



**UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

JAIME GOUVEIA DE SOUZA NETO

**HEURÍSTICAS DO PROBLEMA DO CAMINHO HAMILTONIANO MÍNIMO
APLICADAS À ROTEIRIZAÇÃO DA COLETA DE RESÍDUOS SÓLIDOS NA CIDADE
DE QUIXADÁ-CE**

QUIXADÁ

2025

JAIME GOUVEIA DE SOUZA NETO

HEURÍSTICAS DO PROBLEMA DO CAMINHO HAMILTONIANO MÍNIMO APLICADAS
À ROTEIRIZAÇÃO DA COLETA DE RESÍDUOS SÓLIDOS NA CIDADE DE QUIXADÁ-CE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Paulo Henrique
Macedo de Araujo.

QUIXADÁ

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S238h Souza Neto, Jaime Gouveia de.

Heurísticas do problema do caminho hamiltoniano mínimo : aplicadas à roteirização da coleta de resíduos sólidos na cidade de Quixadá-CE / Jaime Gouveia de Souza Neto. – 2025.

27 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2025.

Orientação: Prof. Dr. Paulo Henrique Macedo de Araujo.

1. Roteirização. 2. Resíduos Sólidos. 3. Caminho Hamiltoniano Mínimo. 4. Heurísticas. 5. Otimização de Rotas. I. Título.

CDD 005

JAIME GOUVEIA DE SOUZA NETO

HEURÍSTICAS DO PROBLEMA DO CAMINHO HAMILTONIANO MÍNIMO APLICADAS
À ROTEIRIZAÇÃO DA COLETA DE RESÍDUOS SÓLIDOS NA CIDADE DE QUIXADÁ-CE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Sistemas de Informação
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Sistemas de Informação.

Aprovada em: 01/08/2025.

BANCA EXAMINADORA

Prof. Dr. Paulo Henrique Macedo de
Araujo (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Lucas Ismaily Bezerra Freitas
Universidade Federal do Ceará (UFC)

Prof. Me. Luis Gustavo Coutinho do Rêgo
Instituto Federal de Educação, Ciência e Tecnologia
do Ceará (IFCE)

A Deus, pela força e guia que iluminaram meu caminho. À minha família, meu alicerce de apoio e confiança. À minha mãe, pelo cuidado e dedicação que me deram esperança para seguir em frente. À minha irmã, pelo incentivo e apoio. Com gratidão, dedico este trabalho a vocês.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001

Ao Prof. Dr. Paulo Henrique Macedo de Araujo, pela excelente orientação.

Aos professores participantes da banca examinadora Lucas Ismaily Bezerra Freitas e Luis Gustavo Coutinho do Rêgo pelo tempo, pelas valiosas colaborações e sugestões.

Aos colegas da turma de Projeto de Pesquisa Científica e Tecnológica, pelas reflexões, críticas e sugestões recebidas.

Ao professor da turma de Projeto de Pesquisa Científica e Tecnológica, Paulo de Tarso Guerra Oliveira, pelos ensinamentos.

"Planejamento urbano não é apenas planejamento do espaço, nem mesmo planejamento do trabalho. Para ser bem-sucedido, deve ser planejamento para o povo." (GEDDES, 1947, p. 22. (tradução nossa))

RESUMO

A gestão eficiente da coleta de resíduos sólidos é um desafio crescente nas cidades brasileiras. Este trabalho propõe a aplicação de heurísticas do Problema do Caminho Hamiltoniano Mínimo (PCHM) para otimizar rotas de coleta em pontos de descarte irregular de resíduos na cidade de Quixadá-CE. A partir de dados reais fornecidos pela prefeitura, foram implementadas adaptações das heurísticas do Vizinho Mais Próximo e da Aresta Mais Barata, utilizando a linguagem Python. O estudo de caso considerou um único caminhão partindo da garagem e finalizando no lixão municipal, visitando todos os pontos mapeados sem retornar à origem. Os resultados demonstraram que a heurística da Aresta Mais Barata apresentou melhor desempenho, com rotas mais eficientes tanto em distância quanto em tempo. Este trabalho contribui com uma solução prática e escalável para a gestão de resíduos, podendo subsidiar políticas públicas e ações logísticas mais eficazes no município.

Palavras-chave: Roteirização; Resíduos sólidos; Caminho Hamiltoniano; Heurísticas; Otimização de rotas; Caminho Hamiltoniano Mínimo.

ABSTRACT

Efficient solid waste collection management is a growing challenge in Brazilian cities. This study proposes the application of heuristics for the Shortest Hamiltonian Path Problem (SHPP) to optimize collection routes at irregular waste disposal points in the city of Quixadá-CE. Based on real data provided by the local government, adapted versions of the Nearest Neighbor and Cheapest Insertion heuristics were implemented using Python. The case study considered a single truck departing from the garage and ending at the municipal dump, visiting all mapped points without returning to the origin. Results showed that the Cheapest Insertion heuristic performed better, producing more efficient routes in both distance and time. This work offers a practical and scalable solution for waste management, providing a useful tool for public policy and more effective logistical planning.

Keywords: Routing; Solid waste; Hamiltonian Path; Heuristics; Route optimization; Shortest Hamiltonian Path.

LISTA DE FIGURAS

Figura 1 – Exemplo de uma representação visual de um grafo não direcionado	20
Figura 2 – Exemplo de grafo direcionado com pesos assimétricos entre os vértices A e B	21
Figura 3 – Grafo não direcionado completo	23
Figura 4 – Grafo direcionado completo	23
Figura 5 – Exemplo de um ciclo em um grafo direcionado	24
Figura 6 – Exemplo de um ciclo hamiltoniano em um grafo direcionado	25
Figura 7 – Problema do Caixeiro Viajante (PCV) representado por um grafo	26
Figura 8 – Comportamento assintótico entre diferentes complexidades da notação Big O	27
Figura 9 – Exemplo de Representação em um Grafo do Problema do Caminho Hamiltoniano Mínimo	31
Figura 10 – Zonas geradoras de lixo da cidade de Quixadá.	36
Figura 11 – Localização da garagem, pontos de lixo e lixão	37
Figura 12 – Rotas geradas pela Heurística da Inserção Mais Barata	44
Figura 13 – Rotas geradas pela heurística Vizinho Mais Próximo	45

LISTA DE TABELAS

Tabela 1 – Matriz de adjacência de um grafo direcionado com três vértices	22
Tabela 2 – Tabela da quantidade de permutações para resolver PCV com força bruta no caso assimétrico	26
Tabela 3 – Comparação entre heurísticas para critérios de distância e tempo	46
Tabela 4 – Matriz de Adjacência Ponderada por Distância em Metros	50
Tabela 5 – Matriz de Adjacência Ponderada por Tempo em Segundos	51

LISTA DE QUADROS

Quadro 1 – Comparação entre trabalhos relacionados e este estudo 34

LISTA DE ALGORITMOS

Algoritmo 1	– Heurística do Vizinho Mais Próximo(Nearest Neighbor Heuristics) . . .	28
Algoritmo 2	– Heurística da Inserção Mais Barata (Cheapest Insertion)	30
Algoritmo 3	– Heurística do Vizinho Mais Próximo adaptada ao Problema do Caminho Hamiltoniano Mínimo (PCHM) — Elaborado pelo autor	39
Algoritmo 4	– Heurística da Inserção Mais Barata (Cheapest Insertion) adaptada para o PCHM — Elaborado pelo autor	41

LISTA DE ABREVIATURAS E SIGLAS

PCHM Problema do Caminho Hamiltoniano Mínimo

PCV Problema do Caixeiro Viajante

WSL2 Windows Subsystem for Linux 2

ZGLS Zonas Geradoras de Lixo

LISTA DE SÍMBOLOS

A	Conjunto de arcos do grafo (para grafos direcionados)
a, b	Vértices consecutivos na rota atual, usados para calcular o custo de inserção
C	Ciclo em um grafo direcionado ou ciclo hamiltoniano
C_{ij}	Custo da aresta ou arco entre os vértices i e j na matriz de adjacência
c_{ij}	Peso da aresta entre os vértices i e j no Problema do Caminho Hamiltoniano Mínimo
ΔC	Custo incremental ao inserir um vértice entre dois vértices consecutivos na heurística da Inserção Mais Barata
$custo_total$	Soma acumulada dos custos entre os vértices consecutivos da rota
E	Conjunto de arestas do grafo (para grafos não direcionados)
e_{ij}	Aresta entre os vértices v_i e v_j em um grafo não direcionado
G	Grafo, definido como um par ordenado $G = (V, E)$ para grafos não direcionados ou $G = (V, A)$ para grafos direcionados
i, j	Índices genéricos para vértices
k	Índice genérico para vértices não visitados ou candidatos a inserção
K	Lista de vértices representando a rota parcial ou ciclo hamiltoniano nas heurísticas
K_n	Grafo completo não direcionado com n vértices
l	Vértice mais recentemente inserido na rota (vértice atual) na heurística do Vizinho Mais Próximo
$matriz$	Matriz de adjacência contendo os custos de tempo ou distância entre pares de vértices
$ V $	Número de vértices no grafo
v_i, v_j	Vértices genéricos do grafo
v_s	Vértice inicial no Problema do Caminho Hamiltoniano Mínimo
v_t	Vértice final no Problema do Caminho Hamiltoniano Mínimo
V	Conjunto de vértices do grafo

<i>visitados</i>	Vetor auxiliar que indica quais vértices já foram visitados na heurística do Vizinho Mais Próximo
<i>v_inseridos</i>	Vetor auxiliar que indica quais vértices já foram inseridos na rota na heurística da Inserção Mais Barata
<i>W</i>	Conjunto de vértices ainda não inseridos na rota nas heurísticas
<i>M</i>	Matriz de adjacência de ordem $n \times n$, que representa as conexões entre vértices do grafo
m_{ij}	Elemento da matriz <i>M</i> que indica a existência (1), ausência (0) ou o peso da aresta/arco de v_i para v_j

