



Universidade Estadual de Campinas
Instituto de Computação



Maria Vitória Rodrigues Oliveira

Sugestões de Rotas para Veículos de Propulsão
Humana no Contexto da Coleta Seletiva de Resíduos
Sólidos Urbanos

CAMPINAS
2022

Maria Vitória Rodrigues Oliveira

**Sugestões de Rotas para Veículos de Propulsão Humana no
Contexto da Coleta Seletiva de Resíduos Sólidos Urbanos**

Dissertação apresentada ao Instituto de
Computação da Universidade Estadual de
Campinas como parte dos requisitos para a
obtenção do título de Mestra em Ciência da
Computação.

Orientadora: Profa. Dra. Islene Calciolari Garcia

Este exemplar corresponde à versão final da
Dissertação defendida por Maria Vitória
Rodrigues Oliveira e orientada pela Profa.
Dra. Islene Calciolari Garcia.

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

OL4s Oliveira, Maria Vitória Rodrigues, 1998-
Sugestões de rotas para veículos de propulsão humana no contexto da coleta seletiva de resíduos sólidos urbanos / Maria Vitória Rodrigues Oliveira. – Campinas, SP : [s.n.], 2022.

Orientador: Islene Calciolari Garcia.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Algoritmos heurísticos. 2. Gestão integrada de resíduos sólidos. I. Garcia, Islene Calciolari, 1971-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Route suggestions for human propulsion vehicles in the context of selective collection of urban solid waste

Palavras-chave em inglês:

Heuristic algorithms

Integrated solid waste management

Área de concentração: Ciência da Computação

Titulação: Mestra em Ciência da Computação

Banca examinadora:

Islene Calciolari Garcia [Orientador]

Joel José Puga Coelho Rodrigues

Juliana Freitag Borin

Data de defesa: 04-04-2022

Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0003-3880-5747>

- Currículo Lattes do autor: <http://lattes.cnpq.br/3713105581578981>



Universidade Estadual de Campinas
Instituto de Computação



Maria Vitória Rodrigues Oliveira

Sugestões de Rotas para Veículos de Propulsão Humana no Contexto da Coleta Seletiva de Resíduos Sólidos Urbanos

Banca Examinadora:

- Profa. Dra. Islene Calciolari Garcia
Universidade Estadual de Campinas
- Prof. Dr. Joel José Puga Coelho Rodrigues
Universidade Federal do Piauí
- Profa. Dra. Juliana Freitag Borin
Universidade Estadual de Campinas

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 04 de abril de 2022

Felicidade

*“As manhãs eram felizes em São Domingos,
A estrada de terra alaranjada,
Entre seringueiras, os nossos caminhos,
Em silêncio guiava.*

*A estrada rodeada de árvores,
Com as folhas que dançavam,
Frestas de luz no chão entre as flores
Alegres de movimentos delicados.*

*No caminho, as águas límpidas dos igarapés
Convidavam na quentura escaldante amazônica,
Um banho nas sombras dos cupuaçuzeiros e outros pés.*

*As águas pediam que o tempo não passasse,
Para que não esvaíssem, perdidas,
Longe da fonte de água e felicidade.”*

(Maria Vitória R. Oliveira, 22 de setembro de 2021)

Agradecimentos

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil.

Agradeço aos meus pais, Dilza Maria e Sandro Marcelo, por todo amor, dedicação, apoio e por sempre priorizarem minha educação. Minha mãe me ensinou a ler quando eu tinha apenas 2 anos e seguiu me ensinando, não só a ler e compreender as coisas do mundo, como também a amar e respeitar todas as pessoas. Meu pai, que sempre falou “um dia você vai estudar na Unicamp”, desde criança me mostrou que nós podemos ir muito além do que nossos olhos podem ver e, não somente me apresentou a computação, como também sempre me incentivou a ser cientista. Ao meu irmão, Sandro Marcelo, pelo carinho, cuidado e incentivo. Ao meu namorado, Oscar Ciceri, pelo amor, confiança e sabedoria. Em todos os momentos, Oscar demonstrou apoio e cumplicidade, ajudando-me a encontrar soluções para os problemas que surgiam ao longo do caminho de forma leve e acolhedora. Ao meu avô, Sandoval Lima, pelo cuidado, por me ensinar a amar a natureza e preservá-la. Agradeço à minha professora orientadora, Islene C. Garcia, por todos os ensinamentos, paciência e compreensão. Sou privilegiada em ser sua aluna, pois pude me inspirar na sua trajetória e aprender muito durante o mestrado. Aos meus amigos Ítalo Silva, Luciano Teran, Jhon Cruz, Sara Mercês e Romário Silva, por sempre estarem presentes e fazerem parte da minha rede de apoio. Agradeço ao Instituto de Computação e à Unicamp pela oportunidade e compromisso com a educação, ciência e tecnologia.

Resumo

Os catadores-carrinheiros coletam materiais recicláveis e reutilizáveis em pontos de parada utilizando veículos de tração humana. Esse trabalho é essencial para o gerenciamento dos Resíduos Sólidos Urbanos. No entanto, as rotas de coleta podem ser cansativas, demoradas e muito longas para esses trabalhadores, dependendo do trajeto escolhido. Sendo assim, este trabalho propõe um algoritmo de sugestão de rotas personalizáveis chamado CATAção De resíduOs Recicláveis otimizada (CATADORA), que emprega três políticas de otimização: Política de Menor Trabalho (PMT), Política de Menor Impedância (PMI) e Política de Menor Distância (PMD). Além disso, as rotas são geradas a partir das estratégias de busca Dijkstra Bidirecional, *Shortest Path Faster Algorithm* e A-star. Os pontos de parada são ordenados de acordo com as heurísticas Vizinho mais Próximo, Inserção do mais Próximo e Inserção do mais Distante. As rotas geradas conforme a combinação das estratégias de busca, heurísticas e políticas são avaliadas de acordo com a potência empregada para empurrar o veículo nos trajetos, a distância, o tempo de percurso e o tempo de processamento do algoritmo. Utilizou-se a ferramenta livre Networkx na construção do grafo para ordenação dos pontos de parada. A ferramenta OSMnx foi implementada para desenvolvimento da rede geográfica, a partir da obtenção de dados geográficos do *Open Street Map* e dados de elevação do Banco de Dados Geomorfométricos do Brasil. Ademais, realizou-se a validação da proposta por meio de simulações computacionais utilizando o *Simulation of Urban MObility* (SUMO). Os resultados obtidos nas simulações demonstraram que o algoritmo proposto pode minimizar a potência aplicada no veículo utilizando as políticas PMT e PMI. A distância e o tempo de percurso podem ser otimizados empregando a PMD. A combinação da estratégia de busca Dijkstra bidirecional, a heurística do Vizinho mais Próximo e a PMD proporcionou o melhor desempenho do algoritmo.

Abstract

“Catadores-carrinheiros” collect recyclable and reusable materials at stopping points using human-powered vehicles. Their work is essential for managing Urban Solid Waste (USW). However, the collection routes could be tiring, time-consuming, and very long for these workers, depending on the chosen path. Therefore, this work proposes a customizable route suggestion algorithm called Optimized Recyclable Waste Collection (CATAção De resíduos Recicláveis otimizada - CATADORA) that employs three optimization policies: Less Work Policy (LWP), Less Impedance Policy (LIP), and Shortest Distance Policy (SDP). Besides, routes are generated from the bidirectional Dijkstra, Shortest Path Faster Algorithm (SPFA), and A-star search strategies. The collect points are sorted according to the Nearest Neighbor, Nearest Insertion, and Farthest Insertion heuristics. The routes generated according to the combination of search strategies, heuristics, and policies are evaluated according to the power used to push the vehicle along the paths, the distance, the travel time, and the algorithm processing time. The free tool Networkx was used to construct the graph to order the stopping points. The OSMnx tool was implemented to develop the geographic network, from obtaining geographic data from the Open Street Map and elevation data from the Brazilian Geomorphometric Database. Also, the proposal was validated through computational simulations using the Simulation of Urban Mobility (SUMO). The results obtained by the simulations showed that the proposed algorithm could minimize the power applied to the vehicle on the route using PMT and PMI policies. The route distance and time could be optimized using PMD. The combination of the bidirectional Dijkstra search strategy, the Nearest Neighbor heuristic, and the PMD provided the best performance of the algorithm.

Lista de Figuras

1.1	Catadora-carrinheira vinculada a uma associação de catadores	15
2.1	Objetivos de Desenvolvimento Sustentável [31].	20
2.2	Ilustração da coleta direta. Adaptado de [21]	22
2.3	Ilustração da coleta de dois estágios. Adaptado de [21]	23
2.4	Ilustração da coleta de três estágios. Adaptado de [21]	23
2.5	Esquema de criação de pedidos de coleta no Cataki.	24
2.6	(a) Exemplo de resumo da solicitação de uma coleta. (b) Número de catadores cadastrados até o dia 13 de setembro de 2021. (c) Perfil de um catador disponível no aplicativo.	25
2.7	Fabiano Soares, um dos catadores já conectados no aplicativo [17]	27
3.1	(a) Jogo de Hamilton e (b) Ciclo Hamiltoniano [10]	30
3.2	Exemplo do Problema do Caixeiro Viajante	31
5.1	Ilustração do processo de inserção do ponto de parada na Rede Geográfica	52
5.2	Aplicação do método dos quadrantes em um caminho irregular	53
5.3	Diagrama de atividades da construção da Rede Geográfica	54
5.4	Ilustração do grafo e rede utilizados para determinação da rota	58
5.5	Ilustração do esquema de aplicação do algoritmo	63
6.1	Área da cidade de São Paulo utilizada nos cenários de simulação	64
6.2	Área da cidade de Belém utilizada nos cenários de simulação	65
6.3	Implementação dos cenários de simulação no <i>Simulation of Urban MObility</i> (SUMO)	66
6.4	Potência aplicada no veículo nas rotas com 12 pontos de parada	69
6.5	Potência aplicada no veículo nas rotas com 18 pontos de parada	70
6.6	Potência aplicada no veículo nas rotas com 24 pontos de parada	70
6.7	Tempo e distância percorridos nas rotas com 12 pontos de parada	72
6.8	Tempo e distância percorridos nas rotas com 18 pontos de parada	72
6.9	Tempo e distância percorridos nas rotas com 24 pontos de parada	73
6.10	Tempo de processamento 12 pontos de coleta	74
6.11	Tempo de processamento 18 pontos de coleta	74
6.12	Tempo de processamento 24 pontos de coleta	75

Lista de Tabelas

3.1	Sumário de símbolos e significados	29
3.2	Sumário de símbolos e significados	32
3.3	Sumário de símbolos e significados	36
4.1	Comparação dos trabalhos relacionados	46
4.2	Comparação dos trabalhos sobre rotas de coleta de resíduos	49
5.1	Sumário de símbolos e significados	59
6.1	Parâmetros utilizados nas simulações	67

Sumário

1	Introdução	13
1.1	Motivação	15
1.2	Objetivos	16
1.3	Questões de Pesquisa	17
1.4	Publicações	17
1.5	Contribuição em Outro Trabalho	18
1.6	Estrutura da Dissertação	18
2	Coleta de Materiais Recicláveis e Reutilizáveis	19
2.1	Objetivos de Desenvolvimento Sustentável	19
2.2	O Gerenciamento de Resíduos Sólidos no Brasil	21
2.3	Procedimentos de Coleta de Materiais	22
2.4	Tecnologia para Agendamento de Coleta	24
2.5	Campanhas de Inclusão dos Catadores	26
2.6	Resumo Conclusivo	27
3	Heurísticas e Estratégias para Criação de Rotas	28
3.1	Definições e Notações	29
3.2	Otimização de Rotas	29
3.3	Problema do Caixeiro Viajante	30
3.4	Modelo Matemático do PCV	31
3.5	Heurísticas	32
3.5.1	Vizinho Mais Próximo	33
3.5.2	Inserção do Mais Próximo	34
3.5.3	Inserção do Mais Distante	34
3.6	Estratégias de Busca	35
3.6.1	Algoritmo de Dijkstra	37
3.6.2	Dijkstra Bidirecional	38
3.6.3	Algoritmo <i>A-Star</i>	39
3.6.4	<i>Shortest Path Faster Algorithm</i>	41
3.7	Resumo Conclusivo	43
4	Trabalhos Relacionados	44
4.1	Problemas de Roteamento de Veículos	44
4.2	Roteamento de Veículos para Coleta de Resíduos	46
4.3	Resumo Conclusivo	49

5	Algoritmo de Otimização de Rotas Personalizadas	50
5.1	CATADORA	50
5.2	Implementação da Rede Geográfica	51
5.3	Política de Menor Trabalho	54
5.4	Política de Menor Impedância	57
5.5	Política de Menor Distância	57
5.6	Implementação do Grafo de Ordenação	58
5.7	Algoritmos de Construção das Rotas	58
5.8	Implementação do Algoritmo em Aplicações	62
5.9	Resumo Conclusivo	63
6	Avaliação de Desempenho do Algoritmo	64
6.1	Cenários de Simulação	64
6.2	Simulador	65
6.3	Parâmetros das Simulações	67
6.4	Análise da Potência Aplicada no Veículo	68
6.5	Tempo de Trajeto e Distância Total das Rotas	71
6.6	Análise de Desempenho das Heurísticas, Estratégias e Políticas Utilizadas .	73
6.7	Resumo Conclusivo	75
7	Conclusão	76
7.1	Limitações do trabalho	77
7.2	Trabalhos Futuros	77
	Referências Bibliográficas	79

Capítulo 1

Introdução

O crescimento demográfico e dos grandes centros urbanos desencadeou a geração massiva de resíduos sólidos que causam problemas sociais, econômicos e ambientais, como a contaminação de lençóis freáticos e a escassez da água potável [70]. Problemas como esses podem ser minimizados por meio da Gestão Municipal de Resíduos Sólidos, processo que coordena ações de geração, transporte e destinação de rejeitos, materiais recicláveis e reutilizáveis. Essas ações envolvem a participação de empresas, cooperativas, catadores, geradores de resíduos e poder público. Além disso, a gestão de materiais compreende as fases de separação, coleta e destinação final ambientalmente adequada. Nessa perspectiva, a fase de coleta exige investimentos financeiros significativos, visto que demanda uma infraestrutura logística complexa [21, 52].

