amplamente utilizada em tarefas de análise urbana, mobilidade e logística. Interseções são modeladas como vértices e trechos de rua como arestas com pesos associados a atributos relevantes. Trabalhos como [Boeing \(2017\)](#) descrevem métodos consolidados para extrair dados da plataforma OpenStreetMap e convertê-los em grafos adequados para análise computacional.

2.2 Chinese Postman Problem

O problema do carteiro chinês foi formulado por Kwan em 1962 e aprofundado por Edmonds e Johnson em [Edmonds & Johnson \(1973\)](#). O objetivo é encontrar a rota de menor custo que percorre todas as arestas de um grafo ao menos uma vez. Quando o grafo apresenta vértices de grau ímpar, é necessário:

- identificar os vértices ímpares;
- calcular caminhos mínimos entre eles usando Dijkstra;
- realizar um emparelhamento de custo mínimo;
- duplicar as arestas correspondentes;
- gerar um circuito euleriano no grafo resultante.

Esse método é utilizado em aplicações como inspeção de vias, coleta porta a porta e monitoramento urbano.

2.3 Caminhos Mínimos

O algoritmo de Dijkstra é a ferramenta fundamental quando se trata de caminhos mínimos em grafos ponderados com pesos não negativos. No contexto deste trabalho, ele é utilizado para:

- calcular distâncias para o emparelhamento do CPP;
- construir subgrafos conectados nos métodos multiagente;
- medir custos entre vértices durante o particionamento.

2.4 Algoritmos Eulerianos

Após a eulerização do grafo, a rota final é obtida por algoritmos clássicos como o de Hierholzer, que encontra circuitos eulerianos de forma eficiente. O algoritmo de Fleury também é tradicionalmente associado ao tema, embora seja menos eficiente. Ambos fazem parte do arcabouço teórico que fundamenta a construção das rotas.

2.5 Clusterização e Divisão do Trabalho entre Agentes

Com múltiplos agentes, o problema passa a se aproximar de versões multiagente do CPP. Estratégias comuns incluem dividir o grafo em regiões menores e resolver o CPP individualmente em cada subgrafo, como discutido por Corberán et al em [Corberán et al. \(2021\)](#). A atribuição de vértices pode ser realizada por heurísticas gulosas baseadas em distância, buscando equilibrar o custo total por agente.

Um ponto importante utilizado neste projeto é o cálculo do nó inicial por meio de uma estimativa da mediana geográfica dos vértices, também chamada de *geodesic median*. Esse nó serve como origem comum às rotas dos agentes e minimiza a soma das distâncias até os demais vértices, sendo um critério mais robusto do que a média aritmética das coordenadas.

2.6 Particionamento de Grafos

Além das heurísticas de clusterização, técnicas clássicas de particionamento de grafos foram empregadas. O algoritmo de Kernighan–Lin [Kernighan & Lin \(1970\)](#) é um dos métodos mais conhecidos para particionar um grafo em dois subconjuntos com corte mínimo. No contexto deste trabalho, versões recursivas do algoritmo foram utilizadas para produzir divisões em mais de duas partes, possibilitando a criação de regiões equilibradas para múltiplos agentes.

Essas abordagens complementam as heurísticas espaciais e contribuem para uma divisão mais estruturada da malha urbana.

2.7 Trabalhos Relacionados

A literatura apresenta diversos estudos sobre problemas de roteamento urbano. Revisões como [Corberán et al. \(2021\)](#) discutem desafios em problemas baseados em arestas, incluindo inspeção e cobertura de redes urbanas. Trabalhos como [Bentley \(1975\)](#) abordam estruturas de dados úteis para processar coordenadas e mapas, enquanto [Boeing \(2017\)](#) destaca o papel do OpenStreetMap na extração de grafos urbanos.

Esses estudos ajudam a situar o projeto dentro de um contexto mais amplo, reforçando a relevância da combinação entre dados geográficos, modelagem por grafos e algoritmos de roteamento.