SUMÁRIO

1	INTRODUÇÃO	17
2	OBJETIVOS	19
2.1	Objetivo Geral	19
2.2	Objetivos Específicos	19
3	FUNDAMENTAÇÃO TEÓRICA	20
3.1	Grafos	20
3.1.1	<i>Grafos Direcionados</i>	20
3.1.2	<i>Matriz de Adjacência</i>	21
3.1.3	<i>Grafos Completos</i>	22
3.1.4	<i>Ciclos em Grafos Direcionados</i>	24
3.1.5	<i>Ciclos Hamiltonianos</i>	24
3.2	Problema do Caixeiro Viajante	25
3.2.1	<i>Métodos Exatos</i>	26
3.3	Métodos Heurísticos para o Problema do Caixeiro Viajante	27
3.3.1	<i>Heurística do Vizinho Mais Próximo (Nearest Neighbor Heuristics) . . .</i>	27
3.3.2	<i>Heurísticas de Inserção (Insertion Heuristics)</i>	29
3.3.2.1	<i>Heurística da Inserção Mais Barata (Cheapest Insertion)</i>	29
3.4	Problema do Caminho Hamiltoniano Mínimo (The Shortest Hamiltonian Path Problem)	31
4	TRABALHOS RELACIONADOS	32
4.1	Otimização de rotas na coleta de resíduos sólidos: um estudo de caso na cidade de Fortaleza	32
4.2	Aplicação do Problema do Caixeiro Viajante para otimização das rotas de uma empresa de doces	32
4.3	Caixeiro Viajante: aplicação da modelagem matemática na otimização de rotas em uma concessionária de energia elétrica	33
4.4	Quadro comparativo	34
5	METODOLOGIA	35
5.1	Estudo de Caso	35
5.1.1	<i>Coleta de Dados</i>	35

5.1.2	<i>Logística Atual da Coleta em Pontos de Lixo na Cidade de Quixadá-CE e Identificação do Problema de Otimização</i>	36
5.1.3	<i>Objeto de Estudo</i>	37
5.2	Adaptação de Heurísticas do Problema do Caixeiro Viajante ao Problema do Caminho Hamiltoniano Mínimo	38
5.2.1	<i>Adaptação da Heurística do Vizinho Mais Próximo (Nearest Neighbor) . .</i>	38
5.2.2	<i>Adaptação da Heurística da Inserção Mais Barata (Cheapest Insertion) . .</i>	40
5.3	Implementação e Análise de Resultados	42
6	RESULTADOS	43
6.1	Rotas Geradas pela Heurística da Inserção Mais Barata	43
6.2	Rotas Geradas pela Heurística do Vizinho Mais Próximo	45
6.3	Comparação das Heurísticas	46
7	CONCLUSÕES E TRABALHOS FUTUROS	47
	REFERÊNCIAS	48
	APÊNDICE A –MATRIZES DE ADJACÊNCIA	50

1 INTRODUÇÃO

A gestão eficiente de resíduos sólidos constitui um dos principais desafios contemporâneos, impulsionado pelo aumento da geração de resíduos decorrente do crescimento do consumo e da alta descartabilidade de produtos (Lima, 2015). Esse cenário demanda das autoridades públicas soluções logísticas eficazes para mitigar os impactos ambientais e sociais causados pelo manejo inadequado de resíduos (Lucena, 2023).

Uma pesquisa de campo realizada em Quixadá-CE, em parceria com a Prefeitura Municipal, identificou os chamados “pontos de lixo” — locais de descarte irregular de resíduos sólidos. Apesar de mapeados, esses pontos ainda não contam com uma rota específica para sua coleta, o que compromete a eficiência do sistema de limpeza urbana. Nesse contexto, a otimização dos processos logísticos torna-se essencial para aprimorar a gestão de resíduos sólidos no município.

Este estudo propõe a aplicação do Problema do PCHM e de heurísticas para otimizar rotas de coleta nos pontos de lixo identificados em Quixadá-CE, minimizando tempo e distância percorridos. Diferentemente do PCV, que pressupõe um ciclo fechado com retorno ao ponto de origem, o PCHM é mais adequado a este contexto, pois modela a rota do caminhão partindo da garagem e terminando no lixão, sem retorno imediato à origem. Essa característica reflete a realidade operacional do município, na qual a coleta é realizada por uma empresa terceirizada, sem informações sobre as atividades do caminhão após a chegada ao lixão — por exemplo, realização de outras tarefas.

O problema é modelado por meio de um grafo orientado completo, no qual os vértices representam a garagem, os pontos de lixo e o lixão, e as arestas indicam as distâncias ou tempos de deslocamento entre esses pontos. A escolha por um grafo orientado completo é justificada pela natureza do processo logístico, que considera direções específicas e a conexão entre todos os pontos, com custos associados.

Dada a complexidade computacional do PCHM, equivalente à do PCV (Reinelt, 1994), optou-se por métodos heurísticos, que oferecem soluções próximas da ótima em tempo computacional viável, especialmente em instâncias com maior número de localidades. Assume-se que a coleta será realizada por um único caminhão com capacidade suficiente para atender todos os pontos mapeados, de modo que a restrição de carga não é considerada na modelagem.

Ao final deste trabalho, será realizada uma análise comparativa das rotas geradas pelas heurísticas propostas, com o objetivo de fornecer subsídios à Prefeitura de Quixadá-CE

para a implementação de um sistema de coleta mais eficiente e econômico.

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver e aplicar heurísticas adaptadas ao PCHM para gerar rotas otimizadas de coleta de resíduos sólidos nos pontos de descarte irregular da cidade de Quixadá-CE, com foco na minimização do tempo e da distância percorridos.

2.2 Objetivos Específicos

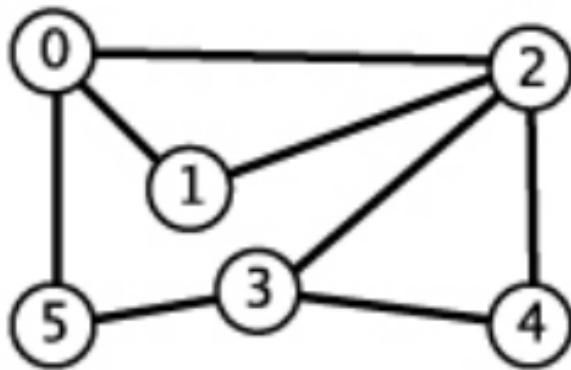
1. Adaptar e implementar heurísticas clássicas do PCV, como o Vizinho Mais Próximo e a Inserção Mais Barata, para o contexto do Caminho Hamiltoniano Mínimo com origem e destino distintos e fixos.
2. Aplicar as heurísticas implementadas sobre dados reais de pontos de lixo mapeados em Quixadá-CE, utilizando métricas reais de tempo e distância obtidas via Google Distance Matrix API.
3. Comparar o desempenho das rotas geradas por cada heurística, avaliando a eficiência em termos de distância total e tempo estimado de coleta.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Grafos

Segundo Bondy e Murty (1976), muitas situações e problemas do mundo real podem ser modelados por meio de um diagrama composto por um conjunto de pontos conectados por linhas. Esse diagrama é uma abstração matemática conhecida como grafo, denotado por G . Formalmente, um grafo é definido como um par ordenado $G = (V, E)$, onde $V = \{v_1, v_2, \dots, v_n\}$ é o conjunto de vértices (os pontos) e $E = \{e_{ij} = (v_i, v_j) \mid v_i, v_j \in V\}$ é o conjunto de arestas (as linhas que conectam pares de vértices). Em um grafo não direcionado, as arestas são simétricas, ou seja, $e_{ij} = e_{ji}$, indicando que a conexão entre v_i e v_j não possui direção.

Figura 1 – Exemplo de uma representação visual de um grafo não direcionado



Fonte: IME-USP (2017)

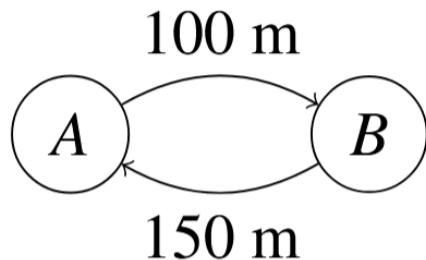
A figura 1 ilustra a representação visual de um grafo não direcionado, onde as conexões entre os vértices não possuem direção.

3.1.1 Grafos Direcionados

Um **grafo direcionado** (ou digrafo) é definido como $G = (V, A)$, onde V é o conjunto de vértices e $A = \{(v_i, v_j) \mid v_i, v_j \in V\}$ é o conjunto de arcos, representando conexões orientadas de v_i para v_j . Diferentemente dos grafos não direcionados, a existência de um arco $(v_i, v_j) \in A$ não implica que $(v_j, v_i) \in A$, caracterizando a assimetria do grafo direcionado (Bondy; Murty, 1976). Um grafo direcionado é dito **assimétrico** quando os pesos associados aos arcos (v_i, v_j) e (v_j, v_i) , se ambos existirem, são diferentes, refletindo, por exemplo, custos ou distâncias distintas em cada direção. Em aplicações práticas, como problemas de roteamento, é comum considerar grafos direcionados, nos quais existem arcos (v_i, v_j) e (v_j, v_i) para todo par de vértices $v_i, v_j \in V$,

com $i \neq j$, mas com pesos potencialmente assimétricos. Essa propriedade é particularmente relevante em problemas de otimização, como o Problema do Caixeiro Viajante e o Problema do Caminho Hamiltoniano Mínimo, onde os pesos dos arcos representam, por exemplo, distâncias ou tempos de deslocamento (Bondy; Murty, 1976, p. 171).

Figura 2 – Exemplo de grafo direcionado com pesos assimétricos entre os vértices A e B



Fonte: Elaborado pelo autor

A figura 2 apresenta um exemplo simplificado de um grafo direcionado, onde os arcos entre os vértices A e B possuem pesos diferentes, ilustrando a assimetria característica desse tipo de grafo.

3.1.2 Matriz de Adjacência

A matriz de adjacência é uma representação matricial de um grafo, amplamente utilizada em computação devido à sua simplicidade e eficiência em algoritmos. Para um grafo $G = (V, E)$ ou um grafo direcionado $G = (V, A)$ com $|V| = n$ vértices, a matriz de adjacência é uma matriz quadrada M de ordem $n \times n$, onde cada elemento m_{ij} indica a presença (ou peso) de uma aresta ou arco entre os vértices v_i e v_j . Formalmente, para um grafo não direcionado, $m_{ij} = 1$ se $(v_i, v_j) \in E$ e $m_{ij} = 0$ caso contrário, sendo a matriz simétrica ($m_{ij} = m_{ji}$). Para um grafo direcionado, $m_{ij} = 1$ se $(v_i, v_j) \in A$ e $m_{ij} = 0$ caso contrário, não havendo necessariamente simetria. Em grafos ponderados, m_{ij} pode armazenar o peso do arco ou aresta correspondente (Bondy; Murty, 1976, p. 7).