A primeira etapa da coleta de resíduos é chamada de coleta primária, que corresponde ao recolhimento dos materiais de casa em casa [78]. No Brasil, essa etapa geralmente é realizada por catadores autônomos ou vinculados a cooperativas. Muitas vezes, os catadores utilizam veículos de tração humana nessa fase de coleta e, por causa do carrinho característico, são chamados de catadores-carrinheiros [66, 18]. Nesse caso, a rota é realizada a pé, empurrando o veículo, e os materiais recolhidos nos pontos de coleta são armazenados no carrinho. Esses pontos são casas ou estabelecimentos onde as pessoas fornecem materiais recicláveis e reutilizáveis periodicamente, em dias e horários previamente combinados. Além disso, a comunicação entre catadores e geradores de resíduos, bem como o agendamento de coleta, é realizada informalmente ou por meio de ferramentas tecnológicas, como o aplicativo móvel Cataki [6, 23].

A utilização de veículos de propulsão humana na coleta de resíduos pode tornar o percurso cansativo e demorado, dependendo da declividade das ruas que os catadores-carrinheiros percorrem, da distância, da massa do veículo, que pode variar de 50 a 110 quilos [15, 45], e da massa dos materiais recolhidos. De acordo com Castilhos Junior et al. [18], o excesso de peso do veículo de tração humana é um dos principais problemas enfrentados pelos catadores-carrinheiros. Isso acontece porque a massa do veículo e dos materiais coletados influenciam diretamente no esforço físico empregado para empurrar o carrinho.

A aplicação de grande esforço físico durante o trajeto dos catadores-carrinheiros pode dificultar a finalização da rota de coleta, tornando-as improdutivas e prejudicando a saúde desses trabalhadores [18]. Nesse sentido, é importante criar rotas otimizadas que minimi-

zam o esforço físico empregado pelos catadores durante o trajeto, facilitando a conclusão do percurso, de modo que todos os pontos de coleta sejam visitados. No entanto, a definição das rotas de coleta não é uma tarefa trivial para os catadores-carrinheiros, visto que é preciso ter conhecimento geográfico da região, das distâncias e da localização dos pontos de recolhimento dos materiais.

Alguns trabalhos sugerem algoritmos para gerar rotas otimizadas a fim de diminuir a distância percorrida e minimizar os custos financeiros [50, 11, 39, 2]. No entanto, esses trabalhos consideram veículos automotores. Vu, Ng e Bolingbroke [78] apresentam um modelo de distribuição geográfica dos pontos de coleta e criação de rotas para diminuir custos financeiros na coleta primária de materiais recicláveis. Além disso, Das e Bhattacharyya [28] apresentam um esquema de coleta primária de Resíduos Sólidos Urbanos (RSU) que minimiza a distância das rotas. Porém, esses trabalhos não avaliam o tempo de trajeto e o esforço físico empregado para empurrar o veículo nas rotas geradas. Nessa perspectiva, Rahaman et al. [65] apresentam um algoritmo de geração de rotas acessíveis para pessoas com deficiência e idosas, considerando a inclinação das vias. Entretanto, não abrangem o contexto da coleta seletiva, que possui múltiplos pontos de coleta.

Este trabalho propõe um algoritmo chamado CATAção De resíduOs Recicláveis otimizada (CATADORA), o qual sugere rotas de coleta de materiais que otimiza o trabalho empregado para empurrar o veículo de tração humana ao longo do trajeto, evita caminhos com aclives e declives acentuados e minimiza a distância do percurso. Considera-se que a localização dos pontos de coleta, bem como a massa dos materiais que serão recolhidos em cada ponto, são pré-definidos. Nesse sentido, a ordenação dos pontos de coleta é realizada por meio de heurísticas e a construção de caminhos entre os pontos é realizada a partir de algoritmos de busca, que serão chamados de estratégias de busca ao longo deste trabalho, para não confundi-los com o algoritmo proposto.

No algoritmo, as rotas de coleta são personalizadas a partir das seguintes políticas: Política de Menor Trabalho (PMT), Política de Menor Impedância (PMI) e Política de Menor Distância (PMD). A primeira minimiza o trabalho empregado para empurrar o veículo em determinada distância, de acordo com a massa do carrinho e a declividade da rua. A PMI escolhe caminhos que não tenham aclives e declives acentuados, de modo que pode ser utilizada quando catadores-carrinheiros possuem limitações para locomoção ou outros problemas de saúde. Por fim, a política PMD minimiza a distância total do trajeto. Sendo assim, o diferencial do algoritmo CATADORA está em atender veículos de tração humana com massa variável ao longo da rota, identificando os melhores caminhos de acordo com o valor de massa do carrinho e as declividades do cenário. Além disso, a relevância social deste trabalho está em propor uma solução para melhorar a qualidade de vida dos catadores-carrinheiros. Dessa forma, é possível contribuir com o Objetivo de Desenvolvimento Sustentável 8 proposto pela Organização das Nações Unidas (ONU), chamado Trabalho Decente e Crescimento Econômico, o qual promove o crescimento econômico sustentado, inclusivo e sustentável [29].

Os cenários de avaliação do algoritmo proposto foram implementados a partir das ferramentas livres de manipulação de grafos Networkx e OSMnx [44, 12]. Ademais, os dados geográficos foram obtidos do *Open Street Map* e a altitude, do Banco de Dados Geomorfo-

métricos do Brasil [46, 77]. A validação da proposta foi realizada por meio de simulações computacionais utilizando o SUMO, um simulador *open source* que permite criar cenários, analisá-los, gerar padrões de mobilidade e simular estratégias de gerenciamento de tráfego [55].

As políticas e as heurísticas implementadas para criação das rotas foram avaliadas de acordo com a potência aplicada no veículo, distância e tempo de trajeto obtidas nas simulações. Avaliou-se também o desempenho das combinações das políticas, heurísticas e estratégias de busca por meio do tempo de processamento para gerar as rotas. Como resultado, o algoritmo CATADORA gerou rotas que minimizam a distância e o tempo de percurso utilizando a política PMD. Porém, a potência média aplicada no veículo é menor quando as políticas PMT e PMI são empregadas. Além de minimizar a potência, utilizando a política PMT, o algoritmo proporciona rotas que diminuem a distância, em comparação com a PMI. Ademais, o melhor desempenho computacional foi alcançado utilizando a estratégia de busca Dijkstra bidirecional e a heurística Vizinho mais Próximo (VP).

1.1 Motivação

Este trabalho foi motivado pela necessidade de melhorar a qualidade de vida e trabalho dos catadores-carrinheiros, que apesar de desempenharem um trabalho essencial para a sociedade, são muitas vezes invisibilizados. Nessa perspectiva, o cotidiano e os problemas que esses trabalhadores enfrentam foram observados a partir do projeto Destino Sustentável [7], que tem como objetivo conectar os recicladores, indústrias, geradores de resíduos e poder público. Além disso, esse projeto trabalha na divulgação de informações sobre meio ambiente, reciclagem, sustentabilidade e a legislação relacionada à gestão de RSU.

A partir do projeto Destino Sustentável, foi possível estudar o contexto ao qual os catadores estão inseridos no Brasil. De acordo com o censo do Instituto Brasileiro de Geografia e Estatística (IBGE) (2010), existem 387.910 catadores de materiais recicláveis no Brasil [74]. Porém, o Movimento Nacional dos Catadores de Materiais Recicláveis estima que são 800 mil coletores [58]. A Figura 1.1 apresenta uma catadora-carrinheira na cidade de Belém, Pará.



Figura 1.1: Catadora-carrinheira vinculada a uma associação de catadores

A legislação brasileira estabeleceu a responsabilidade compartilhada da destinação

adequada de RSU entre cooperativas, que são entidades sem fins lucrativos formadas por catadores [71], poder público e empresas privadas [19] a partir da Política Nacional de Resíduos Sólidos (PNRS) de 2010 [14]. Apesar dos avanços na legislação, ainda há diversos desafios para a inclusão dos catadores de materiais acontecer na prática.

Projetos como o Destino Sustentável e Cataki apostaram na ideia de conectar os geradores de resíduos com os coletores e recicladores por meio de um aplicativo móvel. O aplicativo Cataki permite aos catadores-carrinheiros obterem informações dos pontos de coleta, localização geográfica, horários e dias da semana disponíveis. Nessa perspectiva, esses trabalhadores podem decidir a rota antes de sair para a coleta, visto que é possível identificar os materiais que serão recolhidos, a estimativa da quantidade e a localização dos pontos de parada. Esse aplicativo também possibilita estimar os valores de massa dos materiais que serão coletados em cada ponto de parada. Atualmente, existem 3603 catadores cadastrados no aplicativo espalhados por todas as regiões do Brasil [6].

O agendamento da coleta de materiais recicláveis e reutilizáveis facilita a criação de rotas para catadores-carrinheiros, uma vez que possibilita conhecer previamente os pontos de coleta e os materiais que serão recolhidos. Sendo assim, torna-se possível criar uma ferramenta de sugestão de rotas otimizadas, considerando a variação da massa do veículo e a localização dos pontos de coleta. Além disso, os dados geográficos abertos disponíveis no *Open Street Map* e no Banco de Dados Geomorfométricos do Brasil permitem utilizar informações sobre a distância e inclinação das vias. Nessa perspectiva, as rotas otimizadas podem melhorar o dia-a-dia dos trabalhadores da coleta seletiva e o agendamento da coleta realizado por meio do Cataki contribui para a valorização econômica e social dos catadores.

1.2 Objetivos

Este trabalho tem como objetivo propor um algoritmo de otimização de rotas individuais chamado CATADORA, considerando a utilização de veículos de propulsão humana para minimizar a distância dos trajetos, o esforço físico empregado para empurrar o veículo e evitar declividades acentuadas.

Para alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

1. *Desenvolver políticas de personalização de rotas:* as políticas devem possibilitar personalizar as rotas de acordo com o objetivo, o qual pode ser de minimizar a potência aplicada no veículo, a distância e o tempo, bem como identificar caminhos que não passam por declividades acentuadas.
2. *Definir um procedimento para reconhecer a declividade do cenário:* as informações sobre os aclives e declives da área geográfica são utilizadas na geração de rotas que tem com objetivo minimizar o esforço físico ou evitar caminhos com subidas e descidas.
3. *Ordenar a visita dos pontos de parada por meio de métodos heurísticos:* a ordenação dos pontos de coleta deve ser realizada em tempo de processamento viável para aplicações reais.

4. *Estabelecer mecanismos de busca de caminhos entre pontos de parada:* os caminhos entre pontos de parada são definidos a partir do mapa geográfico, utilizando as políticas como parâmetro de escolha dos trajetos.

1.3 Questões de Pesquisa

A utilização de veículos de tração humana na coleta de resíduos está associada a rotas cansativas e longas para os catadores-carrinheiros. Sendo assim, este trabalho foi norteado pelas seguintes questões de pesquisa:

1. Qual o impacto de evitar aclives e declives acentuados na otimização da potência empregada para empurrar o veículo de tração humana?
2. Como a minimização da distância das rotas afeta a potência empregada para empurrar o veículo ao longo do percurso?
3. Como as heurísticas utilizadas para ordenação dos pontos de coleta influenciam na otimização da potência empregada para empurrar o veículo de tração humana?
4. De que forma a otimização do trabalho empregado para empurrar os veículos nas rotas influencia na distância total dos percursos?
5. O quão computacionalmente eficientes são as combinações de estratégias de busca, heurísticas e políticas empregadas no algoritmo para solucionar o problema?

As questões de pesquisa 1, 2 e 3 foram respondidas na Seção 6.4; a questão 4 foi respondida na Seção 6.5 e a questão 5, na Seção 6.6.