3 Desenvolvimento

3.1 Obtenção e Pré-processamento dos Dados

Os dados foram extraídos da plataforma OpenStreetMap em formato *.pbf*. A primeira etapa consistiu na identificação de vértices correspondentes a interseções e arestas correspondentes às ruas. O script *extrair_vertices_pbf.py* processou o arquivo bruto, aplicou uma clusterização espacial de tolerância métrica para unir pontos muito próximos e converteu coordenadas de WGS84 para UTM, permitindo cálculos de distância em metros. A distância entre pontos foi obtida com a fórmula de Haversine.

Também foram processados atributos adicionais, como a estimativa do número de casas por aresta. Esse valor foi calculado pela razão entre o comprimento da rua e um espaçamento

médio entre residências, gerado pelo script `calc_n_casas_-csv_arestas.py`. Isso permitiu enriquecer o peso das arestas com informações mais próximas do esforço real da coleta imobiliária.

3.2 Construção da Matriz de Adjacência e Atribuição de Pesos

A partir dos CSVs processados, o script `gerar_matriz_-adjacencia.py` reconstruiu a matriz de adjacência contendo apenas as arestas válidas. Durante esse processo foram preservados atributos como distância real e número estimado de casas. A matriz resultante representa um grafo ponderado no qual cada vértice é um cruzamento e cada aresta contém informações completas sobre o trecho viário correspondente.

Um passo adicional consistiu no cálculo do vértice inicial comum aos agentes. Para isso foi utilizada uma aproximação da mediana geográfica (*geodesic median*), escolhendo o vértice cuja soma das distâncias aos demais é mínima. Esse critério é mais robusto do que uma simples média aritmética das coordenadas e contribui para uma distribuição mais equilibrada das rotas. Após a obtenção dos arquivos CSV foi possível a geração de arquivos HTML que geram o grafo no em uma imagem cartográfica real a partir do arquivo `visualizar_-mapa_interativo.py`.



Figura 2: Grafo gerado sobre as imagens cartográficas do município de Elói Mendes

3.3 Solução para Um Agente com Edmonds-Johnson

A solução para um único agente, implementada em `resolver_cpp.py`, aplica o algoritmo de Edmonds-Johnson. O fluxo central da solução inclui:

- identificação dos vértices de grau ímpar;
- cálculo de caminhos mínimos entre cada par de vértices ímpares usando Dijkstra;
- construção de um grafo completo ponderado;
- emparelhamento perfeito mínimo;
- duplicação das arestas selecionadas;
- geração do circuito euleriano por meio de Hierholzer.

Esse método produz uma rota única que percorre todas as ruas ao menor custo possível. Foram desenvolvidas visualizações estáticas e animações que mostram o percurso exato do agente.