A Tabela 1 apresenta a matriz de adjacência de um grafo direcionado com três vértices $\{v_1, v_2, v_3\}$, onde, por exemplo, $m_{12} = 1$ indica um arco de v_1 para v_2 , e $m_{21} = 0$ indica a ausência de um arco de v_2 para v_1 .

A importância computacional da matriz de adjacência reside em sua capacidade

Tabela 1 – Matriz de adjacência de um grafo direcionado com três vértices

	v_1	v_2	v_3
v_1	0	1	0
v_2	0	0	1
v_3	1	0	0

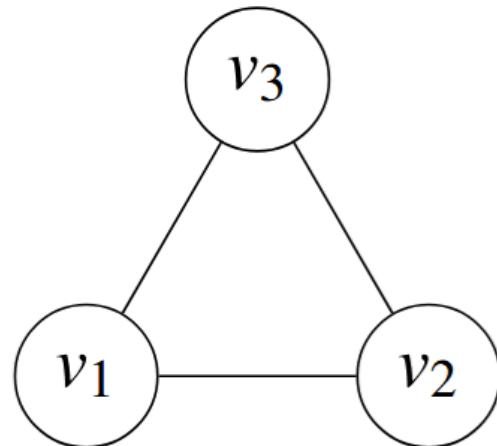
Fonte: Elaborado pelo autor

de representar grafos de forma compacta e eficiente para algoritmos. Em aplicações como o Problema do Caixeiro Viajante ou o Problema do Caminho Hamiltoniano Mínimo, a matriz de adjacência é frequentemente utilizada para armazenar pesos assimétricos, facilitando a implementação de algoritmos de otimização.

3.1.3 *Grafos Completos*

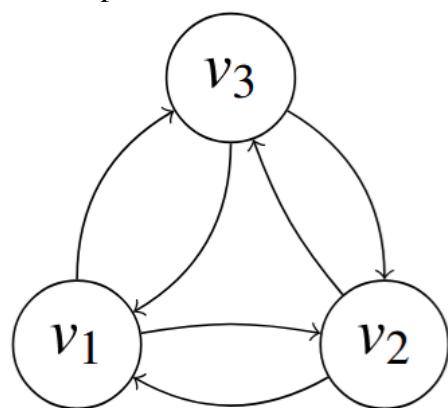
Um **grafo completo** é um grafo em que existe uma conexão entre cada par de vértices distintos. Em um grafo não direcionado $G = (V, E)$, com $|V| = n$, o grafo é completo, denotado por K_n , se existe uma aresta $e_{ij} \in E$ para todo $v_i, v_j \in V$, com $i \neq j$. Nesse caso, o grafo possui $\frac{n(n-1)}{2}$ arestas – ou seja, uma aresta para cada par distinto de vértices, já que a direção não é considerada. Em um grafo direcionado $G = (V, A)$, um grafo completo possui um arco $(v_i, v_j) \in A$ para todo par de vértices $v_i, v_j \in V$, com $i \neq j$, totalizando $n(n - 1)$ arcos – já que, para cada vértice, há $n - 1$ destinos possíveis, considerando a direção dos arcos. Grafos completos, tanto direcionados quanto não direcionados, são amplamente utilizados em problemas de otimização, como o Problema do Caixeiro Viajante, onde se assume que todos os pares de vértices estão conectados, possivelmente com pesos assimétricos nos arcos (Bondy; Murty, 1976, p. 4, 5).

Figura 3 – Grafo não direcionado completo



Fonte: Elaborado pelo autor

Figura 4 – Grafo direcionado completo



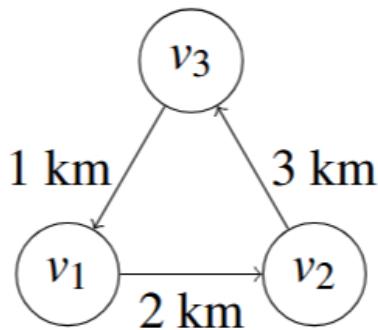
Fonte: Elaborado pelo autor

As figuras 3 e 4 ilustram, respectivamente, um grafo não direcionado completo (K_3), com três vértices conectados por todas as arestas possíveis, totalizando $\binom{3}{2} = 3$ arestas, e um grafo direcionado completo com três vértices, onde cada par de vértices possui arcos bidirecionais, totalizando $3 \cdot 2 = 6$ arcos. Em aplicações como o Problema do Caminho Hamiltoniano Mínimo, grafos direcionados completos são frequentemente utilizados, com pesos assimétricos nos arcos para representar, por exemplo, distâncias ou custos de deslocamento distintos em cada direção.

3.1.4 Ciclos em Grafos Direcionados

Um **ciclo** em um grafo direcionado $G = (V, A)$ é uma sequência de vértices $C = (v_1, v_2, \dots, v_k, v_1)$, onde todos os vértices v_1, v_2, \dots, v_k são distintos, exceto que v_k se conecta a v_1 , e cada par consecutivo (v_i, v_{i+1}) , para $1 \leq i < k$, bem como (v_k, v_1) , corresponde a um arco em A (Bondy; Murty, 1976, p. 14, 176). Em outras palavras, um ciclo é um caminho fechado que respeita a direção dos arcos.

Figura 5 – Exemplo de um ciclo em um grafo direcionado



Fonte: Elaborado pelo autor

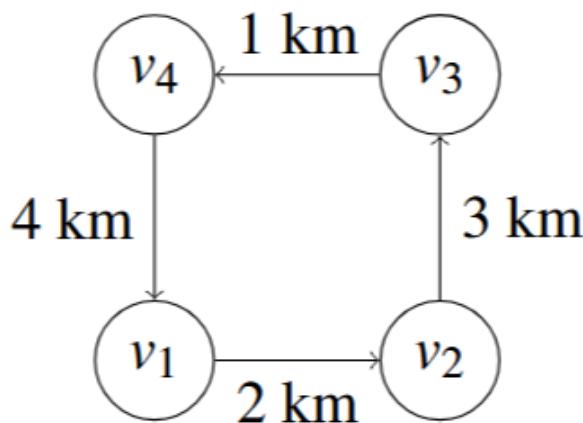
A Figura 5 ilustra um ciclo em um grafo direcionado, onde os vértices v_1 , v_2 e v_3 formam a sequência $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_1$, com pesos nos arcos, respeitando a direção.

3.1.5 Ciclos Hamiltonianos

Um **ciclo hamiltoniano** em um grafo direcionado é um ciclo que visita cada vértice do grafo exatamente uma vez, exceto pelo vértice inicial, que é revisitado para fechar o ciclo. Formalmente, dado um grafo direcionado $G = (V, A)$ com $|V| = n$, um ciclo hamiltoniano é uma sequência $C = (v_1, v_2, \dots, v_n, v_1)$, onde todos os v_i são distintos, $(v_i, v_{i+1}) \in A$ para $1 \leq i < n$,

e $(v_n, v_1) \in A$. Ciclos hamiltonianos são fundamentais em problemas de otimização, como o Problema do Caixeiro Viajante, que busca encontrar um ciclo hamiltoniano de custo mínimo em um grafo direcionado completo, passando por todos os vértices exatamente uma vez (Bondy; Murty, 1976, p. 53).

Figura 6 – Exemplo de um ciclo hamiltoniano em um grafo direcionado



Fonte: Elaborado pelo autor

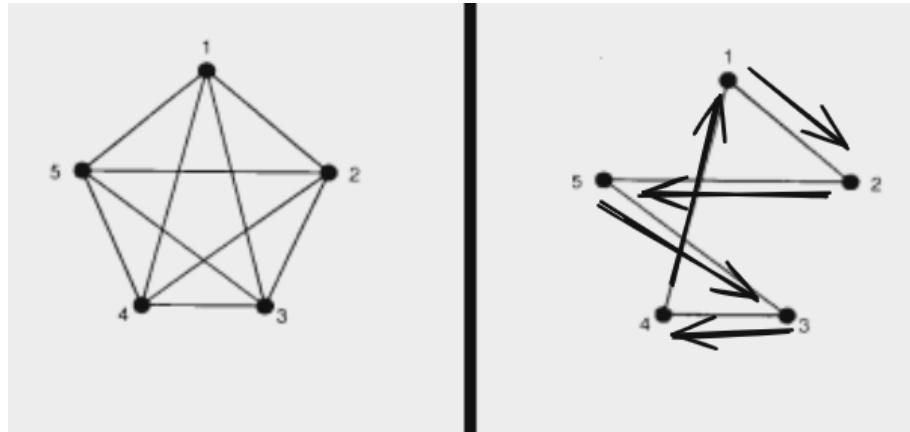
A figura 6 ilustra um ciclo hamiltoniano em um grafo direcionado com quatro vértices, onde a sequência $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_1$ visita todos os vértices exatamente uma vez, fechando o ciclo.

3.2 Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante é um problema clássico de otimização combinatória. Ele consiste em determinar o menor caminho que visita um conjunto de locais exatamente uma vez, retornando ao ponto inicial (Reinelt, 1994, p. 1). O PCV é frequentemente representado como um grafo, conforme mostrado na Figura 7.

O PCV está diretamente relacionado ao problema do ciclo hamiltoniano, que busca determinar se, em um grafo $G = (V, E)$, existe um ciclo que visita todos os vértices exatamente uma vez, retornando ao vértice inicial. No PCV, o objetivo é encontrar o ciclo hamiltoniano com o menor custo total (Reinelt, 1994, p. 6). Apesar de sua formulação simples, o PCV é um problema NP-difícil, e não admite solução em tempo polinomial, assumindo que $P \neq NP$ (Reinelt, 1994, pp. 7–11).

Figura 7 – PCV representado por um grafo



Fonte: Adaptado de Constantino (2012)

3.2.1 Métodos Exatos

Os métodos exatos para o PCV buscam a solução ótima, mas são computacionalmente custosos, especialmente para grafos com muitos vértices (Reinelt, 1994, p. 2).

Por exemplo, o método de força bruta testa todas as permutações possíveis dos vértices para encontrar o ciclo hamiltoniano de menor custo, com complexidade $O((N - 1)!)$ no caso assimétrico ou $O(\frac{(n-1)!}{2})$ no caso simétrico (Picolo, 2021). Essa abordagem torna-se inviável para grandes entradas devido ao crescimento factorial do número de soluções, conforme ilustrado na Tabela 2.