1.4 Publicações

Os resultados obtidos nesse trabalho foram publicados em dois artigos, descritos a seguir.

- **OLIVEIRA, M. V. R.**, GARCIA, I. C. Sugestões de Rotas Personalizadas para Carrinheiros na Coleta Seletiva de Materiais Recicláveis. XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2021, Ambiente virtual.
- **OLIVEIRA, M. V. R.**, GARCIA, I. C. Performance Evaluation of Route Suggestion Approaches for Primary Waste Collection. 3rd International Workshop on Urban Computing, 2021, Ambiente virtual.

O primeiro artigo apresenta o algoritmo de criação de rotas para catadores-carrinheiros e as políticas de minimização da distância e esforço físico. O segundo artigo compreende uma avaliação de desempenho de diferentes heurísticas e algoritmos de busca aplicados no algoritmo de criação de rotas.

1.5 Contribuição em Outro Trabalho

Durante o desenvolvimento da pesquisa, realizou-se uma contribuição em um trabalho que propõe um sistema baseado em um controlador fuzzy e redes neurais para encontrar a velocidade instantânea que reduz o consumo de combustível de um veículo.

- Sanches, M. F., **Oliveira, M. V. R.**, Ciceri, O. J., Ladeira, L. Z., Garcia, I. C., da Fonseca, N. L., Villas, L. A. EFIS-Ecological Fuel-consumption Intelligent System. 17th International Conference on Distributed Computing in Sensor Systems, 2021, Ambiente virtual.

1.6 Estrutura da Dissertação

Esta dissertação está estruturada em 7 capítulos. O Capítulo 2 apresenta o contexto socio-ambiental deste trabalho, com informações sobre o desenvolvimento sustentável, gerenciamento de resíduos sólidos recicláveis e reutilizáveis e tecnologia para agendamento de coleta. Além disso, o Capítulo 3 discorre sobre a fundamentação teórica utilizada no desenvolvimento do trabalho de acordo com os métodos heurísticos de criação de rotas e algoritmos de busca para definição de caminhos da literatura. O Capítulo 4 apresenta os trabalhos relacionados à otimização no contexto do roteamento de veículos e da criação de rotas para coleta de resíduos. O Capítulo 5 apresenta o algoritmo CATADORA, para otimização de rotas personalizadas de catadores-carrinheiros. O Capítulo 6 demonstra os resultados das simulações de acordo com as políticas propostas, estratégias de busca dos caminhos e heurísticas de ordenação implementadas. Por fim, o Capítulo 7 apresenta as conclusões, publicações e trabalhos futuros.

Capítulo 2

Coleta de Materiais Recicláveis e Reutilizáveis

Este capítulo apresenta informações sobre os Objetivos de Desenvolvimento Sustentável da ONU, dados sobre o gerenciamento de resíduos sólidos recicláveis no Brasil e uma tecnologia para o agendamento da coleta de materiais chamada Cataki.

2.1 Objetivos de Desenvolvimento Sustentável

O termo sustentabilidade ganhou notoriedade mundial em 1992, durante a Conferência das Nações Unidas para o Meio Ambiente e o Desenvolvimento, também chamada de Cúpula da Terra, Eco 92 ou Rio 92, que ocorreu na cidade do Rio de Janeiro. O debate sobre a sustentabilidade foi pautado no objetivo de conciliar o desenvolvimento socioeconômico com a conservação e preservação dos ecossistemas da Terra. Nessa ocasião, formulou-se a Agenda 21, a fim de nortear a elaboração de políticas públicas voltadas para a preservação ambiental [53].

O progresso feito desde a formulação da Agenda 21 foi analisado em 2012, 20 anos depois da Rio 92, na conferência Rio +20. Além das análises dos avanços, a Rio +20 foi realizada para assegurar o comprometimento com o desenvolvimento sustentável. Além disso, desafios primordiais da sociedade foram discutidos, dentre eles, a erradicação da pobreza, inclusão social e economia verde [53]. Esses desafios foram descritos de forma mais aprofundada durante a elaboração da Agenda 2030, na sede das Nações Unidas, em Nova York, de 25 a 27 de setembro de 2015 [29].

A Agenda 2030 é descrita como “um plano de ação para as pessoas, o planeta e a prosperidade” [29]. O desenvolvimento dessa Agenda contou com a participação de representantes dos 193 Estados-membros da ONU. Além disso, os países se comprometeram a promover o desenvolvimento sustentável a partir de 17 Objetivos de Desenvolvimento Sustentável (ODS), ilustrados na Figura 2.1, e 169 metas. Dentre os objetivos propostos, destacam-se três que serão abordados ao longo deste capítulo: Trabalho Decente e Crescimento Econômico (Objetivo 8); Cidades e Comunidades Sustentáveis (Objetivo 11); Consumo e Produção Responsáveis (Objetivo 12).

O Objetivo de Desenvolvimento Sustentável 8 caracteriza-se por “promover o cresci-



Figura 2.1: Objetivos de Desenvolvimento Sustentável [31].

mento econômico sustentado, inclusivo e sustentável, emprego pleno e produtivo e trabalho decente para todos" [29]. Para atingir esse objetivo, a Agenda 2030 estabeleceu metas que direcionam a implementação de políticas que favoreçam o crescimento econômico, respeitando os trabalhadores e o meio ambiente. Dentre as metas estabelecidas, destaca-se melhorar a eficiência dos recursos globais no consumo e na produção, além de se empenhar para dissociar o crescimento econômico da degradação ambiental.

O Objetivo 11 é definido por “tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis” [29]. Desta maneira, destaca-se a meta de reduzir o impacto ambiental negativo per capita das cidades até 2030, dando atenção especial à gestão de resíduos municipais e outros fatores. Nesse sentido, as transformações das cidades visando a sustentabilidade devem ser uma prioridade nas políticas públicas, bem como o gerenciamento de resíduos sólidos eficiente e a disposição final dos materiais de modo ambientalmente adequado.

O gerenciamento de resíduos sólidos é um desafio relacionado ao planejamento urbano e gestão pública. Trata-se de um complexo sistema que envolve várias etapas e diversos segmentos da sociedade. Dentre as etapas principais, é possível mencionar a coleta dos resíduos, separação de acordo com o tipo de material, transporte, tratamento e disposição final [27]. Além disso, a gestão de resíduos é essencial para o desenvolvimento social, ambiental e econômico das cidades. Nessa perspectiva, o Objetivo 12 promove assegurar padrões de produção e de consumo sustentáveis [29].

A Agenda 2030 estabeleceu como uma das metas associadas ao Objetivo 12 minimizar a geração de resíduos por meio da prevenção, redução, reciclagem e reuso. A prevenção e redução da geração dos resíduos sólidos estão associadas à recusa de produtos que não poderão ser reaproveitados e ao consumo consciente. Ademais, a reciclagem é um conjunto de operações que envolvem alterações físico-químicas ou biológicas, realizadas com o objetivo de reintroduzir os materiais recicláveis no processo de produção. Esse processo

é mais complexo do que o reuso, que corresponde ao reaproveitamento dos materiais sem alterações físico-químicas ou biológicas [29].

A prevenção, redução, reciclagem e reuso representam princípios fundamentais para a gestão eficiente dos materiais. Além de enfatizar estes princípios, a Agenda 2030 e as conferências internacionais para o meio ambiente evidenciaram a necessidade de ações que possibilitam realizar o desenvolvimento sustentável e a preservação ambiental por meio da gestão eficiente dos resíduos sólidos. Sendo assim, é necessário analisar o contexto social, ambiental e econômico dos países para diagnosticar os problemas associados ao descarte dos materiais, bem como estabelecer políticas públicas para solucioná-los gradativamente. O Brasil, apesar de ter avançado após a criação da Política Nacional de Resíduos Sólidos, apresenta grandes desafios para colocar esta política em prática [29].

2.2 O Gerenciamento de Resíduos Sólidos no Brasil

A PNRS, Lei Nº 12.305 de 5 de agosto de 2010, regulamentou a gestão de RSU no Brasil, estabelecendo diretrizes para a destinação adequada de rejeitos, materiais recicláveis e reutilizáveis. Nesse sentido, essa política define a gestão integrada de RSU como um conjunto de ações com o objetivo de encontrar soluções para a destinação dos resíduos, considerando as dimensões política, econômica, ambiental, cultural e social [14]. A regulamentação da PNRS foi um avanço para o país, uma vez que mostrou as ações que devem ser tomadas para realizar o gerenciamento dos resíduos sólidos de forma eficiente e adequada. No entanto, o Brasil ainda enfrenta diversos problemas e muitos municípios ainda não se adequaram à lei [30].

De acordo com o Panorama dos Resíduos Sólidos no Brasil 2020 [1], constata-se que a geração de RSU aumentou 19%, entre 2010 e 2019. Considerando a geração de resíduos per capita, houve aumento de 348 quilos por ano, para 379 quilos por ano, no mesmo período. A análise regional demonstra que os resíduos gerados pela região sudeste do país correspondem a aproximadamente 50% do que foi gerado em todo o Brasil. Nessa perspectiva, demonstra-se dificuldades de implementação dos princípios de recusar e reduzir a produção de resíduos.

No período de 2010 a 2019, a cobertura de coleta passou de 88% para 92% do RSU gerado. Demonstrando, assim, que a PNRS proporcionou avanços importantes. Ademais, os municípios que possuem iniciativas de coleta seletiva aumentaram de 56,6% em 2010, para 73,1%, em 2019. Destaca-se, porém, que essas iniciativas de coleta seletiva muitas vezes não abrangem a totalidade da área urbana [1]. Apesar dos avanços, os índices de reciclagem permanecem em patamares inferiores a 4%. O baixo índice reflete, dentre muitos problemas, as dificuldades logísticas de encaminhamento dos materiais para a reciclagem.

A PNRS enfatizou a importância da logística reversa, caracterizada por um conjunto de ações que possibilitam a coleta e restituição dos resíduos ao setor empresarial, para que tenham disposição final ambientalmente adequada, ou sejam reaproveitados. De modo geral, a responsabilidade de gestão dos resíduos é compartilhada entre os geradores (que incluem consumidores e cidadãos), poder público e pessoas físicas ou jurídicas, do setor

público ou privado, que desenvolvam ações relacionadas ao gerenciamento dos materiais. Na logística reversa, as responsabilidades das organizações não governamentais, catadores, associações, cooperativas, empresas e indústrias ficam mais evidentes, uma vez que desempenham o serviço de coleta, transporte e encaminhamento dos resíduos para as indústrias de reciclagem [14].

A inclusão dos catadores nos processos de gestão de materiais recicláveis e reutilizáveis foi determinada pela PNRS. Para que a reciclagem ou o reuso dos materiais ocorram, os catadores precisam ter condições favoráveis para realizarem a coleta, triagem e transporte dos materiais. Apesar de oferecerem um serviço essencial para as pessoas e ao meio ambiente, esses trabalhadores enfrentam a exclusão social e uma série de preconceitos. Por isso, muitos catadores tem se organizado economicamente e socialmente, a partir de ações coletivas, buscando melhores condições de trabalho, reconhecimento social e direito à cidadania [14].

A organização por meio de associação de catadores e cooperativas de reciclagem geralmente promove melhorias na produtividade e na qualidade de vida dos associados, possibilitando mais trabalho e renda para os catadores. De acordo com a Política Nacional de Cooperativismo, Lei Nº 5.764, de 16 de dezembro de 1971, cooperativas são sociedades sem fins lucrativos constituídas para prestar serviços aos associados [13]. Sendo assim, a PNRS tem como instrumento o incentivo à criação de cooperativas ou outras formas de associações de catadores [14].

Nas cooperativas de reciclagem, os catadores somam forças por meio do trabalho coletivo. Nessa perspectiva, cooperativas consideradas de alta eficiência recolhem 1800 quilos de materiais recolhidos por catador(a)/mês ou mais [73]. Sendo assim, diferentes procedimentos de coleta podem ser empregados de acordo com a infraestrutura disponível, como veículos de grande porte, e a média de materiais que costumam coletar.

2.3 Procedimentos de Coleta de Materiais

Existem três metodologias principais de coleta de materiais recicláveis em ambientes urbanos: a coleta direta, a de dois estágios e a de três estágios. Na coleta direta, utiliza-se um único veículo automotor, o qual realiza o recolhimento dos materiais nos pontos de coleta. Os pontos de coleta ou pontos de parada podem ser residências e estabelecimentos comerciais. Depois, o veículo transporta os resíduos até o local de disposição final [21]. A Figura 2.2 apresenta a ilustração de um exemplo de coleta direta.