Algoritmo 1: Edmonds-Johnson

Dados: Grafo conectado $G = (N, E)$ com pesos não negativos c_e nas arestas

Resultado: Circuito de menor custo que percorre cada aresta de G ao menos uma vez

```

1 /* Passo 1 - Identificação dos vértices de
   grau ímpar */
2  $O \leftarrow []$ ; // conjunto de nós de grau ímpar
3 para cada nó  $v \in N$  faça
4   calcular  $\deg(v)$ ;
5   se  $\deg(v)$  é ímpar então
6     adicionar  $v$  a  $O$ ;
7   fim
8 fim
9 se  $|O| = 0$  então
10  /*  $G$  já é Euleriano */           */
11  ir para Passo 5;
12 fim
13 /* Passo 2 - Cálculo de caminhos mínimos
   entre pares de vértices ímpares */
14 construir  $G_p$  como grafo completo com vértices  $O$ ;
15 para cada pares distintos  $(i, j)$  com  $i, j \in O$  faça
16   executar Dijkstra em  $G$  de  $i$  até  $j$ ;
17    $dist(i, j) \leftarrow$  custo do caminho mínimo encontrado;
18   atribuir peso  $w_{ij} \leftarrow dist(i, j)$  à aresta  $(i, j)$  em  $G_p$ ;
19   armazenar também a sequência de arestas do
   caminho mínimo correspondente;
20 fim
21 /* Passo 3 - Emparelhamento perfeito de
   custo mínimo em  $G_p$  */
22 encontrar o emparelhamento perfeito de custo
mínimo  $M$  em  $G_p$ ;
23 ; /* (utilizar algoritmo polinomial
   apropriado para matching mínimo) */
24 /* Passo 4 - Eulerização do grafo original
   */
25  $G' \leftarrow G$ ; // iniciar  $G'$  como cópia de  $G$ 
26 para cada aresta  $(i, j) \in M$  faça
27   recuperar o caminho mínimo correspondente em  $G$ 
   entre  $i$  e  $j$ ;
28   para cada aresta  $e$  desse caminho faça
29     duplicar  $e$  em  $G'$  (adicionar cópia da aresta);
30   fim
31 fim
32 /* Agora  $G'$  é um multigrafo Euleriano */
33 /* Passo 5 - Geração do circuito Euleriano
   de custo mínimo */
34 aplicar o algoritmo de Hierholzer em  $G'$  para extrair
   um circuito Euleriano  $C$ ;
35 retorna  $C$ ;
```

3.4 Clusterização para Múltiplos Agentes

A primeira estratégia utilizada para dividir o grafo entre múltiplos agentes foi baseada em uma heurística gulosa. A partir do nó central calculado, aplica-se Dijkstra para obter o custo de cada vértice e então distribui-se os vértices entre os agentes buscando equilibrar o acúmulo de custo.

Após a atribuição inicial, foram construídos subgrafos co-

nectados para cada grupo de vértices, utilizando caminhos mínimos sempre que necessário. Essa etapa garantiu que cada agente recebesse uma região contínua adequada para aplicação do CPP individual.

3.5 Particionamento Recursivo com Kernighan–Lin

Outra abordagem empregada para a divisão do grafo foi o particionamento estrutural, considerou-se a implementação uma versão recursiva do algoritmo de Kernighan–Lin para gerar partições balanceadas. O método funciona da seguinte forma:

- aplica-se *Kernighan–Lin bisection* para dividir o grafo em duas partes;
- cada uma das partes é subdividida novamente até atingir o número desejado de agentes;
- verificam-se potenciais desconexões e elas são corrigidas pela adição de caminhos mínimos.

Esse método proporciona divisões mais regulares e fundamentadas na estrutura global do grafo, em contraste com a heurística gulosa baseada em distância. Temos no Algoritmo 2, a implementação do algoritmo de Kernighan–Lin pra divisão do grafos em duas partes.

Algoritmo 2: Kernighan–Lin

Dados: G, S , partições iniciais A e B com $|A| = |B| = n$

```

1 enquanto verdadeiro faça
2   /* Inicialização do Passe */ *
3   Ganho_Maximo ← 0;
4   Melhor_K ← 0;
5   A_livres ← A; B_livres ← B;
6   Pares_Trocados ← [ ];
7   Ganhos_Parciais ← [ ];
8   /* Cálculo dos Custos Iniciais  $D_z$  */
9   nó  $z \in S$  Calcular  $E_z$  (custo externo);
10  Calcular  $I_z$  (custo interno);
11   $D_z \leftarrow E_z - I_z$ ;
12  /* Loop de Trocas Sequenciais */
13  para  $p \leftarrow 1$  até  $n$  faça
14    /* Selecionar par ótimo  $(a_i, b_j)$  */
15    Escolher  $(a_i, b_j)$  com  $a_i \in A_{\text{livres}}, b_j \in B_{\text{livres}}$ 
       que maximize;
16     $g_p = D_{a_i} + D_{b_j} - 2c_{a_i b_j}$ ;
17    /* Registrar e bloquear */
18    Pares_Trocados.push( $a_i, b_j$ );
19    Ganhos_Parciais.push( $g_p$ );
20    Remover  $a_i$  de  $A_{\text{livres}}$ ;
21    Remover  $b_j$  de  $B_{\text{livres}}$ ;
22    /* Atualização dos custos  $D_z$  */
23     $x \in A_{\text{livres}} D_x \leftarrow D_x + 2c_{xa_i} - 2c_{xb_j}$ ;
24     $y \in B_{\text{livres}} D_y \leftarrow D_y + 2c_{yb_j} - 2c_{ya_i}$ ;
25  fim
26  /* Determinação do ganho acumulado
     máximo */
27  Calcular  $G_k = \sum_{p=1}^k g_p$  para cada  $k$ ;
28  Encontrar  $k^*$  que maximiza  $G_k$ ;
29  Ganho_Maximo ←  $G_{k^*}$ ;
30  /* Decisão */
31  se  $Ganho\_Maximo > 0$  então
32    X ← primeiros  $k^*$  elementos de A em
      Pares_Trocados;
33    Y ← primeiros  $k^*$  elementos de B em
      Pares_Trocados;
34    A ← (A X) ∪ Y;
35    B ← (B Y) ∪ X;
36    continue para o próximo passe;
37  fim
38  senão
39    break; /* Partição localmente ótima
      (Phase 1 optimal) */
40  fim
41 fim
42 retorna  $A, B$ ;
```