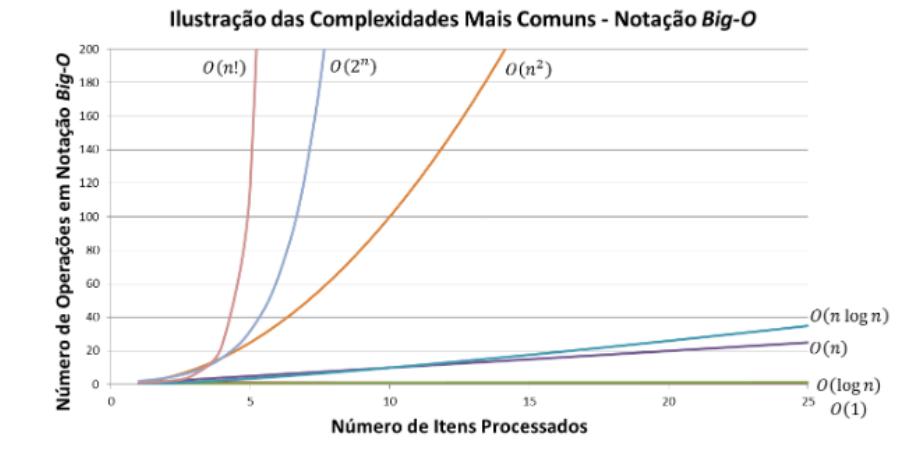
Tabela 2 – Tabela da quantidade de permutações para resolver PCV com força bruta no caso assimétrico

Número de Vértices	Número de Soluções Possíveis
5	24
9	40.320
13	479.001.600
17	20.922.789.888.000
21	2.432.902.008.176.640.000
26	15.511.210.043.330.985.984.000.000

Fonte: Adaptado de Picolo (2021)

O crescimento factorial da complexidade $O(n!)$ implica que o número de operações aumenta rapidamente com o tamanho da entrada, como mostrado na Figura 8. Essa característica torna o PCV um desafio computacional significativo.

Figura 8 – Comportamento assintótico entre diferentes complexidades da notação Big O



Fonte: Ribas (2018)

3.3 Métodos Heurísticos para o Problema do Caixeiro Viajante

Métodos heurísticos são algoritmos exploratórios que buscam resolver problemas através de aproximações, sem necessariamente encontrar uma solução ótima, utilizando estratégias que produzem soluções viáveis próximas do ótimo em tempo reduzido (Reinelt, 1994, p. 2).

Devido ao alto custo computacional dos métodos exatos, os métodos heurísticos são frequentemente utilizados para resolver o Problema do Caixeiro Viajante (Reinelt, 1994, p. 2). Abaixo, nas subseções, serão apresentados alguns exemplos.

3.3.1 Heurística do Vizinho Mais Próximo (Nearest Neighbor Heuristics)

A Heurística do Vizinho Mais Próximo é uma abordagem construtiva clássica amplamente utilizada para resolver o Problema do Caixeiro Viajante. Segundo Reinelt (1994, p. 73, 74), essa heurística consiste em construir uma rota de forma iterativa, sempre escolhendo o vértice mais próximo daquele que foi visitado por último e ainda não pertence à rota. Esse processo é repetido até que todos os vértices tenham sido inseridos, e ao final, o ciclo é fechado conectando o último vértice ao inicial.

A seguir, são apresentadas as definições utilizadas na formulação do algoritmo:

- V : conjunto total de vértices do grafo.
- W : conjunto dos vértices ainda não inseridos na rota.
- C_{ij} : custo da aresta entre quaisquer vértices i e j na matriz de adjacência.
- K : lista de vértices representando a rota parcial construída.

- l : vértice mais recentemente inserido na rota (vértice atual).

O pseudocódigo a seguir descreve a heurística:

Algoritmo 1: Heurística do Vizinho Mais Próximo(Nearest Neighbor Heuristics)

Entrada: Grafo completo com matriz de adjacência

Saída: Ciclo hamiltoniano

```

Escolha arbitrariamente um vértice  $j \in V$ ;
 $K \leftarrow [j]$ ;                                // Inicializa a rota com o vértice escolhido
 $l \leftarrow j$ ;                                  // Nó atual
 $W \leftarrow V \setminus \{j\}$ ;                     // Conjunto de vértices não inseridos

enquanto  $W \neq \emptyset$  faça
     $menor\_custo \leftarrow \infty$ ;
     $proximo \leftarrow \text{nulo}$ ;
    para cada  $k \in W$  faça
        Calcule o custo de inseri-lo na rota do ciclo atual  $C_{lk}$ 
        se  $C_{lk} < menor\_custo$  então
             $menor\_custo \leftarrow C_{lk}$ ;
             $proximo \leftarrow k$ ;
        fim se
    fim
    Conecte  $l$  a  $proximo$  na rota  $K$ ;
     $K \leftarrow K \cup \{proximo\}$ ;                  // Adiciona  $proximo$  ao conjunto da rota  $K$ 
     $W \leftarrow W \setminus \{proximo\}$ ;
     $l \leftarrow proximo$ ;
    Conecte o último vértice ao vértice inicial na rota  $K$  para fechar o ciclo;
    retorna  $K$ ;                                // Ciclo hamiltoniano com inserções sucessivas

```

3.3.2 Heurísticas de Inserção (Insertion Heuristics)

De acordo com Reinelt (1994, p. 82) as heurísticas de inserção são técnicas usadas para construir soluções aproximadas ao problema de encontrar o ciclo Hamiltoniano mais curto, elas funcionam começando com um ciclo inicial, geralmente pequeno, e inserindo os nós restantes ao decorrer do processo.

3.3.2.1 Heurística da Inserção Mais Barata (Cheapest Insertion)

Uma das estratégias construtivas clássicas para o PCV é a Heurística da Inserção Mais Barata (*Cheapest Insertion*), conforme descrito por Winston e Goldberg (2004, p. 534,551). Essa abordagem constrói iterativamente um ciclo hamiltoniano, partindo de um *subtour* inicial com dois vértices e expandindo-o até incluir todos os vértices do grafo. Esta heurística foi posteriormente adaptada para o PCHM neste trabalho, como descrito na seção 5.2.

A seguir, são apresentadas as definições utilizadas na descrição do algoritmo:

- V : conjunto total de vértices do grafo.
- W : conjunto dos vértices ainda não inseridos no ciclo.
- C_{ij} : custo da aresta entre quaisquer vértices i e j na matriz de adjacência.
- K : ciclo parcial atual, representado como uma lista ordenada de vértices.
- ΔC : custo incremental ao inserir um vértice entre dois vértices consecutivos do ciclo.

Inicialmente, seleciona-se um vértice arbitrário e seu vizinho mais próximo para formar um *subtour* inicial com duas arestas (por exemplo, (v_a, v_b) e (v_b, v_a)). Os demais vértices pertencem ao conjunto W , que representa os vértices ainda não inseridos na solução. A cada iteração, para cada vértice $k \in W$, o algoritmo calcula o custo de inseri-lo entre todos os pares consecutivos de vértices (i, j) já existentes no ciclo. Essa variação de custo, chamada ΔC , é calculada pela fórmula:

$$\Delta C = C_{ik} + C_{kj} - C_{ij}$$

Essa equação mede o aumento de custo causado ao substituir a aresta (i, j) por duas novas arestas (i, k) e (k, j) . Após calcular os custos de todas as inserções possíveis, o algoritmo escolhe a que apresenta o menor valor de ΔC . Em seguida, o vértice k é inserido na posição entre os vértices consecutivos que resultaram no menor ΔC , e é removido do conjunto

W . Esse processo é repetido até que todos os vértices tenham sido inseridos, formando um ciclo hamiltoniano.

O pseudocódigo a seguir descreve essa heurística:

Algoritmo 2: Heurística da Inserção Mais Barata (Cheapest Insertion)

Entrada: Grafo completo com matriz de adjacência

Saída: Ciclo hamiltoniano

Escolha um vértice $v_a \in V$ e seu vizinho mais próximo $v_b \in V \setminus \{v_a\}$;
 Inicialize um ciclo K com os vértices (v_a, v_b, v_a) formando um *subtour* fechado;
 $W \leftarrow V \setminus \{v_a, v_b\}$; // Conjunto de vértices não inseridos

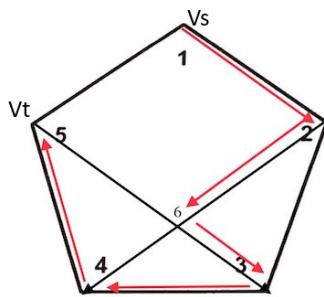
```

enquanto  $W \neq \emptyset$  faça
     $menor\_custo \leftarrow \infty;$ 
     $melhor\_k \leftarrow \text{nulo};$ 
     $melhor\_posicao \leftarrow \text{nulo};$ 
    para cada  $k \in W$  faça
        para cada par consecutivo  $(i, j)$  em  $K$  faça
            Calcule  $\Delta C = C_{ik} + C_{kj} - C_{ij}$  ;
            se  $\Delta C < menor\_custo$  então
                 $menor\_custo \leftarrow \Delta C;$ 
                 $melhor\_k \leftarrow k;$ 
                 $melhor\_posicao \leftarrow \text{posição entre } i \text{ e } j;$ 
            fim se
        fim
    fim
    Insira  $melhor\_k$  na  $melhor\_posicao$  em  $K$ ;
     $W \leftarrow W \setminus \{melhor\_k\}$ ;
retorna  $K$ ; // Ciclo hamiltoniano com inserções sucessivas
  
```

3.4 Problema do Caminho Hamiltoniano Mínimo (The Shortest Hamiltonian Path Problem)

Dado um grafo $G = (V, E)$ com pesos nas arestas c_{ij} e dois vértices especiais v_s (inicial) e v_t (final) pertencentes a V , o Problema do Caminho Hamiltoniano Mínimo é uma variação do PCV que busca um caminho de v_s a v_t que passe por cada vértice exatamente uma vez, minimizando a soma dos pesos das arestas Reinelt (1994, p.32), enquanto o PCV encontra um ciclo hamiltoniano de menor custo retornando ao ponto inicial (Reinelt, 1994, p.1).

Figura 9 – Exemplo de Representação em um Grafo do Problema do Caminho Hamiltoniano Mínimo



Fonte: Tatomir (2023)

Em abordagens exatas para o PCHM, é comum transformar o problema em uma instância do Problema do Caixeiro Viajante adicionando uma aresta entre os vértices v_s e v_t com peso $-M$, conforme discutido por Reinelt (1994, p. 32). No entanto, como este trabalho se baseia em heurísticas adaptadas diretamente ao PCHM, essa técnica não foi utilizada, pois pode interferir negativamente na lógica das heurísticas, como no caso do Vizinho Mais Próximo, cuja escolha local pode ser distorcida pela presença de uma aresta artificial com custo muito baixo.