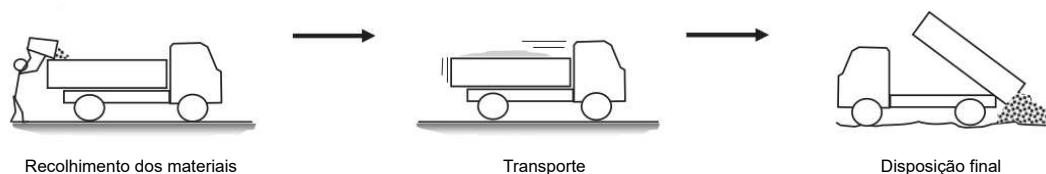


Figura 2.2: Ilustração da coleta direta. Adaptado de [21]

Na coleta de dois estágios, utiliza-se inicialmente um veículo automotor para o recolhimento dos materiais nos pontos de parada. Em seguida, os resíduos são transferidos para

um veículo de grande porte, que realiza o transporte até o local de disposição final [21]. A Figura 2.3 apresenta a ilustração de um exemplo de coleta de dois estágios.

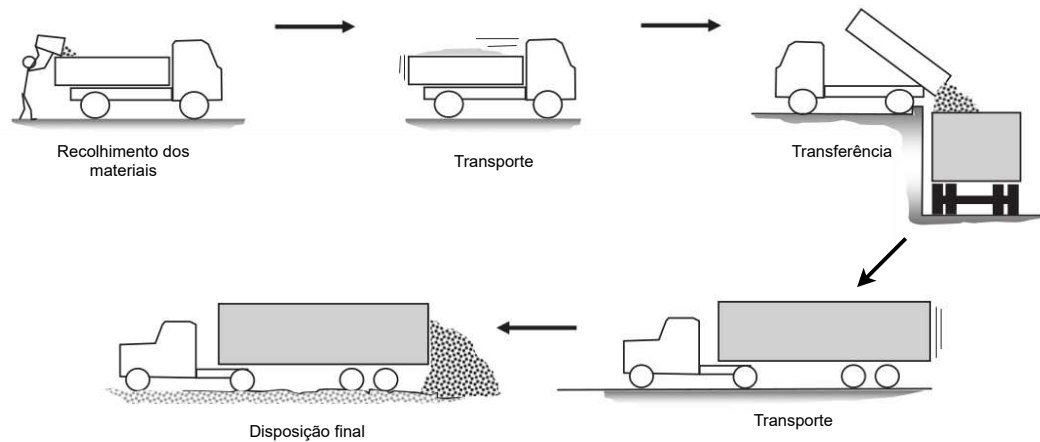


Figura 2.3: Ilustração da coleta de dois estágios. Adaptado de [21]

Na coleta de três estágios, utiliza-se veículo(s) de tração humana para armazenar os resíduos durante o recolhimento de materiais nos pontos de parada, etapa também chamada de pré-coleta ou coleta primária [78]. A utilização de veículos de tração humana possibilita que os catadores-carrinheiros passem por lugares onde não podem trafegar veículos de grande porte, como vias estreitas, ruas com casas muito antigas e áreas com frequente congestionamento de veículos [21]. No entanto, por questões de segurança, vias onde trafegam carros em alta velocidade são evitadas quando possível.

Após a coleta primária, um caminhão transporta os materiais coletados até o galpão. Depois, um veículo de grande porte leva os materiais até o local de disposição final [21]. A Figura 2.4 apresenta a ilustração de um exemplo de coleta de três estágios.

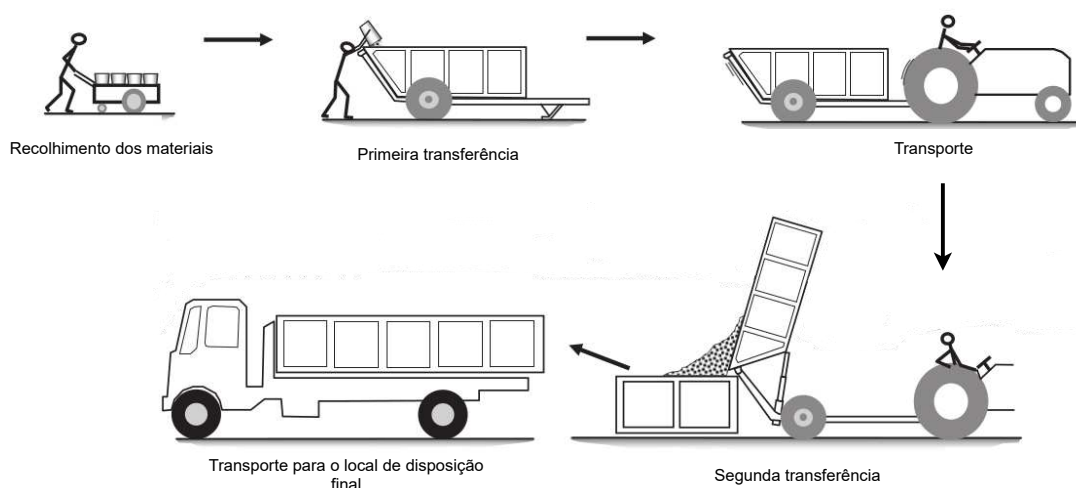


Figura 2.4: Ilustração da coleta de três estágios. Adaptado de [21]

As organizações responsáveis pelas fases da coleta de três estágios muitas vezes são diferentes. Por exemplo, cooperativas de reciclagem podem realizar a coleta primária, enquanto um serviço municipal ou regional realiza a segunda e terceira etapa. É possível

citar também o exemplo dos catadores autônomos que fazem a coleta primária e vendem os materiais para as cooperativas, que realizam a segunda e terceira fase. Sendo assim, adicionar mais agentes responsáveis e mais fases de coleta aumenta a complexidade do sistema, mas distribui os custos e os esforços [21].

Os catadores que utilizam veículos de tração humana estão inseridos na coleta primária da metodologia de três estágios. Nesse caso, pode ser mais eficiente trabalhar com coletas agendadas para evitar longas caminhadas sem ter a certeza dos materiais que serão coletados. Para isso, é preciso identificar a localização dos pontos de coleta previamente. Isso é possível por meio da tecnologia, que possibilita que os catadores verifiquem os materiais disponíveis de uma determinada região.

2.4 Tecnologia para Agendamento de Coleta

A identificação prévia dos pontos de parada é importante para que os catadores possam criar rotas de coleta eficientes. Nesse sentido, a falta de comunicação entre as pessoas que querem descartar e os catadores impossibilita a identificação dos pontos de parada e, conseqüentemente, dificulta a realização da coleta. Sendo assim, muitas vezes esses profissionais precisam procurar os materiais recicláveis dentro dos sacos de lixo que ficam na frente da casa das pessoas, causando desconforto e perigos para a saúde.

Os agentes responsáveis pela gestão de RSU devem estar integrados para aumentar a quantidade de resíduos destinados à reciclagem e reutilização nos municípios brasileiros. Esses agentes correspondem às pessoas que descartam materiais, assim como os catadores, indústrias, cooperativas, associações e artesãos. Nessa perspectiva, o movimento Pimp My Carroça promove a inclusão dos catadores de materiais recicláveis e reutilizáveis no Brasil desde 2012, por meio da tecnologia, arte, conscientização e participação coletiva.

O movimento Pimp My Carroça desenvolveu e lançou o aplicativo móvel Cataki em 2017 para possibilitar o agendamento de coleta dos materiais, facilitando a comunicação entre as pessoas que descartam resíduos e os catadores [6]. O Cataki é considerado o “LinkedIn” dos profissionais de coleta, pois amplia a divulgação dos serviços de recolhimento dos materiais. Sendo assim, esse aplicativo impacta positivamente no reconhecimento e remuneração justa dos catadores. Além disso, está disponível para os *smartphones* com sistemas Android e iOS.

A funcionalidade principal do Cataki é a criação de pedidos de coleta, na qual os usuários podem agendar o recolhimento dos materiais com catadores. Nessa perspectiva, a Figura 2.5 apresenta um esquema resumido da criação de pedidos de coleta no Cataki.



Figura 2.5: Esquema de criação de pedidos de coleta no Cataki.

De acordo com a Figura 2.5, a aplicação possibilita a integração de dois agentes essenciais na gestão de RSU, os catadores e os gerados de resíduos. Sendo assim, as pessoas que desejam criar um pedido de coleta no Cataki devem informar os seguintes dados: tipos dos materiais que desejam descartar (vidro, plástico, latas, papel, dentre outros), condição dos resíduos (ótima, boa ou ruim), horários para recebimento do(a) catador(a), dia da semana, preferência de gênero do(a) catador(a) e a frequência de recolhimento dos materiais, que pode ser única ou frequente (semanal, quinzenal ou mensal). Sendo assim, a Figura 2.6(a) apresenta o resumo da solicitação de uma coleta para exemplificar o processo de agendamento.

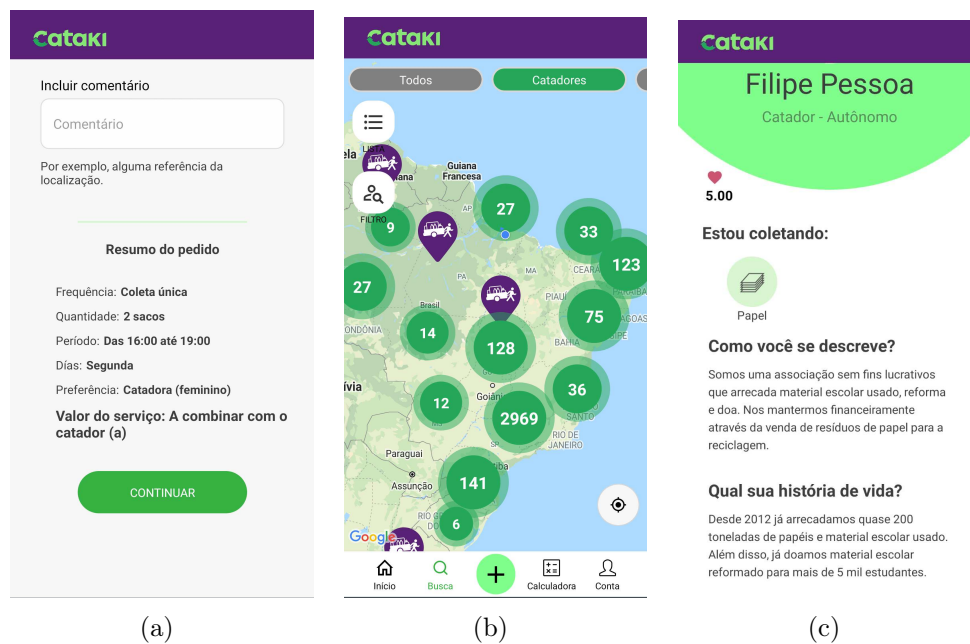


Figura 2.6: (a) Exemplo de resumo da solicitação de uma coleta. (b) Número de catadores cadastrados até o dia 13 de setembro de 2021. (c) Perfil de um catador disponível no aplicativo.

Na Figura 2.6(a), é possível verificar que o resumo do pedido de coleta inclui informações básicas necessárias para realizar o agendamento. Além disso, o valor do serviço de recolhimento dos materiais deve ser combinado entre a pessoa que solicita a coleta e o(a) catador(a), de modo que o aplicativo não cobra nenhuma taxa. O pagamento dos catadores é incentivado, visto que estes profissionais estão prestando um serviço muito importante e pouco remunerado na gestão de RSU.

A conclusão da solicitação de coleta no Cataki se dá após o aplicativo indicar um(a) catador(a) geograficamente próximo para realizar a coleta. É possível ter vários catadores disponíveis para realizarem a coleta, dependendo da região. No Brasil, existem 3603 catadores cadastrados no aplicativo, distribuídos geograficamente de acordo com a Figura 2.6(b).

Na Figura 2.6(b), identifica-se que a região sudeste, especialmente o estado de São Paulo, concentra grande parte dos catadores cadastrados no Cataki. Esse cadastro pode ser realizado facilmente pelo aplicativo ou pelo site, de forma gratuita [6]. Além disso, na criação do perfil dos catadores é possível identificar o tipo de material que coletam,

fazer uma auto-descrição e contar um pouco sobre a história de vida. Nessa perspectiva, a Figura 2.6(c) apresenta o perfil de um catador que coleta papel e reforma materiais escolares para doar aos estudantes.

O aplicativo Cataki tem inúmeras funcionalidades, além da criação de pedido de coleta. Por exemplo, é possível cadastrar compradores de materiais, os quais podem informar o preço de compra dos resíduos por quilo e buscar compradores geograficamente próximos ao usuário. Além disso, o aplicativo permite verificar no mapa os catadores (autônomos, líderes de cooperativas e cooperados), cooperativas, ferro velho, pontos de doação, pontos de entrega voluntária (públicos e privados) e indústrias. Outra funcionalidade importante é a classificação dos usuários por meio de avaliações e comentários, a partir das experiências de coleta. Ademais, a navegação no aplicativo é intuitiva e rápida.

O desenvolvimento do Cataki pode ser visto como um passo a frente na integração dos agentes que fazem parte do ciclo de vida dos materiais. No entanto, para que essa tecnologia faça a diferença nos índices de materiais reciclados no Brasil é preciso conscientizar os geradores de resíduos sobre as suas responsabilidades quanto aos resíduos gerados e a importância do reaproveitamento dos materiais. Nessa perspectiva, entidades de divulgação de conteúdos sobre meio ambiente, como o Instituto GEA e o Destino Sustentável, fornecem recursos essenciais para a conscientização das pessoas.