3.6 Eulerização e Construção das Rotas

Para cada subgrafo obtido pelas estratégias anteriores, aplicou-se um processo de eulerização aprimorado, também presente em `depacho_rotas_k_agentes.py`. Esse processo consiste em:

- identificar vértices ímpares do subgrafo;

- calcular caminhos mínimos reais entre esses vértices, preservando atributos das arestas;
- construir um grafo completo ponderado;
- gerar um emparelhamento mínimo entre vértices ímpares;
- duplicar os caminhos selecionados no grafo original;
- construir o multigrafo euleriano final.

A trilha euleriana é obtida pela função *eulerian_circuit* do NetworkX. Em seguida ela é ajustada para iniciar no vértice central calculado anteriormente. Caso o subgrafo não conteña esse vértice, é inserida uma trilha prévia que conecta o vértice mais próximo ao ponto inicial comum. Esse ajuste garante compatibilidade com o requisito do trabalho de que todos os agentes começem do mesmo ponto.

3.7 Geração e Divisão das Rotas entre Agentes

Após a construção do circuito euleriano de cada subgrafo, o percurso completo é dividido proporcionalmente entre os agentes. Em abordagens alternativas, também foi gerado um único circuito euleriano global que depois é segmentado em blocos iguais, permitindo comparar diferentes estratégias de distribuição da carga.

Essa diversidade de métodos possibilitou testar vários cenários e avaliar como características estruturais do grafo influenciam na qualidade da divisão das rotas.

3.8 Visualização das Rotas

Foi implementado um conjunto de ferramentas de visualização para auxiliar na interpretação dos resultados. Entre elas estão:

- mapas interativos produzidos com Folium;
- animações que exibem a ordem de cobertura das arestas;
- mapas estáticos estilizados que destacam arestas cobertas, não cobertas e o percurso de cada agente.

Essas ferramentas permitiram validar a coerência das rotas e comparar os métodos testados de maneira clara.

4 Resultados

4.1 Modelagem e Estratégia de Otimização

A etapa inicial consistiu na resolução do Problema do Carreiro Chinês (CPP), garantindo que todas as arestas fossem percorridas com custo mínimo. Para isso, foi empregado o Algoritmo de Edmonds–Johnson, assegurando a construção de rotas ótimas no cenário de agente único. Em seguida, os pesos das arestas foram ajustados pela incorporação da estimativa do número de casas por quarteirão, permitindo que o custo representasse não apenas distância, mas também esforço operacional.

Nos cenários multiagente, a divisão espacial da malha urbana foi realizada por meio de uma abordagem híbrida envolvendo o particionamento recursivo de Kernighan–Lin, utilizado para balancear o tamanho das sub-regiões, aliado à aproximação da mediana geográfica, responsável pela determinação de um ponto inicial comum que minimizasse deslocamentos iniciais para todos os agentes. Essa metodologia permitiu a geração de rotas mais equilibradas e coerentes com a distribuição espacial da cidade.