4 TRABALHOS RELACIONADOS

A otimização de rotas é um tema amplamente explorado na literatura, com diversas abordagens propostas para melhorar a eficiência operacional em diferentes contextos.

Nesta seção, são revisados os principais trabalhos relacionados, destacando as contribuições mais relevantes e as metodologias empregadas.

4.1 Otimização de rotas na coleta de resíduos sólidos: um estudo de caso na cidade de Fortaleza

Conforme Mederos Nepomuceno (2018), foi realizada uma pesquisa de campo na cidade de Fortaleza-CE, com o objetivo de coletar dados sobre a logística de coleta de resíduos sólidos. A partir dos dados obtidos, o estudo identificou que a situação observada correspondia ao PCV. Diferente deste trabalho, que aborda o PCHM, uma variação do PCV.

Os autores propuseram um modelo exato de programação linear inteira para otimizar as rotas de coleta, justificado pelo número reduzido de pontos de coleta, que permitia o uso de métodos exatos. Em contraste, este trabalho optou por métodos heurísticos devido ao maior número de pontos considerados.

O modelo exato foi aplicado com dois objetivos distintos: na primeira abordagem, priorizou-se a redução da distância, resultando em uma diminuição de 5,94% na distância percorrida e de 2,35% no tempo de viagem; na segunda, a ênfase foi na redução do tempo, alcançando uma redução de 5,64% na distância percorrida e de 3,23% no tempo de viagem.

O estudo concluiu que a aplicação de técnicas de otimização melhora a eficiência do serviço de coleta, contribui para uma gestão mais sustentável dos resíduos e reduz os custos operacionais.

4.2 Aplicação do Problema do Caixeiro Viajante para otimização das rotas de uma empresa de doces

Conforme Queiroz *et al.* (2021), o PCV foi aplicado no contexto de uma microempresa do setor de confeitoraria que enfrentava desafios logísticos em suas entregas. A empresária, especializada na produção de doces artesanais e salgados, atendia principalmente escolas localizadas em diferentes pontos de Campo Grande-MS.

A análise da logística revelou a ausência de uma metodologia para as rotas de entrega,

resultando em custos operacionais elevados e prazos de entrega prolongados. A microempresa precisava visitar 11 pontos distintos na cidade, duas vezes por semana. Diante desse cenário, foi utilizado um modelo exato de programação linear inteira, resolvido com o Solver do Microsoft Excel, contrastando com este trabalho, que utiliza métodos heurísticos. O modelo gerou rotas que reduziram a quilometragem percorrida em 10,96%.

O estudo, além de aplicar o PCV de forma prática, forneceu um modelo replicável para outras microempresas, evidenciando a importância da pesquisa operacional na otimização de processos logísticos.

4.3 Caixeiro Viajante: aplicação da modelagem matemática na otimização de rotas em uma concessionária de energia elétrica

Conforme Ferreira (2020), foi conduzido um estudo de caso em uma concessionária de distribuição de energia elétrica no Brasil, aplicando o PCV para otimizar as rotas das equipes de manutenção de subestações. A pesquisa demonstrou que a falta de planejamento eficiente das rotas resultava em deslocamentos desnecessários, aumentando os custos operacionais e o tempo de execução dos serviços.

Para abordar o problema, os autores utilizaram um modelo de programação linear integrado ao Visual Basic for Applications (VBA) e ao Microsoft Excel 2007. O modelo proposto gerou rotas que minimizaram a distância percorrida pelas equipes.

Os resultados mostraram que a aplicação do modelo de otimização reduziu significativamente as distâncias percorridas, melhorando a eficiência operacional da concessionária. O estudo concluiu que a implementação de técnicas de otimização não apenas aprimora a logística das equipes de manutenção, mas também contribui para uma gestão mais eficaz dos recursos.

4.4 Quadro comparativo

O Quadro 1 compara os trabalhos relacionados com este estudo, destacando características como o problema abordado, a abordagem utilizada, a aplicação prática e o contexto de coleta de resíduos sólidos.

Quadro 1 – Comparação entre trabalhos relacionados e este estudo

Trabalho	Problema abor-dado	Abordagem	Aplicação prática	Coleta de resíduos
Mederos Nepomuceno (2018)	PCV	Programação linear inteira	Sim	Sim
Queiroz <i>et al.</i> (2021)	PCV	Programação linear inteira	Sim	Não
Ferreira (2020)	PCV	Programação linear com VBA	Sim	Não
Este trabalho	PCHM	Heurísticas (Vizinho Mais Próximo, Inserção Mais Barata)	Sim	Sim

Fonte: Elaborado pelo autor.

5 METODOLOGIA

Neste capítulo, são descritos os métodos utilizados para a realização deste trabalho, os quais envolvem as etapas de coleta de dados, estudo de caso, adaptação de heurísticas do Problema do Caixeiro Viajante para a resolução do Problema do Caminho Hamiltoniano Mínimo, bem como a implementação e a análise dos resultados. O desenvolvimento dessas atividades segue uma abordagem exploratória, prática e aplicada.

5.1 Estudo de Caso

Este capítulo apresenta um estudo de caso sobre a gestão de resíduos sólidos em Quixadá-CE, com foco na otimização das rotas de coleta em pontos de lixo. A análise revelou que o problema pode ser modelado como um PCHM.

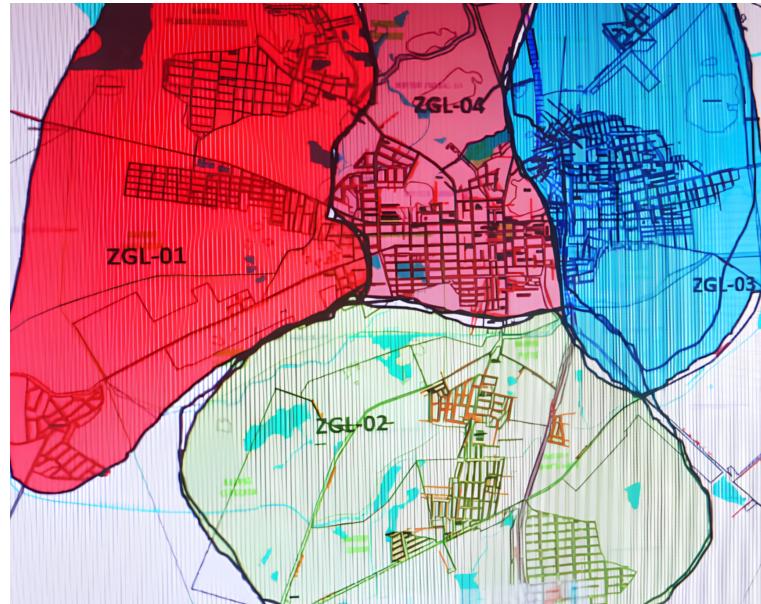
5.1.1 *Coleta de Dados*

Uma pesquisa de campo foi realizada para obter dados sobre a coleta de resíduos sólidos em Quixadá-CE, incluindo a localização geográfica dos pontos de lixo, da garagem dos caminhões e do lixão. Essas informações, fornecidas pela Prefeitura Municipal de Quixadá-CE, são essenciais para a modelagem do problema de otimização.

5.1.2 Logística Atual da Coleta em Pontos de Lixo na Cidade de Quixadá-CE e Identificação do Problema de Otimização

Para melhorar a gestão de resíduos sólidos, Quixadá-CE foi dividido em quatro Zonas Geradoras de Lixo (ZGLS), conforme ilustrado na Figura 10.

Figura 10 – Zonas geradoras de lixo da cidade de Quixadá.

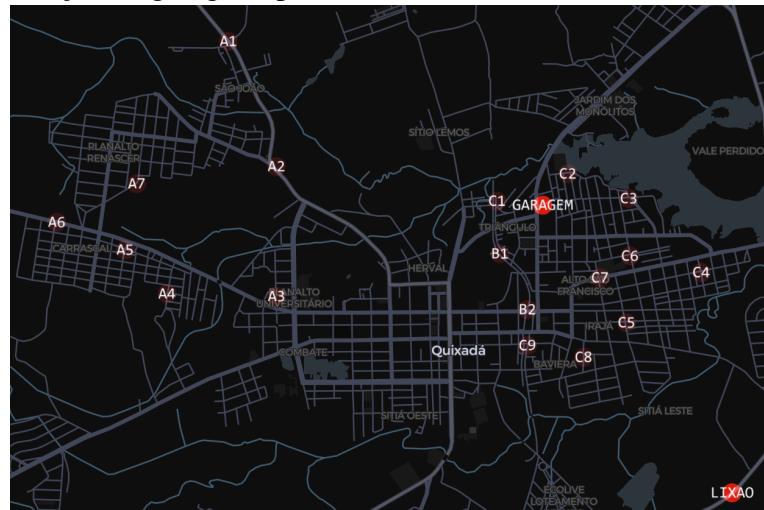


Fonte: Fornecido pela Prefeitura de Quixadá-CE

Como em muitos municípios brasileiros, Quixadá enfrenta desafios na gestão de resíduos, especialmente devido a pontos de lixo — locais de descarte irregular em áreas públicas, como terrenos baldios e margens de rios. Três das quatro ZGLS possuem pontos de lixo mapeados, mas sem rotas definidas, o que reduz a eficiência da coleta. A falta de comunicação entre a prefeitura e a empresa terceirizada responsável pela coleta dificulta a obtenção de dados detalhados sobre a operação logística. Assim, este estudo considera que a coleta é realizada por um único caminhão com capacidade suficiente para atender todos os pontos mapeados, eliminando a restrição de carga na modelagem.

A prefeitura forneceu mapas separados das três ZGLS, contendo as localizações dos pontos de lixo (A1–A7 na ZGL1, B1–B2 na ZGL4, C1–C9 na ZGL3), da garagem e do lixão. Esses mapas foram unificados em uma única representação utilizando a ferramenta Kepler.gl, para facilitar a visualização da distribuição geográfica, conforme apresentado na Figura 11.

Figura 11 – Localização da garagem, pontos de lixo e lixão



Fonte: Elaborado pelo autor com base em kepler.gl (2025)

O problema foi modelado como um PCHM, uma variação do Problema do Caixeiro Viajante. Nesse caso, é necessário que a rota se inicie no ponto inicial (a garagem), percorra todos os pontos de lixo e termine em um ponto final distinto (o lixão). Essa modelagem é apropriada devido à operação logística do município, na qual o caminhão, operado por uma empresa terceirizada, segue diretamente ao lixão após a coleta, sem informações sobre atividades posteriores.