O Instituto GEA realizou projetos de capacitação de coleta de resíduos eletrônicos e aerossóis para as cooperativas, hortas comunitárias, implantação de coleta seletiva em um *shopping* de São Paulo, dentre outros projetos. Ademais, o instituto ministrou diversos cursos, como por exemplo reciclagem de resíduos eletrônicos, alimentação sustentável, práticas sustentáveis dentro de casa e empreendedorismo na área da reciclagem [63]. No ambiente virtual, o *site* do instituto fornece manuais de como fazer a coleta seletiva em condomínios, na escola e construção de hortas em casa [8].

Em consonância com as ações do Instituto GEA, o projeto Destino Sustentável promove a educação ambiental a partir de matérias divulgadas *online* para divulgação de conteúdos sobre sustentabilidade e reciclagem, escritas por voluntários. Dentre os conteúdos abordados nas matérias, é possível mencionar: o que são microplásticos, descarte de materiais plásticos no oceano, benefícios do reaproveitamento da matéria orgânica, o ciclo dos metais pesados, reciclagem de lixo eletrônico, o problema das fibras de amianto, dentre outros [7]. Dessa forma, a associação do aplicativo móvel Cataki com a conscientização sobre a importância da reciclagem possibilita que cada vez mais pessoas façam sua parte na gestão de RSU.

2.5 Campanhas de Inclusão dos Catadores

O movimento Pimp My Carroça se uniu à página Quebrando o Tabu em 2019 para lançar uma campanha de doações de celulares para os catadores. O objetivo da campanha era arrecadar 500 celulares e foi um sucesso. Muitas pessoas enviaram *smartphones* pelos correios ou entregaram pessoalmente, nas centrais do Pimp My Carroça. Além disso, a Motorola fez doação de celulares. Os *smartphones* arrecadados foram distribuídos para os catadores que ainda não tinham acesso à essa tecnologia [17].



Figura 2.7: Fabiano Soares, um dos catadores já conectados no aplicativo [17]

A campanha de doação de celulares possibilitou expandir o acesso ao aplicativo Cataki em 2019. Com o início da pandemia em 2020, o Pimp My Carroça teve como foco ações para garantir segurança, saúde e renda para os catadores. Sendo assim, lançou-se a campanha de financiamento coletivo “Renda Mínima para os Catadores”, que arrecadou mais de 1,5 milhões de reais. Dessa forma, cada um dos 2117 catadores beneficiados recebeu 650 reais [16].

Além do repasse financeiro, o Pimp My Carroça providenciou alimentos, equipamentos de proteção individual, cestas básicas e outros kits. Como resultado de todas as ações desempenhadas durante a pandemia, no ano de 2020 foram realizados 2.188 novos cadastros de catadores autônomos no Cataki e 43.357 novos downloads do aplicativo, de acordo com o Relatório de Atividades 2020 do Pimp My Carroça. Além disso, no início do ano o aplicativo estava presente em 355 cidades e, no final de 2020, estava presente em mil cidades [16].

2.6 Resumo Conclusivo

O gerenciamento de resíduos sólidos é uma das prioridades para o desenvolvimento sustentável do Brasil e do mundo. Apesar dos avanços na legislação brasileira desde a regulamentação da PNRS em 2010, o país enfrenta muitos desafios para aumentar a quantidade de materiais destinados à reciclagem e reutilização. Dentre estes desafios, estão as dificuldades enfrentadas por catadores autônomos ou que trabalham coletivamente, para realizar a coleta dos resíduos nos municípios. A coleta acontece a partir da comunicação entre os geradores de resíduos, que precisam descartar os materiais de forma ambientalmente adequada, e os catadores. Sendo assim, o movimento Pimp My Carroça desenvolveu o aplicativo Cataki para facilitar a comunicação entre geradores e coletores de resíduos por meio do agendamento de coletas. Possibilitando, assim, a identificação prévia dos pontos de parada para criação de rotas.

Capítulo 3

Heurísticas e Estratégias para Criação de Rotas

Os sistemas de coleta inteligente são caracterizados pela utilização de tecnologias que permitem às pessoas solicitarem e agendarem a coleta de resíduos recicláveis e reutilizáveis por meio de aplicativos. Diferentemente da coleta informal, na qual os catadores não conseguem prever os locais onde têm materiais disponíveis e, muitas vezes, precisam olhar dentro das lixeiras na rua para encontrarem resíduos recicláveis, na coleta inteligente utiliza-se tecnologias para organizar, padronizar e monitorar os procedimentos de coleta [82]. Sendo assim, torna-se mais fácil identificar previamente o tempo de percurso, os pontos de coleta e definir áreas de abrangência geográfica para os catadores.

Na coleta informal, geralmente emprega-se rotas empíricas para realizar o recolhimento de resíduos de casa em casa. Nesse caso, os catadores criam caminhos baseados na experiência adquirida no trabalho, sobre a topologia da região e os locais de coleta. No entanto, a organização e identificação prévia dos pontos de parada por meio dos sistemas de coleta inteligente permite a criação de rotas otimizadas. Sendo assim, é possível construir rotas minimizando a distância, o tempo de percurso ou o esforço físico empregado para empurrar o veículo durante o trajeto.

Para gerar rotas otimizadas, é preciso definir a ordem de visita dos pontos de coleta e os caminhos entre esses pontos. A ordenação e a definição de caminhos entre os pontos podem ser realizadas com o auxílio de grafos, os quais são objetos matemáticos que podem ser aplicados em sistemas onde as relações entre os objetos desempenham um papel dominante [57]. No grafo utilizado para ordenação dos pontos de coleta, cada ponto é representado por um nó, o qual é conectado com todos os outros nós/pontos de coleta. Na definição de caminhos entre pontos de parada, o grafo se comporta como uma rede de ruas, na qual os links representam segmentos de ruas e os nós, as junções entre estes segmentos.

Neste trabalho, aplicou-se métodos heurísticos para definir a ordem de visita dos pontos de coleta, visto que esses métodos podem ser mais eficientes em relação ao tempo de processamento, em comparação com métodos exatos. Além disso, utilizou-se algoritmos de busca, que foram denominados de estratégias de busca, para criar caminhos entre os pontos de coleta.

Este capítulo apresenta as estratégias de busca Dijkstra Bidirecional, A-star e *Shortest*

Path Faster Algorithm (SPFA), além das heurísticas Vizinho mais Próximo (VP), Inserção do mais Próximo (IP) e Inserção do mais Distante (ID). Para descrever as estratégias de busca e as heurísticas, utiliza-se definições e notações de teoria dos grafos.

3.1 Definições e Notações

Um grafo pode ser definido como $G = (V, L)$, onde $V = \{v_1, \dots, v_n\}$ é o conjunto de vértices, também chamados de nós. Além disso, $L = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$ é o conjunto de arestas, também chamadas de *links* ou arcos, que conectam dois nós diretamente. Em grafos orientados, dizemos que v_i e v_j são adjacentes se $(v_i, v_j) \in L$. No entanto, em grafos não orientados, v_i e v_j são adjacentes se $(v_i, v_j) \in L$ ou $(v_j, v_i) \in L$. Ademais, assume-se que G é um grafo ponderado, no qual $C = [c(v_i, v_j)]$ é uma $n * m$ matriz associada ao conjunto de links L , onde $c(v_i, v_j)$ é o peso da aresta (v_i, v_j) . Sendo assim, o custo total de um caminho S é dado pela soma dos pesos das arestas que fazem parte do caminho [43]. A Tabela 3.1 apresenta um sumário dos símbolos relacionados às definições de grafos e os respectivos significados.

Tabela 3.1: Sumário de símbolos e significados

Símbolo	Significado
G	Grafo
L	Conjunto de links do grafo
V	Conjunto de vértices do grafo
C	Matriz com os valores de custo entre dois vértices
c	Custo do link que conecta dois vértices
C_{total}	Custo total
S	Conjunto de vértices que fazem parte da rota

Os grafos são amplamente utilizados para a criação e otimização de rotas. Nesse sentido, as notações e símbolos apresentados na Tabela 3.1 serão utilizados ao longo deste capítulo para descrever algoritmos de ordenação de nós e criação de caminhos.

3.2 Otimização de Rotas

Em 1857, o problema de definição de rotas foi abordado pelo matemático irlandês William Rowan Hamilton. Ele propôs o jogo *Around the World*, composto por um dodecaedro, que está ilustrado na Figura 3.1(a). No jogo de Hamilton, os vértices representam cidades e o objetivo é estabelecer uma rota para percorrer todos os nós, como demonstrado na Figura 3.1(b). Para isso, é preciso percorrer cada vértice apenas uma vez e voltar para o nó inicial, encerrando a rota que ficou conhecida como Ciclo Hamiltoniano [10].

Os Caminhos Hamiltonianos são construídos de forma semelhante aos ciclos, pois visitam os vértices de um grafo apenas uma vez. No entanto, diferem-se em não voltar para o nó de início após percorrer todos os vértices. Além disso, um grafo é considerado Hamiltoniano-conectado se, para cada par de vértices v_i e v_j do grafo, existe um caminho

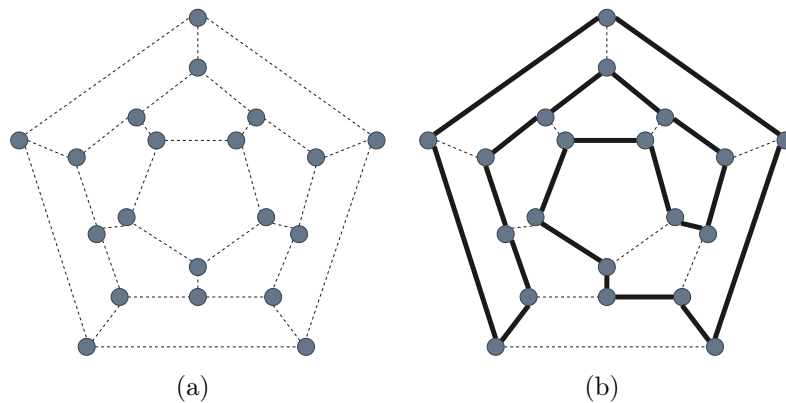


Figura 3.1: (a) Jogo de Hamilton e (b) Ciclo Hamiltoniano [10]

Hamiltoniano de v_i até v_j [72]. As definições de caminhos e ciclos Hamiltonianos permitem criar rotas em diferentes cenários representados por grafos.

Nas aplicações do mundo real, além de definir rotas, é preciso otimizá-las. A otimização de rotas é um processo caracterizado pela busca de melhores caminhos a partir de uma ou mais variáveis e restrições. Esse conceito está relacionado, mas não restrito, a encontrar a(s) rota(s) mais curta(s) entre dois pontos. Considerando como exemplo o cenário de otimização de rotas para veículos, tem-se que as rotas devem passar pelos pontos de parada, que representam espaços geográficos dispersos. Nesse caso, as possíveis variáveis são a distância, o tempo de trajeto, o consumo de combustível, dentre outros. Além disso, como exemplos de restrições, é possível mencionar o número de pontos de parada, limites de tempo e capacidade de armazenamento do veículo. Esse processo de otimização está associado ao Problema do Caixeiro Viajante.

3.3 Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) é um problema de otimização combinatória amplamente abordado na literatura, desde 1949, quando provavelmente o termo Problema do Caixeiro Viajante foi adotado pela primeira vez [67]. Esse problema de otimização corresponde a encontrar a rota de um vendedor que parte de uma localização residencial, visita um conjunto pré-estabelecido de cidades e retorna ao local de origem, de forma que a distância total percorrida seja mínima e cada cidade seja visitada exatamente uma vez [43]. Sendo assim, o Problema do Caixeiro Viajante tem como objetivo encontrar um ciclo Hamiltoniano no grafo G , de forma que a soma dos custos das arestas desse ciclo seja a menor possível.

É possível classificar o PCV como simétrico se os custos das arestas do conjunto L apresentarem a característica $c(v_i, v_j) = c(v_j, v_i)$. De qualquer outra forma, é considerado assimétrico. Ademais, esse problema pode ser classificado como completo se existir caminho direto entre todos os pontos. Se não existir, é chamado não completo. Além disso, o número de combinações de ciclos Hamiltonianos possíveis em um grafo pode ser calculado por $(|V| - 1)!$, sendo $|V|$ o número de vértices do grafo G . Esse problema é *Nondeterministic Polynomial-time* (NP-difícil), então, aumentar o valor de $|V|$ pode invi-

abilizar a comparação entre todas as rotas para encontrar a solução ótima [10] [64]. Nessa perspectiva, a Figura 3.2 apresenta um exemplo do menor ciclo Hamiltoniano possível considerando um grafo simétrico completo de cinco vértices.