4.2 Tabela de Métricas Operacionais

Os principais indicadores de desempenho obtidos para cada cenário são apresentados na Tabela 1. Os valores consolidados incluem tempo total de execução das rotas, dias úteis estimados para completar o serviço, custo total e economia percentual de tempo em relação ao cenário de agente único.

Tabela 1: Métricas comparativas dos cenários com 1, 2 e 3 agentes.

Métrica	1 Agente	2 Agentes	3 Agentes
Tempo Total (horas)	12.94	6.69	5.32
Dias Úteis	1.62	0.84	0.66
Custo (R\$)	647.08	669.21	797.55
Economia de Tempo	–	49.3%	63.3%

4.3 Análise Comparativa dos Cenários

Os resultados quantitativos indicam que a transição do cenário com um agente para o cenário com dois agentes proporcionou ganhos expressivos de eficiência. Observou-se uma redução de 49.3% no tempo total necessário para percorrer todas as arestas, com aumento marginal de apenas 1.4% no custo operacional. Esse equilíbrio entre tempo e custo caracteriza o cenário de dois agentes como a solução mais vantajosa entre as alternativas avaliadas.

O cenário de três agentes apresentou o menor tempo absoluto, com redução de 63.3% em relação ao cenário base. No entanto, o custo adicional de 23.2% torna essa configuração menos competitiva sob a perspectiva custo-benefício. A análise evidencia que, embora o aumento no número de agentes reduza o tempo total, o ganho incremental não é proporcional ao acréscimo de custo após dois agentes.

4.4 Visualização e Validação das Rotas

Para complementar a análise numérica, foram gerados mapas interativos em HTML e animações em vídeo (MP4), permitindo validar visualmente o comportamento das rotas e verificar a coerência espacial da divisão obtida pelo particionamento. Essas visualizações facilitaram a inspeção do平衡amento entre agentes, da distribuição das arestas no espaço urbano e da eficiência dos trajetos gerados.

A Figura 4 apresenta um exemplo do relatório de métricas utilizado para comparar o desempenho entre os diferentes cenários.



Figura 3: Mapa das rotas geradas para o cenário com três agentes, ilustrando a divisão espacial resultante do particionamento e os trajetos percorridos por cada agente.

```
relatorio_metricas_3_agentes.txt - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
=====
RELATORIO DE METRICAS E CUSTOS OPERACIONAIS
=====
Data: 25/11/2025 21:41:18
Número de agentes: 3

RESUMO POR AGENTE:
-----
Agente 0: 266.01 min (4.43 h) - R$ 221.67
Agente 1: 319.02 min (5.32 h) - R$ 265.85
Agente 2: 283.47 min (4.72 h) - R$ 236.22

TEMPO TOTAL:
-----
Sequencial: 14.47 h (1.81 dias)
Paralelo: 5.32 h (0.66 dias)
Economia: 63.3%

CUSTOS OPERACIONAIS:
-----
Sequencial: R$ 723.75
Paralelo: R$ 797.55

Ln 14, Col 13 170% Unix (LF) UTF-8
```

Figura 4: Captura de tela do relatório de métricas e custos, comparando os cenários de cada agente.

5 Conclusões

Os resultados obtidos neste projeto demonstram de forma clara que a abordagem proposta para otimização das rotas de coleta em Elói Mendes/MG é eficiente, metodologicamente robusta e alinhada com o estado da arte em Problemas de Roteamento de Arcos (ARPs), conforme discutido em Corberán et al. (2021). A hipótese inicial de que a adoção de uma estratégia multiagente poderia melhorar significativamente o desempenho operacional em relação ao cenário single-agent foi confirmada. A configuração com dois agentes apresentou o melhor equilíbrio entre eficiência e custo, reduzindo o tempo total necessário para percorrer todas as arestas em 49,3%, ao mesmo tempo em que elevou o custo operacional em apenas 1,4%. Esse resultado evidencia um aumento real na produtividade sem comprometer o orçamento, reforçando a aplicabilidade prática do modelo desenvolvido e revelando que o particionamento adequado do grafo pode gerar ganhos substanciais no processo de coleta.