O problema é representado por um grafo completo direcionado assimétrico, com vértices correspondendo à garagem, aos pontos de lixo e ao lixão, e arestas com pesos baseados em distâncias ou tempos de deslocamento. O grafo é completo, pois há deslocamento direto entre quaisquer pontos, conforme dados da Distance Matrix API do Google. É direcionado, devido à trajetória unidirecional (garagem ao lixão), e assimétrico, pois as distâncias e tempos variam conforme a direção, como evidenciado nas matrizes de distância (Tabela 4) e de tempo (Tabela 5). Por exemplo, a distância da garagem ao ponto A1 (3246 m) difere da distância de A1 à garagem (3355 m), devido a fatores como sentidos únicos ou tráfego.

5.1.3 Objeto de Estudo

Este estudo abrange três ZGLS, incluindo os pontos de lixo mapeados (A1–A7 na ZGL1, B1–B2 na ZGL4, C1–C9 na ZGL3), a garagem (ponto inicial) e o lixão (ponto final), conforme ilustrado na Figura 11. As matrizes de distância (em metros) e tempo (em segundos) entre esses pontos, obtidas via Distance Matrix API do Google e apresentadas nas Tabelas 4 e 5, viabilizam a modelagem computacional do problema como um grafo completo direcionado

assimétrico, viabilizando a aplicação de algoritmos para otimização de rotas.

5.2 Adaptação de Heurísticas do Problema do Caixeiro Viajante ao Problema do Caminho Hamiltoniano Mínimo

As heurísticas descritas na fundamentação teórica (3.3), originalmente desenvolvidas para o PCV, que exige um ciclo fechado, foram adaptadas por este autor para o PCHM. Foram adaptadas a Heurística da Inserção Mais Barata e a Heurística do Vizinho Mais Próximo, considerando os pontos de início e fim fixos. A seguir, são apresentados os pseudocódigos dessas adaptações.

5.2.1 Adaptação da Heurística do Vizinho Mais Próximo (*Nearest Neighbor*)

A heurística do Vizinho Mais Próximo é uma técnica clássica para construção de rotas em problemas de otimização, e neste trabalho foi adaptada para o contexto do PCHM.

A estratégia consiste em partir do vértice de origem e, a cada iteração, selecionar o vértice não visitado mais próximo (com menor custo direto a partir do ponto atual). Após todos os vértices intermediários serem percorridos, o destino é adicionado ao final da rota.

Para a implementação da heurística, são utilizados os seguintes conceitos:

- **matriz**: matriz de adjacência contendo os custos de tempo ou distância entre cada par de vértices;
- **origem**: vértice fixo que representa o ponto de partida da rota (exemplo: garagem);
- **destino**: vértice fixo que representa o ponto final da rota (ex: lixão);
- **visitados**: vetor auxiliar que indica quais vértices já foram visitados;
- **ponto_atual**: vértice onde a busca está no momento, buscando o vizinho mais próximo;
- **proximo_ponto**: vértice mais próximo não visitado (excluindo o destino);
- **caminho**: lista ordenada de vértices representando a solução em construção;
- **custo_total**: soma acumulada dos custos entre os vértices consecutivos da rota.

Algoritmo 3: Heurística do Vizinho Mais Próximo adaptada ao PCHM — Elaborado pelo autor

Entrada: matriz de adjacência (*matriz*), vértice de origem, vértice de destino
Saída: Rota final e custo total da rota

```

visitados[i] ← Falso, ∀i ∈ V ;           // Inicializa todos como não visitados
ponto_atual ← origem ;
visitados[origem] ← Verdadeiro ;
caminho ← [origem] ;
custo_total ← 0 ;

// -2: ignora origem e destino; -1: índice começa em 0; logo -3
para i = 0 até |V| - 3 faça
    menor_distancia ← ∞ ;
    proximo_ponto ← nulo ;
    para cada vértice k ∈ V, tal que visitados[k] = Falso e k ≠ destino faça
        se matriz[ponto_atual][k] < menor_distancia então
            menor_distancia ← matriz[ponto_atual][k] ;
            proximo_ponto ← k ;
        fim se
    fim
    se proximo_ponto = nulo então
        interromper laço ;                  // Nenhum ponto válido restante
    fim se
    visitados[proximo_ponto] ← Verdadeiro ;
    caminho.adiciona(proximo_ponto) ;
    custo_total ← custo_total + menor_distancia ;
    ponto_atual ← proximo_ponto ;
fim
caminho.adiciona(destino) ;
custo_total ← custo_total + matriz[ponto_atual][destino] ;
retorna caminho, custo_total ;

```

5.2.2 Adaptação da Heurística da Inserção Mais Barata (*Cheapest Insertion*)

A Heurística da Inserção Mais Barata (ou Cheapest Insertion) é uma abordagem construtiva amplamente utilizada para resolver variantes do Problema do Caixeiro Viajante e, neste trabalho, foi adaptada ao contexto do Problema do Caminho Hamiltoniano Mínimo com pontos de início e fim fixos. A ideia central da adaptação da heurística é construir progressivamente uma rota a partir de dois vértices fixos — um de origem e outro de destino — inserindo, a cada iteração, o vértice não visitado que cause o menor aumento de custo total ao ser incluído na posição ideal da rota atual.

Para a implementação da heurística, são utilizados os seguintes conceitos:

- **matriz**: matriz de adjacência contendo os custos de tempo ou distância entre cada par de vértices;
- **origem**: vértice fixo que representa o ponto de partida da rota (exemplo: garagem);
- **destino**: vértice fixo que representa o ponto final da rota (ex: lixão);
- **rota**: lista ordenada de vértices representando a solução em construção (inicialmente $[origem, destino]$);
- **v_inseridos**: vetor auxiliar que indica quais vértices já foram inseridos na rota;
- **candidato**: vértice ainda não inserido que está sendo avaliado para inclusão;
- **aumento_custo**: diferença entre o custo original de um par consecutivo (a, b) da rota e o custo resultante da inserção do vértice candidato entre eles.
- C_{ab} : representa o custo entre dois vértices quaisquer a e b .

Algoritmo 4: Heurística da Inserção Mais Barata (Cheapest Insertion) adaptada para o PCHM — Elaborado pelo autor

Entrada: matriz de adjacência ($matriz$), vértice de origem, vértice de destino

Saída: Rota final e custo total da rota

```

rota ← [origem,destino] ;           // Inicia a rota com origem e destino
v_inseridos[i] ← Falso, ∀i ∈ V; ;   // Todos os vértices como falso
v_inseridos[origem] ← Verdadeiro ;   // Vértice origem como verdadeiro
v_inseridos[destino] ← Verdadeiro ;  // Vértice destino como verdadeiro
custo_total ← 0

enquanto existir  $v \in V$  tal que  $v\_inseridos[v] = Falso$  faça
    menor_aumento_custo ←  $\infty$  ;
    melhor_vertice ← nulo ;
    melhor_posicao ← nulo ;

    para cada candidato ∈ V onde  $v\_inseridos[candidato] = Falso$  faça
        para cada par de vértices consecutivos ( $a,b$ ) na rota atual faça
            custo_sem_candidato =  $matriz[a][b]$  ;
            custo_com_candidato =  $matriz[a][candidato] + matriz[candidato][b]$  ;
            aumento_custo = custo_com_candidato - custo_sem_candidato ;

            se aumento_custo < menor_aumento_custo então
                menor_aumento_custo ← aumento_custo ;
                melhor_vertice ← candidato ;
                melhor_posicao ← posição entre  $a$  e  $b$  ;
            fim se
        fim
    fim
    Inserir(melhor_vertice em rota na posição melhor_posicao);
    v_inseridos[melhor_vertice] ← Verdadeiro;

    para cada pares consecutivos ( $a,b$ ) na rota faça
        custo_total ← custo_total +  $C_{ab}$  ;
    fim
    retorna rota, custo_total ;

```

5.3 Implementação e Análise de Resultados

Foram implementadas duas heurísticas adaptadas ao PCHM, utilizando a linguagem de programação Python: a Heurística do Vizinho Mais Próximo (Algoritmo 3) e a Heurística da Inserção Mais Barata (Cheapest Insertion) (Algoritmo 4). Ambas têm como objetivo gerar rotas otimizadas para a coleta de resíduos sólidos em pontos de descarte localizados na cidade de Quixadá-CE.

O código-fonte desenvolvido está disponível em repositório público no GitHub, acessível em: github.com/jaimegsn/pchm-coleta-residuos.

As execuções e testes foram realizados em um computador pessoal com as seguintes especificações: processador AMD Ryzen 7 5700G, 16 GB de memória RAM e armazenamento SSD de 1 TB, rodando o sistema operacional Windows 11 com subsistema Windows Subsystem for Linux 2 (WSL2). Foi utilizado o ambiente de desenvolvimento Anaconda com um ambiente virtual personalizado.

Posteriormente, os resultados das rotas geradas foram comparados e analisados com base em métricas de desempenho, como o tempo total de percurso e a distância percorrida, a fim de avaliar a eficácia das heurísticas implementadas.

6 RESULTADOS

Este capítulo apresenta os resultados obtidos a partir da aplicação das heurísticas desenvolvidas no contexto do Problema do Caminho Hamiltoniano Mínimo, com pontos de início e fim fixos. As heurísticas implementadas — Vizinho Mais Próximo e Inserção Mais Barata — foram aplicadas sobre dados reais dos pontos de descarte de resíduos sólidos urbanos na cidade de Quixadá-CE.