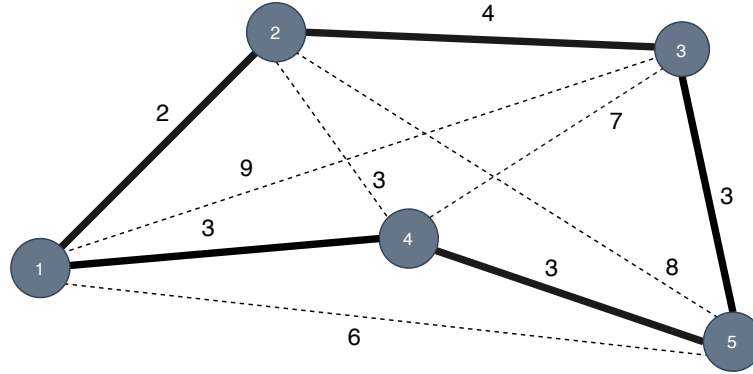


Figura 3.2: Exemplo do Problema do Caixeiro Viajante

De acordo com a Figura 3.2, observa-se que o menor ciclo Hamiltoniano possível é dado pelo conjunto $S = \{1, 2, 3, 5, 4, 1\}$, então S é a solução do PCV para o grafo dado.

3.4 Modelo Matemático do PCV

A formulação matemática de Dantzig, Fulkerson e Johnson para o PCV pode ser expressa da seguinte forma [26] [9]:

$$\text{Min } z = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} x_{i,j} \quad (3.1)$$

Sujeito a:

$$\sum_{i=1}^n x_{i,j} = 1 \quad \forall j \in N \quad (3.2)$$

$$\sum_{j=1}^n x_{i,j} = 1 \quad \forall i \in N \quad (3.3)$$

$$\sum_{i,j \in S} x_{i,j} \leq |S| - 1 \quad \forall S \subset N \quad (3.4)$$

$$x_{i,j} \in \{0, 1\} \quad \forall i,j \in N \quad (3.5)$$

A Equação 3.1 representa a função objetivo, a qual minimiza o custo da rota. Além disso, as restrições na Equação 3.2 e na Equação 3.3 garantem que cada nó seja visitado apenas uma vez. Considerando um grafo G , tem-se que S é subgrafo de G e $|S|$ é o número de vértices deste subgrafo. Nesse sentido, a restrição expressa pela Equação 3.4 garante que sub-rotas não sejam geradas, visto que o objetivo é encontrar uma rota que visite todos os nós. Para isso, considera-se $|S| - 1$ o número de nós da rota. Por fim, a restrição 3.5 impõe variáveis de decisão binárias [9].

O PCV é um exemplo clássico na literatura de problemas NP-completos, uma vez que não existe algoritmo em tempo polinomial que possa solucioná-lo, a menos que $P = NP$. Sendo assim, os métodos exatos exigem alto custo computacional e solucionar o PCV pode ser impraticável dependendo do número de vértices do grafo. De acordo com Prestes [64], métodos exatos tem como desvantagem o grande aumento do custo computacional, tempo de processamento e consumo de memória em relação ao tamanho das instâncias. Em vista disso, abordagens heurísticas tem sido aplicadas na literatura, para obter soluções ótimas ou aproximadas do PCV.

3.5 Heurísticas

As heurísticas são estratégias de resolução de problemas que não garantem encontrar a solução ótima, mas permitem retornar uma solução em tempo adequado para as necessidades da aplicação [35]. Geralmente são aplicadas em problemas nos quais se prefere uma boa solução em tempo razoável ou rápido, do que uma solução ótima que demanda muitos recursos computacionais e tempo. Dentre as heurísticas que podem ser aplicadas ao Problema do Caixeiro Viajante, é possível destacar: Vizinho mais Próximo, Inserção do mais Próximo e Inserção do mais Distante. Para descrevê-las, os símbolos apresentados na Tabela 3.1 são utilizados. Além disso, a Tabela 3.2 apresenta o sumário de outros símbolos e significados que também são utilizados para descrever as heurísticas.

Tabela 3.2: Sumário de símbolos e significados

Símbolo	Significado
$v_{inicial}$	Vértice inicial de uma rota
v_{final}	Vértice final de uma rota
v_{atual}	Vértice atual
N	Conjunto de vértices não visitados
n_i	Elemento i de N
s_i	Elemento i de S
i_s	índice do vértice s_i no conjunto S
p	índice para inserir o vértice atual no conjunto S
C_{min}	Custo mínimo do vértice atual até o próximo vértice

As heurísticas de construção de rotas para o PCV requerem uma matriz de custos das arestas do grafo, definida previamente como C . A partir de um conjunto inicial de vértices (V), os algoritmos geram circuitos viáveis (S), modificando o circuito a cada iteração. Para isso, utiliza-se um critério de escolha de vértices, o qual pode variar de acordo com a heurística escolhida [24]. Nestes algoritmos, que serão descritos a seguir, o critério de escolha é baseado no vértice de menor custo. Sendo assim, o Algoritmo 1 apresenta a função `extraí_minimo`, que extrai o vértice de menor custo do conjunto de nós não visitados.

O Algoritmo 1 tem como entrada o conjunto de nós não visitados (N), a matriz de custos (C) e o vértice atual. A variável do custo mínimo é inicializada com valor infinito (linha 1). Depois, um laço de repetição é iniciado para percorrer todos os elementos do

Algoritmo 1: função `extraí_minimo()`

Entrada: N, C, v_{atual}
Saída: v_{atual}

```

1  $C_{min} \leftarrow \infty$ 
2 para cada  $n_i$  em  $N$  faça
3   se  $C[v_{atual}][n_i] < C_{min}$  então
4      $C_{min} \leftarrow C[v_{atual}][n_i]$ 
5      $n \leftarrow n_i$ 
6  $v_{atual} \leftarrow n$ 
7 retorna  $v_{atual}$ 

```

conjunto N (linha 2). Dentro do laço, se o custo da aresta (v_{atual}, n_i) for menor do que o custo mínimo encontrado até agora (linha 3), atualiza-se a variável do custo mínimo (C_{min}) com o valor do custo da aresta (v_{atual}, n_i) (linha 4). Além disso, o vértice atual passa a ser n (linha 6) e depois é retornado pela função (linha 7).

3.5.1 Vizinho Mais Próximo

A Vizinho mais Próximo representa a heurística mais intuitiva para solucionar o Problema do Caixeiro Viajante. Considerando um grafo não direcionado e com os pesos das arestas definidos, essa heurística inicia o ciclo a partir de um nó presente no grafo, que passa a ser o vértice atual; depois, encontra-se o nó mais próximo ao último nó inserido na rota, que passa a ser o vértice atual. Esse processo se repete até que todos os nós estejam no ciclo. Por fim, o último nó se conecta com o primeiro, para fechar o ciclo Hamiltoniano [10] [24]. A heurística VP está descrita mais detalhadamente no Algoritmo 2, permitindo a construção de ciclos e caminhos Hamiltonianos, ou seja, considera-se que o vértice final não necessariamente é o mesmo vértice inicial. Ademais, assume-se que o grafo é completo e ponderado.

Algoritmo 2: Heurística Vizinho mais Próximo

Entrada: $v_{inicial}, v_{final}, V$
Saída: S

```

1  $S \leftarrow \{v_{inicial}\}$ 
2  $V \leftarrow V - v_{final}$ 
3  $N \leftarrow V - v_{inicial}$ 
4  $v_{atual} \leftarrow v_{inicial}$ 
5 enquanto  $|S| < |V|$  faça
6    $v_{atual} \leftarrow \text{extraí\_minimo}(N, C, v_{atual})$ 
7    $S \leftarrow S \cup \{v_{atual}\}$ 
8    $N \leftarrow N - \{v_{atual}\}$ 
9  $S \leftarrow S \cup \{v_{final}\}$ 
10 retorna  $S$ 

```

O Algoritmo 2 inicia adicionando o vértice inicial ($v_{inicial}$) na rota S (linha 1). Depois, o vértice final é retirado do conjunto de vértices do grafo (V) (linha 2). Isso se dá para que o vértice final não seja escolhido, visto que será adicionado como último nó da rota. Sendo

assim, o vértice inicial é retirado do conjunto de vértices não visitados (N) (linha 3). Em seguida, o vértice atual (v_{atual}) é inicializado de acordo com o vértice inicial ($v_{inicial}$) (linha 4). O algoritmo cria um laço que adiciona o nó de menor custo à rota enquanto o número de nós da rota ($|S|$) for menor que o número de vértices desconsiderando o nó final ($|V|$) (linhas 5-8). Após a rota conter todos os nós do conjunto V , o vértice final é adicionado na rota S (linha 9). Por fim, retorna-se o conjunto S (linha 10), o qual contém os nós do ciclo/caminho Hamiltoniano.

Além da heurística VP, a IP, descrita a seguir, é amplamente utilizada para definição de rotas utilizando grafos.

3.5.2 Inserção do Mais Próximo

A heurística IP possui três níveis de decisão: definição do ciclo inicial, que corresponde ao vértice de origem e o nó mais próximo; escolha do próximo nó que será inserido e a posição de inserção deste nó escolhido, na rota. Baseado nesses processos, o Algoritmo 3 apresenta a heurística IP.

No Algoritmo 3, inicialmente, adiciona-se o vértice de origem ($v_{inicial}$) na rota (S). Além disso, inicializa-se os conjuntos de vértices do grafo (V), dos nós não visitados (N) e o custo mínimo (C_{min}) (linhas 1-4). Então, a função `extraí_minimo` é utilizada para encontrar o vértice atual, que é o nó não visitado mais próximo do vértice de origem (linha 5). Em seguida, inicia-se um *loop*, que é executado enquanto o tamanho da rota ($|S|$) for menor que o número de vértices do conjunto $|V|$ (linha 7). Em seguida, cria-se outros dois laços para encontrar o vértice não visitado que está mais próximo de qualquer vértice do ciclo (S) (linhas 9-13). Então, o nó mais próximo de qualquer vértice de S passa a ser o vértice atual (linha 13).

Após redefinir o vértice atual, o algoritmo da heurística IP encontra o índice/posição para inserir o vértice atual no ciclo. Para isso, cria-se um laço que percorre todos os vértices da rota S e calcula o custo da inserção do nó atual em determinada posição (p) de S (linhas 15-19). Esse custo de inserção do vértice atual é calculado de acordo com a Equação 3.6 [80] [10] [24].

$$\min_{s_i, s_{i+1} \in S} (C_{s_i v} + C_{v s_{i+1}} - C_{s_i s_{i+1}}) \quad (3.6)$$

Onde S é a rota, s_i e s_{i+1} são elementos de S , $C_{s_i v}$ representa o custo da aresta que conecta s_i ao vértice atual, $C_{v s_{i+1}}$ é o custo da aresta que interliga o vértice atual à s_{i+1} e $C_{s_i s_{i+1}}$ é o custo da aresta que conecta s_i à s_{i+1} .

No Algoritmo 3, o vértice atual é inserido no ciclo de acordo com a posição encontrada que garante o menor custo (C_{total}) (linha 20). Em seguida, o vértice atual é removido do conjunto de nós não visitados N (linha 21). Por fim, o algoritmo adiciona o vértice final na rota e retorna o conjunto S (linha 22).

3.5.3 Inserção do Mais Distante

A heurística Inserção do mais Distante é muito semelhante à Inserção do mais Próximo. Porém, neste caso, ao invés de escolher o vértice mais próximo de qualquer um do ciclo,

Algoritmo 3: Heurística Inserção do mais Próximo

```

Entrada:  $v_{inicial}$ ,  $v_{final}$ ,  $V$ 
Saída:  $S$ 
1  $S \leftarrow \{v_{inicial}\}$ 
2  $V \leftarrow V - v_{final}$ 
3  $N \leftarrow V - v_{inicial}$ 
4  $C_{min} \leftarrow \infty$ 
5  $v_{atual} \leftarrow \text{extrai\_minimo}(N, C, v_{inicial})$ 
6  $S \leftarrow S \cup \{v_{atual}\}$ 
7 enquanto  $|S| < |V|$  faça
8    $C_{min} \leftarrow \infty$ 
9   para cada  $s_i$  em  $S$  faça
10    para cada  $n_i$  em  $N$  faça
11      se  $C[s_i][n_i] < C_{min}$  então
12         $C_{min} \leftarrow C[s_i][n_i]$ 
13         $v_{atual} \leftarrow n_i$ 
14     $C_{min} \leftarrow \infty$ 
15    para cada  $s_i$  em  $S$  faça
16       $C_{total} \leftarrow C[s_i][v_{atual}] + C[v_{atual}][s_{i+1}] - C[s_i][s_{i+1}]$ 
17      se  $C_{total} < C_{min}$  então
18         $C_{min} \leftarrow C_{total}$ 
19         $p \leftarrow i_s$ 
20     $S[p] \leftarrow v_{atual}$ 
21     $N \leftarrow N - \{v_{atual}\}$ 
22  $S \leftarrow S \cup \{v_{final}\}$ 
23 retorna  $S$ 

```

escolhe-se o mais distante [10] [24].

As heurísticas apresentadas podem ser utilizadas para encontrar rotas baseadas em ciclos e caminhos Hamiltonianos, considerando grafos completos ponderados. Porém, para encontrar caminhos mínimos entre dois pontos de um grafo ponderado não completo, é preciso utilizar algoritmos de busca.