A eficiência alcançada decorre da combinação de três

elementos centrais. A utilização do Algoritmo de Edmonds-Johnson para resolver o Problema do Carteiro Chinês garantiu a obtenção de rotas ótimas para o cenário de agente único, assegurando que todas as arestas fossem percorridas com custo mínimo, conforme estabelecido por Edmonds e Johnson em Edmonds & Johnson (1973). A modelagem enriquecida dos pesos, ao incorporar a estimativa do número de casas em cada quarteirão, permitiu aproximar o grafo das condições reais enfrentadas pela equipe de coleta ao adicionar uma noção de esforço operacional. A etapa multiagente foi viabilizada por uma abordagem híbrida que integrou o particionamento recursivo de Kernighan-Lin, originalmente proposto em Kernighan & Lin (1970), com a mediana geográfica, o que garantiu uma divisão espacial equilibrada e minimizou os deslocamentos iniciais dos agentes.

Essas contribuições posicionam o projeto em consonância com o panorama contemporâneo dos ARPs discutido por Corberán et al. (2021), que destaca a expansão das aplicações práticas, a importância de múltiplos objetivos e a crescente relevância de problemas multiagente e cenários híbridos. A utilização de técnicas de busca espacial como a mediana geográfica e, em potenciais extensões, estruturas como as k-d trees descritas por Bentley em Bentley (1975), revela uma aproximação com tendências modernas de integração entre otimização combinatória e estruturação espacial de dados. Além disso, o pipeline completo apresentado integra otimização, cálculo de métricas operacionais e geração de visualizações ricas, incluindo mapas construídos a partir de OpenStreetMap OpenStreetMap (2025) e processados com OSMnx Boeing (2017), que foram essenciais para validar o comportamento das soluções e reforçar a compreensão do balanceamento entre agentes.

5.1 Trabalhos Futuros

Embora o projeto tenha atingido plenamente seu objetivo central, há perspectivas promissoras para evoluções futuras em consonância com as tendências discutidas na literatura. Uma possível extensão é o desenvolvimento de um ARP dinâmico e em tempo real, permitindo que mudanças no ambiente, como bloqueios temporários ou alterações na malha viária, influenciem a rota durante sua execução, conforme apontado em Corberán et al. (2021). Outra linha de avanço consiste na inclusão de janelas de tempo, que se tornariam relevantes caso a coleta de dados imobiliários passe a seguir restrições horárias específicas ou prioridades temporais. Estruturas de dados avançadas, como as k-d trees apresentadas em Bentley (1975), podem ser incorporadas para acelerar processos de clusterização espacial e aprimorar etapas preliminares do particionamento multiagente. O modelo também pode ser expandido para cenários híbridos e multicritério, contemplando objetivos adicionais relacionados ao equilíbrio de carga de trabalho, minimização de ruído ou redução de emissões. Finalmente, a aplicação da metodologia em municípios maiores ou em malhas urbanas mais complexas permitiria avaliar sua escalabilidade e compará-la com heurísticas consolidadas para o CARP e suas variantes.

Referências

- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9), 509–517.

Boeing, G. (2017). Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, environment and urban systems*, 65, 126–139.

Corberán, A., Eglese, R., Hasle, G., Plana, I., & Sanchis, J. M. (2021). Arc routing problems: A review of the past, present, and future. *Networks*, 77(1), 88–115.

Edmonds, J. & Johnson, E. L. (1973). Matching, euler tours and the chinese postman. *Mathematical programming*, 5(1), 88–124.

Kernighan, B. W. & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2), 291–307.

OpenStreetMap (2025). Openstreetmap. <https://www.openstreetmap.org>. Acessado em: 20 nov. 2025.