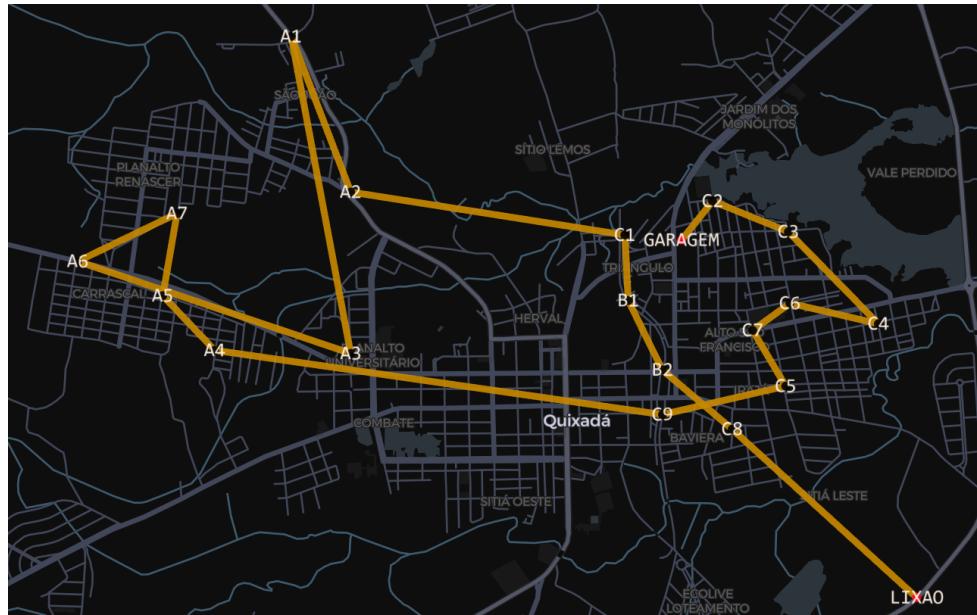
As distâncias e os tempos entre os pontos foram obtidos por meio da API Google Distance Matrix, assegurando que a matriz de adjacência utilizada nas simulações representa trajetos reais entre os locais. As Tabelas 4 e 5 apresentam, respectivamente, as matrizes de adjacência ponderadas por distância (em metros) e por tempo (em segundos), e servem como base para a modelagem do grafo utilizado pelos algoritmos.

Contudo, vale ressaltar que as rotas exibidas nas figuras ao longo deste capítulo têm caráter ilustrativo. Elas representam a ordem de visitação dos pontos, conforme determinado por cada heurística, mas não correspondem necessariamente ao traçado real das vias urbanas. Seu propósito é exclusivamente facilitar a compreensão visual da sequência de deslocamento entre os pontos.

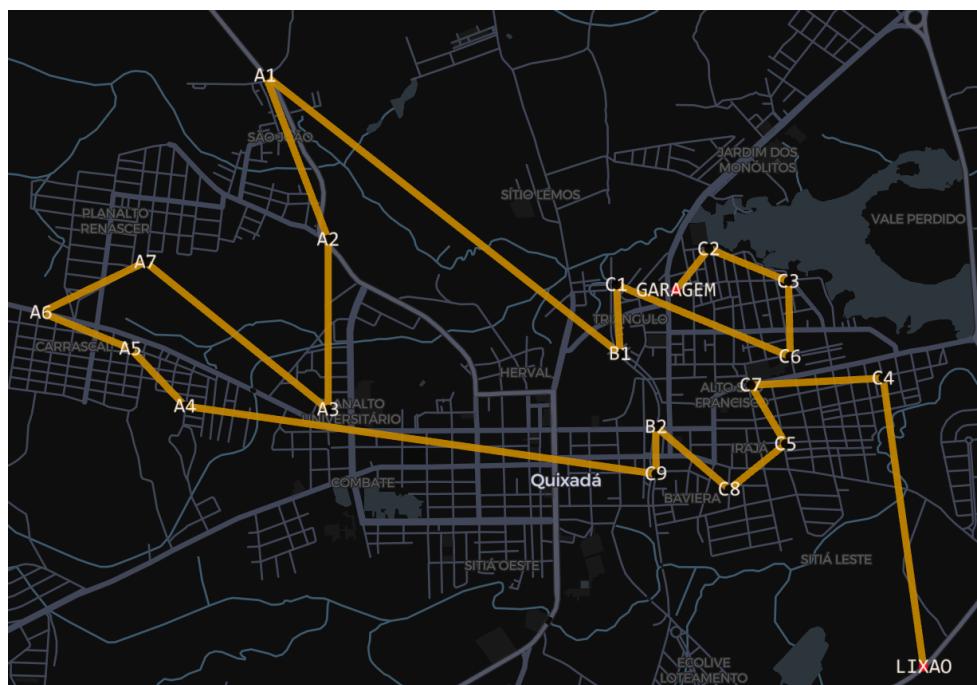
As subseções a seguir detalham os resultados obtidos com a aplicação de cada heurística sob os dois critérios de otimização considerados: distância e tempo.

6.1 Rotas Geradas pela Heurística da Inserção Mais Barata

As Figuras 12a e 12b apresentam as rotas geradas pela Heurística da Inserção Mais Barata, descrita no Algoritmo 4, considerando dois critérios distintos: distância mínima e tempo mínimo.



(a) Critério: distância

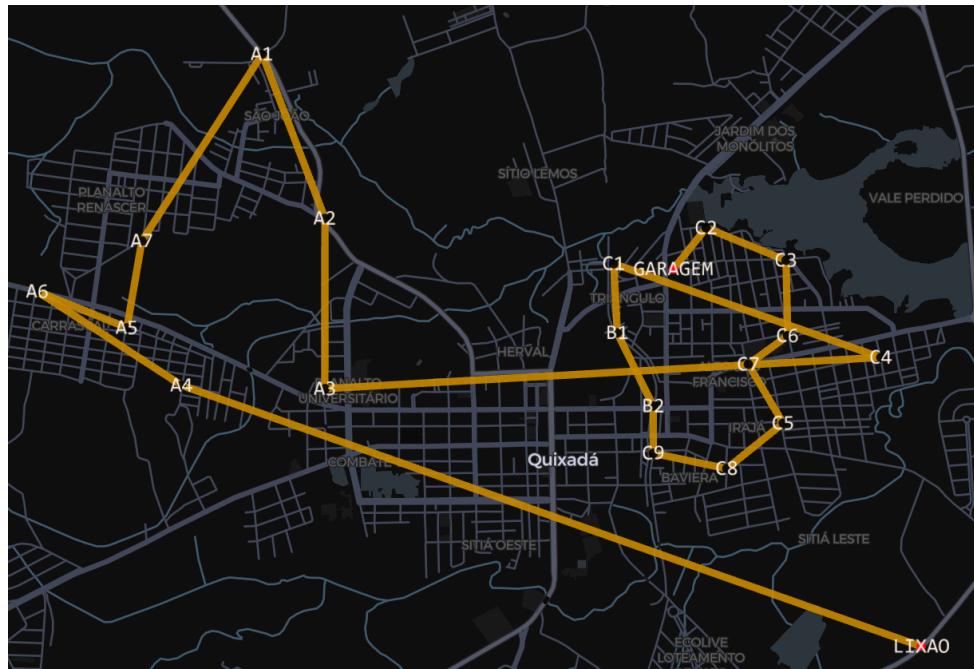


(b) Critério: tempo

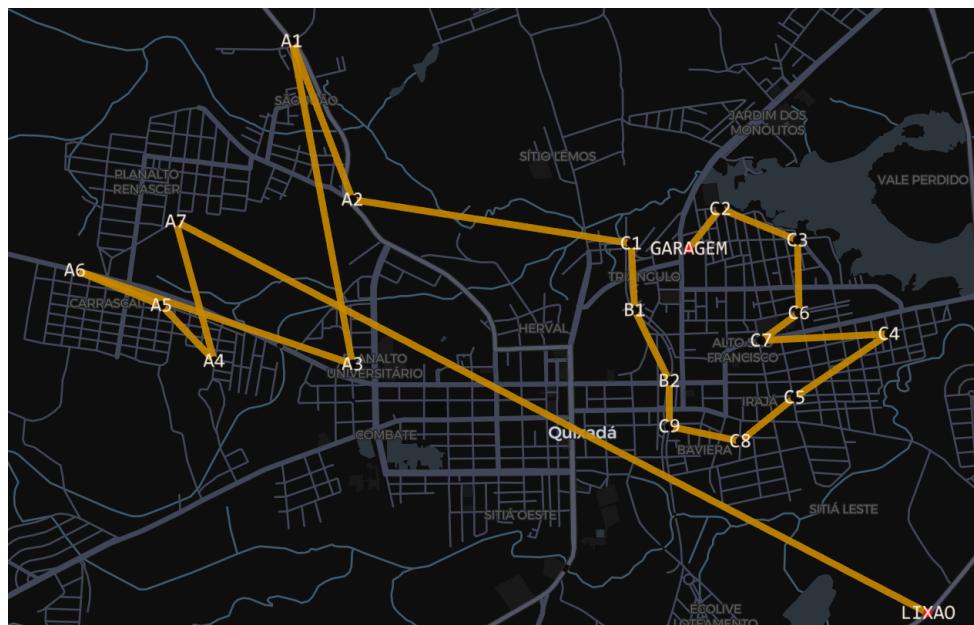
Figura 12 – Rotas geradas pela Heurística da Inserção Mais Barata
Fonte: Elaborado pelo autor com base em kepler.gl (2025)

6.2 Rotas Geradas pela Heurística do Vizinho Mais Próximo

As Figuras 13a e 13b ilustram as rotas obtidas pela heurística do Vizinho Mais Próximo, também considerando os critérios de distância e tempo.



(a) Critério: distância



(b) Critério: tempo

Figura 13 – Rotas geradas pela heurística Vizinho Mais Próximo

Fonte: Elaborado pelo autor com base em kepler.gl (2025)

6.3 Comparação das Heurísticas

A Tabela 3 apresenta os resultados comparativos entre as heurísticas implementadas, considerando os critérios de distância e tempo. Para cada rota gerada com base em um critério, também foi avaliado o impacto no critério alternativo, permitindo uma análise mais completa da eficiência de cada abordagem.

Tabela 3 – Comparação entre heurísticas para critérios de distância e tempo

Heurística	Foco em Distância		Foco em Tempo	
	Distância (m)	Tempo (s)	Tempo (s)	Distância (m)
Vizinho Mais Próximo	21.034	3.389	20.092	3.080
Inserção Mais Barata	19.080	3.070	19.367	2.916

Fonte: Elaborado pelo autor

Com base nos resultados obtidos, observa-se que a Heurística da Inserção Mais Barata produziu rotas mais eficientes em ambos os critérios avaliados. Quando o foco foi minimizar a distância, essa abordagem gerou um trajeto aproximadamente 10% menor em relação à Heurística do Vizinho Mais Próximo. De forma semelhante, ao priorizar o tempo, a Heurística da Inserção Mais Barata também obteve melhor desempenho, com redução tanto no tempo total quanto na distância percorrida.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma abordagem prática para o Problema do Caminho Hamiltoniano Mínimo adaptado à realidade da coleta de resíduos sólidos urbanos na cidade de Quixadá-CE. Foram implementadas e avaliadas duas heurísticas clássicas — Vizinho Mais Próximo e Inserção Mais Barata — com o objetivo de gerar rotas otimizadas de coleta, considerando tanto o tempo quanto a distância como critérios principais.