3.6 Estratégias de Busca

As estratégias de busca são amplamente utilizadas para encontrar o caminho mínimo entre dois vértices em grafos direcionados. As aplicações que empregam esse tipo de algoritmo geralmente utilizam informações geográficas, como sistemas de sugestões de rotas, mapas cartográficos, banco de dados espaciais, dentre outros. Nesse sentido, o problema da busca por um caminho mínimo entre dois pontos específicos em um mapa é chamado de problema ponto-a-ponto (*Point-to-Point* - P2P) [59].

No problema P2P, dado um grafo G orientado e ponderado, encontra-se um caminho mínimo de um vértice inicial até um vértice destino. Um dos algoritmos mais conhecidos na literatura para solucionar este problema é o Algoritmo de Dijkstra, o qual considera grafos com pesos de arestas não negativos. Além do clássico Dijkstra, é possível mencionar também a variação Dijkstra bidirecional, o *Shortest Path Faster Algorithm*, que aceita

pesos de arestas negativos, e o A-star, os quais serão descritos neste capítulo. Para isso, utiliza-se as notações de símbolos apresentados na Tabela 3.3, além dos símbolos já apresentados nas Tabelas 3.1 e 3.2.

Tabela 3.3: Sumário de símbolos e significados

Símbolo	Significado
Z	Conjunto de vértices escaneados
A	Matriz de vértices adjacentes
a_i	Elemento i do conjunto de adjacentes $A[v_{atual}]$
Y	Conjunto com chaves/valores associados a cada vértice
P	Conjunto com o predecessor de cada vértice do grafo
Q	Fila de prioridades
q_i	Elemento i da fila de prioridades Q
H	Conjunto de valores heurísticos associados a cada vértice
E	Conjunto de custos estimados do vértice inicial até os vértices do grafo
F	Conjunto de valores da função de avaliação dos vértices

Os algoritmos de busca consideram que o grafo de entrada possui uma lista de adjacências, de modo que o conjunto dos vizinhos do vértice $v_i \in V$ é denominado $A[v_i]$. Além disso, utiliza-se a função `extraí_minimo_fila()`, apresentada no Algoritmo 4, para identificar o vértice que possui a menor chave de uma fila. Nesse caso, a fila de prioridades (Q) representa o conjunto dos vértices do grafo que foram abertos, porém ainda não foram visitados. Além disso, as chaves associadas aos vértices são representadas pelo conjunto Y .

Algoritmo 4: função `extraí_minimo_fila()`

Entrada: Q, Y
Saída: v_{atual}

```

1  $v_{atual} \leftarrow \emptyset$ 
2  $C_{min} \leftarrow \infty$ 
3 para cada  $q_i$  em  $Q$  faça
4   se  $Y[q_i] < C_{min}$  então
5      $C_{min} \leftarrow Y[q_i]$ 
6      $v_{atual} \leftarrow q_i$ 
7 retorna  $v_{atual}$ 
```

O Algoritmo 4 inicializa as variáveis vértice atual (v_{atual}) e custo mínimo (C_{min}) (linhas 1-2). Em seguida, para cada vértice da fila de prioridades (Q), se a chave associada ao vértice ($Y[q_i]$) for menor do que o custo mínimo encontrado até agora, atualiza-se o custo mínimo e o vértice atual (linhas 3-6). Por fim, o vértice com menor chave associada, ou seja, menor custo, é retornado como vértice atual (linha 7).

Além do conjunto com chaves associadas a cada um dos vértices do grafo, os algoritmos apresentados neste capítulo utilizam um conjunto de predecessores (P). Este conjunto representa a árvore de caminhos mínimos que os algoritmos constroem. Nesse sentido, $P[v_i]$ armazena o predecessor de v_i , ou seja, o nó antecessor a v_i no caminho que vai do vértice de origem até o objetivo [59]. Com base nessa ideia, o Algoritmo 5 apresenta a

função `reconstroi_caminho()`, a qual utiliza o conjunto de predecessores obtido a partir dos algoritmos de busca, para reconstruir o caminho do $v_{inicial}$ até o v_{final} .

Algoritmo 5: função `reconstroi_caminho()`

Entrada: $P, v_{final}, v_{inicial}$ Saída: S <pre> 1 $S \leftarrow \{v_{final}\}$ 2 $v_{atual} \leftarrow v_{final}$ 3 enquanto $v_{atual} \neq v_{inicial}$ faça 4 $v_{atual} \leftarrow P[v_{atual}]$ 5 $S \leftarrow S \cup \{v_{atual}\}$ 6 $S \leftarrow \text{inverte_sequencia}(S)$ 7 retorna S </pre>

O Algoritmo 5 inicializa a rota S com a variável de entrada v_{final} , que representa o nó objetivo (linha 1). Além disso, o vértice atual é inicializado com a variável v_{final} (linha 2). Depois, um laço de repetição é criado para ser executado enquanto o vértice atual não for igual ao vértice inicial (linha 3). Nesse laço, o predecessor do v_{atual} passa a ser o novo vértice atual, o qual é inserido na rota (S) (linhas 4-5). Após reconstruir o caminho, a sequência da rota é invertida, para iniciar na origem e finalizar no objetivo (linha 6).

Os algoritmos de busca encontram caminhos mínimos entre dois vértices utilizando as funções `extraí_minimo_fila()` e `reconstroi_caminho()`. A seguir, descreve-se os algoritmos de Dijkstra, Bidirecional, SPFA e A-star.

3.6.1 Algoritmo de Dijkstra

O Algoritmo de Dijkstra foi proposto por Edsger W. Dijkstra motivado pelos problemas de caminhos mínimos e árvore geradora mínima, relacionados a teoria dos grafos [32]. Esse algoritmo cria uma árvore contendo inicialmente apenas o vértice inicial e a cada iteração, o algoritmo acrescenta um novo vértice à árvore usando a estratégia gulosa. Por fim, uma árvore de caminhos mínimos é formada. O Algoritmo 6 apresenta esses processos mais detalhadamente.

De acordo com o Algoritmo 6, inicializa-se o conjunto E com valores infinitos, que representam as distâncias entre os vértices do grafo e o vértice inicial. Além disso, o conjunto P de predecessores é inicializado com valores nulos (linhas 1-3). Se $E[v_i]$ é infinito, significa que ainda não foi encontrado um caminho do vértice v_i até a origem. Além disso, $|E| = |V|$ e $|P| = |V|$, ou seja, os conjuntos E e P são do mesmo tamanho que o conjunto de vértices (V), pois cada vértice do grafo possui um custo até o $v_{inicial}$ e um predecessor. Em seguida, atualiza-se a distância do vértice inicial até a origem (que é o próprio $v_{inicial}$), com o valor de 0 (linha 4).

O conjunto de nós escaneados (Z) é inicializado com valor nulo e o conjunto Q , o qual representa a fila de prioridades, recebe os vértices do grafo (linhas 5-6). Depois, um laço de repetição é iniciado para ser executado enquanto o vértice atual for diferente do destino (v_{final}) (linha 7). O nó de menor chave considerando o conjunto E é extraído de Q , por meio da função `extraí_minimo_fila()` (linha 8). Então, o vértice extraído

Algoritmo 6: Algoritmo de Dijkstra

Entrada: $v_{inicial}$, v_{final} , V , A , C
Saída: S

```

1 para cada  $v_i$  em  $V$  faça
2    $E[v_i] \leftarrow \infty$ 
3    $P[v_i] \leftarrow NULL$ 
4  $E[v_{inicial}] \leftarrow 0$ 
5  $Z \leftarrow \emptyset$ 
6  $Q \leftarrow V$ 
7 enquanto  $v_{atual} \neq v_{final}$  faça
8    $v_{atual} \leftarrow \text{extrai\_minimo\_fila}(Q, E)$ 
9   para cada  $a_i$  em  $A[v_{atual}]$  faça
10    se  $E[a_i] > E[v_{atual}] + C[v_{atual}][a_i]$  então
11       $E[a_i] \leftarrow E[v_{atual}] + C[v_{atual}][a_i]$ 
12       $P[a_i] \leftarrow v_{atual}$ 
13     $Z \leftarrow Z \cup \{v_{atual}\}$ 
14  $S \leftarrow \text{reconstroi\_caminho}(P, v_{atual}, v_{inicial})$ 
15 retorna  $S$ 

```

de Q é considerado o v_{atual} . Na primeira execução, o $E[v_{inicial}]$ possui o menor valor (0), então, o vértice inicial será selecionado para ser o vértice atual. Após identificar o v_{atual} , inicia-se outro laço de repetição que percorre os vértices adjacentes do v_{atual} (linha 9). Se a distância da origem até o vértice adjacente ($E[a_i]$) for maior do que o resultado obtido a partir da Equação 3.7, o valor de $E[a_i]$ e $P[a_i]$ são atualizados (linhas 11-12). Em seguida, o v_{atual} é inserido no conjunto Z (linha 13).

$$C_{total} = E_{v_{atual}} + C_{a_i v_{atual}} \quad (3.7)$$

Onde $E_{v_{atual}}$ representa o custo estimado do vértice atual até a origem e $C_{a_i v_{atual}}$ corresponde ao custo da aresta (a_i, v_{atual}) , sendo a_i um vértice adjacente do vértice atual.

O Algoritmo de Dijkstra possui complexidade de tempo de $O(n^2)$, sendo n o número de vértices do grafo. Esta versão é chamada de *implementação ingênua*, mas existem outras variações que diminuem a complexidade do algoritmo. É possível citar como exemplo de variação o Algoritmo de Dijkstra com heap de Fibonacci, descrito em [38, 20], o qual apresenta complexidade de $O(m + n \log n)$, sendo m o número de arestas. Além disso, cita-se também o Algoritmo de Dijkstra usando o Radiz heap, proposto em [3], o qual apresenta complexidade $O(m + n \log C)$, considerando C o custo máximo das arestas. Por fim, o Algoritmo de Dijkstra bidirecional, descrito a seguir, é uma variação que minimiza o espaço de busca em comparação com a implementação ingênua.

3.6.2 Dijkstra Bidirecional

Na década de 1960, uma versão do Algoritmo de Dijkstra específica para o problema P2P foi proposta, chamada Dijkstra Bidirecional [60, 34, 25]. Essa variação executa duas buscas do algoritmo de Dijkstra: a busca direta, que começa no $v_{inicial}$, e a busca inversa, a qual inicia no v_{final} . Nesse caso, cada busca possui um conjunto de estimativas, sendo E_d o

conjunto de estimativas de distância da busca direta e E_i o conjunto de estimativas de distância da busca inversa, ambos no grafo G . Nesse sentido, $E_d[v_i]$ representa a distância do vértice inicial até v_i e $E_i[v_i]$ representa a distância do vértice final até v_i [42].

Inicialmente, os vértices inicial e final são escaneados pelas buscas direta e inversa, respectivamente. Nesse estágio, o C_{total} do caminho de $v_{inicial}$ até v_{final} é igual a ∞ . As buscas podem ser executadas de forma alternada, então, os vértices adjacentes de $v_{inicial}$ são relaxados se a busca direta for executada primeiro, ou os vértices adjacentes de v_{final} , caso contrário. As buscas são executadas da mesma forma que ocorreram no Algoritmo de Dijkstra, com a diferença que dois conjuntos de estimativas são atualizados (E_d e E_i). Encontra-se o C_{total} do caminho de $v_{inicial}$ até v_{final} quando uma aresta (v_i, v_{i+1}) é relaxada pela busca direta/inversa e v_{i+1} já foi escaneado durante a busca na direção contrária. Nesse caso, os caminhos mínimos do vértice inicial até v_i e do v_{i+1} até o vértice final são descobertos. Em seguida, atualiza-se o valor do custo total se $C_{total} > E_d[v_i] + C[v_i][v_{i+1}] + E_i[v_{i+1}]$ [41].

Um possível critério de parada é quando uma das buscas escanear um vértice que já foi escaneado pela busca na direção contrária. De forma mais sofisticada, considerando min_d e min_i as menores chaves da fila de prioridade nas buscas direta e inversa, o algoritmo é finalizado quando $min_d + min_i \geq C_{total}$. A prova de que este critério está correto pode ser estudada em [41]. Nesse sentido, o Dijkstra bidirecional expande dois círculos, centrados nos vértices inicial e final. Quando estes círculos se alcançam, o caminho mínimo foi descoberto, então o algoritmo é finalizado. Sendo assim, o espaço de busca deste algoritmo pode ser reduzido por um fator 2, comparando com o algoritmo de Dijkstra [49].

3.6.3 Algoritmo A-Star

A-Star é um algoritmo de busca proposto por Hart, Nilsson e Raphael [48], para solucionar o problema P2P. Estes pesquisadores perceberam que, considerando o problema de busca de caminhos mínimos em grafos, as abordagens matemáticas geralmente priorizam mais a obtenção de soluções finais ótimas do que a viabilidade computacional dos algoritmos desenvolvidos. Enquanto que as heurísticas utilizam conhecimento do domínio do problema para realizar as buscas de forma computacionalmente eficiente, porém não garantem encontrar a solução ótima. Nessa perspectiva, o algoritmo A* reúne as duas abordagens, matemática e heurística, descrevendo como as informações do domínio do problema podem ser utilizadas em uma abordagem matemática formal [48].