Os resultados demonstraram que a Heurística da Inserção Mais Barata apresentou melhor desempenho em ambos os critérios avaliados, superando a abordagem do Vizinho Mais Próximo em termos de eficiência das rotas geradas. A diferença observada reforça a importância de escolher a estratégia de construção de rota mais adequada ao problema, especialmente em cenários reais, onde economia de tempo e distância impactam diretamente na operação logística.

Além disso, a visualização das rotas sobre o mapa da cidade permitiu avaliar qualitativamente a viabilidade das soluções propostas, contribuindo para a análise prática da aplicabilidade do modelo.

Como trabalhos futuros, sugere-se explorar heurísticas mais sofisticadas, como algoritmos genéticos ou colônia de formigas, além de incorporar restrições reais como capacidade dos veículos, janelas de tempo ou tipos de resíduos.

REFERÊNCIAS

- BONDY, J.; MURTY, U. **Graph Theory with Applications.** [S. l.]: Elsevier Science Publishing Co., Inc., 1976.
- CONSTANTINO, P. A. **Grafos Hamiltonianos e o Problema do Caixeiro Viajante.** Departamento de Informática Universidade Estadual de Maringá, 2012. Disponível em: <https://malbarbo.pro.br/arquivos/2012/1747/problema-do-caixeiro-viajante.pdf>. Acesso em: 10 set. 2024.
- FERREIRA, I. A. Caixeiro viajante: aplicação da modelagem matemática na otimização de rotas em uma concessionária de energia elétrica. **Revista Produção Online**, v. 20, n. 1, p. 221–246, mar. 2020. Disponível em: <https://www.producaoonline.org.br/rpo/article/view/3491>. Acesso em: 20 ago. 2024.
- IME-USP, P. F. **Exemplo de um Grafo Não Direcionado.** IME-USP, 2017. Disponível em: https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/graphs.html. Acesso em: 1 ago. 2025.
- KEPLER.GL. **kepler.gl.** 2025. Disponível em: <https://kepler.gl/>. Acesso em: jul. 2025.
- LIMA, G. F. da C. Consumo e resíduos sólidos no brasil: as contribuições da educação ambiental. **Brazilian Journal of Environmental Sciences (RBCIAMB)**, n. 37, p. 47–57, 2015.
- LUCENA, M. V. P. J. Isabela Pinho de. Políticas públicas na gestão de resíduos sólidos. In: **Anais da XI Jornada Internacional de Políticas Públicas**. São Luís: Universidade Federal do Maranhão, Programa de Pós-Graduação em Políticas Públicas, 2023. ISSN 2175-280X. Disponível em: <https://www.joinpp.ufma.br/jornadas/joinpp2023/anais.html>. Acesso em: 15 jul. 2024.
- MEDEROS NEPOMUCENO, N. V. N. L. T. Otimização de rotas na coleta de resíduos sólidos: Um estudo de caso na cidade de fortaleza. **SBPO**, Galoá, 2018.
- PICOLO, A. P. **Aplicação do Problema do Caixeiro Viajante para determinação de rotas turísticas.** 2021. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Veranópolis, Veranópolis, RS. Disponível em: <https://dspace.ifrs.edu.br/xmlui/handle/123456789/1278>. Acesso em: 7 ago. 2024.
- QUEIROZ, C. Z. d.; NETO, V. d. A.; WAGATUMA, L. S. U.; CAMARGO, C. L. M. Pompeo de. Aplicação do problema de caixeiro viajante para otimização das rotas de uma empresa de doces. In: **Anais do I Simpósio Nacional de Engenharia de Produção.** Campo Grande, MS: Universidade Federal da Grande Dourados, 2021. Disponível em: <https://ocs.ufgd.edu.br/index.php?conference=sinep&schedConf=ISINEP&page=paper&op=view&path%5B%5D=1154>. Acesso em: 13 ago. 2024.
- REINELT, G. **The Traveling Salesman:** Computational solutions for tsp applications. 1. ed. [S. l.]: Springer-Verlag Berlin Heidelberg, 1994.
- RIBAS, B. C. **Complexidade Assintótica.** Universidade Tecnológica Federal do Paraná, 2018. Disponível em: <https://www.brunoribas.com.br/apostila-eda/ribas-complexidade-assintotica.pdf>. Acesso em: 11 set. 2024.

TATOMIR, J. **Hamiltonian Circuit & Path | Differences Examples**. Study.com, 2023.
Disponível em: <https://study.com/academy/lesson/hamilton-circuits-and-hamilton-paths.html>.
Acesso em: 14 set. 2024.

WINSTON, W. L.; GOLDBERG, J. B. **Operations Research**: Applications and algorithms. 4. ed. Boston: Duxbury, 2004.

APÊNDICE A – MATRIZES DE ADJACÊNCIA

Tabela 4 – Matriz de Adjacência Ponderada por Distância em Metros

	Garagem	A1	A2	A3	A4	A5	A6	A7	B1	B2
Garagem	0	3246	2294	2803	3896	4004	4424	3688	569	895
A1	3355	0	952	1886	2979	3086	3506	2118	3041	3029
A2	2403	952	0	934	2027	2134	2554	1394	2082	2077
A3	2474	1903	951	0	1033	1141	1561	1769	2080	1695
A4	3363	2994	2042	1066	0	600	1057	1265	2969	2585
A5	3471	3102	2150	1174	636	0	513	721	3077	2692
A6	3891	3522	2570	1594	1057	515	0	770	3497	3112
A7	4099	2118	1394	1802	1265	723	770	0	3483	3320
B1	596	3045	2093	2601	3695	3802	4222	3482	0	515
B2	779	3789	2837	2146	3036	3143	3563	3771	555	0
C1	586	3047	2094	2603	3697	3804	4224	3488	369	884
C2	339	3585	2633	3142	4236	4343	4763	4027	908	1234
C3	563	3809	2857	3366	4460	4567	4987	4251	1132	1807
C4	1766	4672	3719	3480	4369	4477	4897	5105	1502	1778
C5	1351	3865	2913	2674	3563	3671	4091	4299	1088	971
C6	958	3969	3017	3070	3959	4066	4486	4410	1039	1367
C7	881	3892	2940	2763	3653	3760	4180	4388	785	1061
C8	1304	3598	2646	2407	3296	3403	3823	4031	1103	704
C9	1126	3241	2289	2050	2939	3047	3467	3675	742	347
Lixão	5338	5353	4401	3999	4888	4995	5416	5624	3654	2684

	C1	C2	C3	C4	C5	C6	C7	C8	C9	Lixão
Garagem	571	347	559	1527	1363	903	844	1275	982	5208
A1	3043	3797	3915	4374	3726	3963	3657	3638	3268	6467
A2	2091	2845	2963	3421	2774	3011	2705	2686	2334	5515
A3	2082	2915	3060	3040	2392	2629	2323	2304	1952	4142
A4	2971	3805	3950	3929	3281	3518	3212	3194	2842	5015
A5	3079	3912	4057	4036	3389	3626	3320	3301	2949	5122
A6	3499	4332	4477	4457	3809	4046	3740	3721	3369	5542
A7	3485	4540	4685	4665	4017	4254	3948	3929	3577	5750
B1	369	1037	1233	1514	1099	910	797	1011	718	3607
B2	1114	1220	1365	1344	697	934	628	609	257	3438
C1	0	784	1297	1882	1468	1155	1166	1380	1080	5428
C2	910	0	513	1432	1369	809	777	1272	1321	5049
C3	1134	495	0	830	1001	440	781	1436	1620	2924
C4	2101	1326	828	0	753	624	751	1407	1591	2299
C5	1686	1190	951	753	0	520	414	501	785	1921
C6	1293	809	431	624	560	0	340	996	1180	2717
C7	1216	777	772	751	414	306	0	690	874	2845
C8	1434	1272	1427	1407	522	996	690	0	517	2164
C9	1077	1567	1630	1610	804	1199	893	536	0	2516
Lixão	4253	3478	2980	2357	1921	2776	2903	2164	2497	0

Fonte: elaborado pelo autor.

Tabela 5 – Matriz de Adjacência Ponderada por Tempo em Segundos

	Garagem	A1	A2	A3	A4	A5	A6	A7	B1	B2
Garagem	0	417	322	392	573	554	550	550	108	191
A1	448	0	103	232	413	394	390	304	401	406
A2	345	95	0	128	309	290	287	227	297	302
A3	420	235	140	0	171	152	148	208	350	288
A4	532	396	301	158	0	150	199	259	462	400
A5	513	377	282	139	116	0	79	139	443	381
A6	525	388	293	151	128	53	0	111	454	393
A7	601	298	226	227	204	129	124	0	523	469
B1	109	388	294	363	544	525	522	521	0	115
B2	132	510	415	402	549	530	527	587	116	0
C1	143	422	327	397	578	559	556	555	94	209
C2	80	497	402	472	652	633	630	629	188	271
C3	156	573	479	548	729	710	707	706	264	351
C4	285	692	597	576	723	704	701	761	264	312
C5	279	582	487	467	613	594	591	651	259	202
C6	172	550	455	509	655	636	633	683	206	244
C7	174	551	456	476	622	603	600	660	163	211
C8	267	534	439	419	565	546	543	603	255	154
C9	225	473	378	358	505	486	482	542	195	93
Lixão	346	674	579	522	669	650	646	706	347	341
	C1	C2	C3	C4	C5	C6	C7	C8	C9	Lixão
Garagem	96	67	166	267	266	176	183	237	191	338
A1	422	506	614	597	554	554	517	525	496	708
A2	318	402	511	493	451	451	413	421	393	605
A3	371	477	540	479	436	436	399	407	378	552
A4	483	590	652	591	548	548	511	519	490	655
A5	464	571	633	572	529	529	492	500	471	636
A6	475	582	644	584	541	541	504	512	483	647
A7	544	659	721	660	617	617	580	588	559	724
B1	86	166	256	254	234	206	174	205	159	365
B2	189	189	252	191	148	148	111	119	90	302
C1	0	165	264	348	328	263	268	299	253	421
C2	176	0	98	262	307	171	191	273	270	348
C3	252	99	0	179	240	104	163	270	304	290
C4	341	277	178	0	192	112	123	230	264	185
C5	336	281	228	189	0	125	101	106	154	202
C6	229	177	100	87	132	0	55	162	196	198
C7	230	180	164	103	104	66	0	129	163	214
C8	321	286	281	221	113	178	141	0	106	248
C9	260	283	317	256	172	213	176	115	0	318
Lixão	424	360	260	179	197	194	206	239	293	0

Fonte: elaborado pelo autor.