No Algoritmo *A-Star*, utiliza-se um método heurístico que associa um valor $h(v_i)$ a todos os vértices do grafo G , sendo $v_i \in V$. Nesse sentido, $h(v_i)$ representa o menor custo estimado do caminho do vértice v_i até o objetivo (v_{final}), descrito na Equação 3.8.

$$h(v_i) = \min_{v_i, v_{final} \in V} C(v_i, v_{final}) \quad (3.8)$$

Onde $C(v_i, v_{final})$ corresponde ao custo do caminho que inicia em v_i e finaliza no v_{final} .

Para garantir a obtenção de soluções ótimas nas buscas em grafo, a heurística $h(v_i)$ possui o requisito de consistência, provado por Hart, Nilsson e Raphael em [48]. Esse requisito, apresentado na Equação 3.9, define que para todo vértice v_i e todo sucessor v_{i+1}

gerado por uma ação, o custo estimado de alcançar o vértice final a partir de v_i é menor ou igual ao custo da ação de se chegar ao sucessor somado ao custo estimado de alcançar o vértice final a partir de v_{i+1} .

$$h(v_i) \leq C(v_i, v_{i+1}) + h(v_{i+1}) \quad (3.9)$$

Além de $h(v_i)$, o algoritmo *A-Star* também utiliza o valor de $e(v_i)$, que corresponde ao custo estimado do caminho do nó inicial até v_i [68]. Associando a heurística $h(v_i)$ ao valor de $e(v_i)$, tem-se a função de avaliação dos vértices, descrita na Equação 3.10.

$$f(v_i) = e(v_i) + h(v_i) \quad (3.10)$$

A função de avaliação é utilizada para identificar qual vértice do grafo deve ser expandido durante a busca. A cada iteração, expande-se o nó que possui o menor valor de $f(v_i)$. O Algoritmo 7 descreve o passo-a-passo do A-star.

Algoritmo 7: Algoritmo A-star	
Entrada: $v_{inicial}$, v_{final} , V , H , A , C	
Saída: S	
1	$Q \leftarrow \{v_{inicial}\}$
2	$Z \leftarrow \emptyset$
3	para cada v_i em V faça
4	$E[v_i] \leftarrow \infty$
5	$F[v_i] \leftarrow \infty$
6	$P[v_i] \leftarrow \text{NULL}$
7	$E[v_{inicial}] \leftarrow 0$
8	$F[v_{inicial}] \leftarrow E[v_{inicial}] + H[v_{inicial}]$
9	enquanto $ Q > 0$ faça
10	$v_{atual} \leftarrow \text{extrai_minimo_fila}(Q, F)$
11	$Q \leftarrow Q - \{v_{atual}\}$
12	$Z \leftarrow Z \cup \{v_{atual}\}$
13	se $v_{atual} = v_{final}$ então
14	$S \leftarrow \text{reconstroi_caminho}(P, v_{atual}, v_{inicial})$
15	retorna S
16	para cada a_i em $A[v_{atual}]$ faça
17	se $a_i \notin Z$ então
18	$e \leftarrow E[v_{atual}] + C[v_{atual}][a_i]$
19	$f \leftarrow e + H[a_i]$
20	se $E[a_i] > e$ então
21	$F[a_i] \leftarrow f$
22	$E[a_i] \leftarrow e$
23	$P[v_i] \leftarrow v_{atual}$
24	se $a_i \notin Q$ então
25	$Q \leftarrow Q \cup \{a_i\}$

No Algoritmo 7, o vértice inicial, o vértice final, o conjunto de vértices do grafo (V) e o conjunto de valores heurísticos associados a cada vértice (H) são entradas. Sendo assim, o algoritmo inicia a partir da inserção do vértice inicial na fila de prioridades (Q) (linha 1). Além disso, o conjunto de vértices escaneados (Z) é inicializado como vazio

(linha 2). Em seguida, inicializa-se o conjunto E com valores infinitos, que representam a distância do menor caminho encontrado do nó inicial até cada um dos vértices do grafo ($e(v_i)$) (linha 4). O único valor de E conhecido no começo do algoritmo é o $E[v_{inicial}]$, o qual equivale a zero (linha 7). O conjunto F , que corresponde aos valores da função de avaliação dos vértices $f(v_i)$, é inicializado com valores infinitos (linha 5) e o conjunto dos predecessores (P), com valores nulos (linha 6). Ademais, o valor da função de avaliação do vértice inicial é calculado e adicionado no conjunto $F[v_{inicial}]$ (linha 8).

Um laço de repetição é criado no Algoritmo 7 para ser executado enquanto a fila de prioridades (Q) ainda tiver vértices não visitados (linha 9). Sendo assim, o nó da fila de prioridades que possui o menor valor da função de avaliação (F) é definido como nó atual (linha 10), a partir da função `extraí_minimo_fila()`. Em seguida, esse vértice é retirado da fila de prioridades e adicionado no conjunto de nós visitados (Z) (linhas 11-12). Se o vértice atual é igual ao objetivo (v_{final}), a função `reconstroi_caminho()` é chamada para criar a rota S a partir dos predecessores dos vértices (linhas 14-15). Caso o vértice final ainda não tenha sido alcançado, identifica-se os nós adjacentes do vértice atual por meio de um laço de repetição (linha 16). Nesse laço, se o nó adjacente ($a_i \in A[v_{atual}]$) ainda não foi visitado, calcula-se os valores de $e(a_i)$ e $f(a_i)$, representados pelas variáveis e e f no algoritmo (linhas 17-19). O valor de e é calculado a partir da Equação 3.11.

$$g \leq E(v_{atual}) + C(v_{atual}, a_i) \quad (3.11)$$

Onde $E(v_{atual})$ representa o custo para alcançar o vértice atual partindo da origem e $C(v_{atual}, a_i)$ corresponde ao custo da aresta que conecta o vértice atual ao adjacente a_i .

Os valores e e f são atualizados nos conjuntos E e F somente se o valor obtido em e for menor do que o que está armazenado em $E[a_i]$ (linhas 20-22). Em outras palavras, se o caminho encontrado para chegar em a_i for menor do que qualquer outro caminho encontrado antes. Caso nenhum caminho tenha sido encontrado antes, o valor de $E[a_i]$ será maior porque é inicializado como infinito. Nesse caso, o predecessor do nó adjacente é atualizado (linha 23). Além disso, adiciona-se a_i na fila de prioridades, caso ainda não esteja (linhas 24-25).

A complexidade do algoritmo será exponencial, a menos que a condição expressa na Equação 3.12 seja satisfeita [68].

$$|h(v_i) - h^*(v_i)| \leq O(\log(h^*(v_i))) \quad (3.12)$$

Onde $h(v_i)$ é o custo estimado para alcançar o objetivo a partir de v_i , $h^*(v_i)$ é o custo verdadeiro para ir de v_i até o objetivo e $v_i \in V$, considerando V o conjunto de vértices do grafo G . Além do algoritmo de Dijkstra e *A-star*, outro algoritmo de busca amplamente utilizado na literatura é o SPFA.

3.6.4 Shortest Path Faster Algorithm

O Algoritmo de busca SPFA foi proposto por Fanding [36] e construído com base no Algoritmo de Bellman-Ford, o qual relaxa todas as arestas para encontrar a solução. O diferencial do SPFA está em utilizar uma fila *First In, First Out* (FIFO) para armazenar

os nós candidatos. Assim como o clássico algoritmo de Bellman-Ford, o SPFA tem como vantagem realizar buscas em grafos que possuem arestas de pesos negativos, além de identificar ciclos de peso negativo, diferentemente dos algoritmos de Dijkstra e A-star [81, 83]. O Algoritmo 8 descreve os processos do SPFA.

Algoritmo 8: Algoritmo SPFA	
	Entrada: $v_{inicial}, v_{final}, V, A, C$
	Saída: S
1	$Q \leftarrow \{v_{inicial}\}$
2	para cada v_i em V faça
3	$E[v_i] \leftarrow \infty$
4	$P[v_i] \leftarrow \text{NULL}$
5	$E[v_{inicial}] \leftarrow 0$
6	enquanto $ Q > 0$ faça
7	$v_{atual} \leftarrow \text{extraí_primeiro_item}(Q)$
8	$Q \leftarrow Q - \{v_{atual}\}$
9	para cada a_i em $A[v_{atual}]$ faça
10	$C_{total} \leftarrow E[v_{atual}] + C[a_i][v_{atual}]$
11	se $E[a_i] > C_{total}$ então
12	$E[a_i] \leftarrow C_{total}$
13	$P[a_i] \leftarrow v_{atual}$
14	se $a_i \notin Q$ então
15	$Q \leftarrow Q \cup \{a_i\}$
16	$S \leftarrow \text{reconstroi_caminho}(P, v_{final}, v_{inicial})$
17	retorna S

No Algoritmo 8, uma fila FIFO (Q) é criada e inicializada com o vértice de origem (linha 1). Em seguida, o conjunto que armazena as distâncias entre o vértice inicial e os vértices do grafo (E) é inicializado com valor infinito para todos os nós (linha 3). Além disso, o conjunto de predecessores é inicializado com valor nulo para todos os vértices do grafo (linha 4). Depois, a distância do vértice inicial é atualizada com valor zero (linha 5).

Um laço de repetição é iniciado para ser executado enquanto houver vértices na fila Q (linha 6). Nesse laço, o primeiro item da fila se torna o vértice atual (linha 7) e é removido de Q (linha 8). Além disso, cria-se outro laço de repetição que percorre os nós adjacentes do vértice atual (linha 9). Nesse caso, calcula-se o custo total de cada nó adjacente (linha 10) da mesma forma que no Algoritmo de Dijkstra, de acordo com a Equação 3.7. Se o custo total calculado for menor do que a distância do vértice adjacente até o vértice inicial, armazenado em $E[a_i]$, atualiza-se o valor de $E[a_i]$ (linha 12), sendo a_i o elemento i do conjunto de nós adjacentes do vértice atual. Além disso, atualiza-se também o predecessor de a_i ($P[a_i]$) (linha 13). Em seguida, adiciona-se o vértice adjacente à fila de prioridades, caso ainda não esteja (linhas 14-15). Por fim, a rota é reconstruída por meio da função *reconstroi_caminho* (linha 16).

O pior caso de complexidade do algoritmo SPFA é $O(ne)$, onde n representa o número de nós e e , o número de arestas. Nesse sentido, é a mesma complexidade do algoritmo de Bellman-Ford. No entanto, este caso quase nunca é alcançado no SPFA [81].

3.7 Resumo Conclusivo

O Problema do Caixeiro Viajante tem como objetivo encontrar um ciclo Hamiltoniano em um grafo G , de modo que a soma dos custos das arestas desse ciclo seja a menor possível. Esse problema é NP-difícil e dependendo do número de vértices do grafo, torna-se inviável calcular a solução ótima por meio de métodos exatos. Sendo assim, existem diversos métodos heurísticos na literatura que permitem criar um ciclo Hamiltoniano em tempo de processamento viável sem garantia de encontrar a solução ótima. Dentre as heurísticas para solução do PCV, é possível mencionar a Vizinho mais Próximo, Inserção do mais Próximo e Inserção do mais Distante. A heurística VP tem a vantagem de ser mais simples e intuitiva. Essas heurísticas podem ser aplicadas na construção de rotas onde há obrigatoriedade de passar por todos os vértices de um grafo. Para gerar rotas sem essa obrigatoriedade, é preciso utilizar uma estratégia de busca, a qual encontra um caminho mínimo entre dois pontos. As estratégias de busca apresentadas neste capítulo foram Dijkstra bidirecional, A-star e SPFA, amplamente conhecidas na literatura.

Capítulo 4

Trabalhos Relacionados

Este capítulo apresenta os trabalhos relacionados, separados em duas seções. A Seção 4.1 é denominada Problemas de Roteamento de Veículos e se refere a trabalhos que expõem diferentes tipos de problemas em qualquer contexto de roteamento de veículos. A Seção 4.2 é de Roteamento de Veículos para Coleta de Resíduos, que compreende os trabalhos com soluções de roteamento para a coleta de Resíduos Sólidos Urbanos, utilizando veículos automotores ou de tração humana.

4.1 Problemas de Roteamento de Veículos

O trabalho de Liao e Ting [54] apresenta uma categoria de problema logístico denominada *Selective Pickup and Delivery Problem* (SPDP). Neste caso, deve-se coletar mercadorias de alguns vértices e cumprir todas as respectivas entregas, de modo que a carga do veículo em cada nó não fique negativa. Para tanto, foi proposto um Algoritmo Genético estrategicamente adaptado à restrição de carga. Esta solução mostra-se viável para algumas aplicações de coleta e entrega no mundo real. Se o problema proposto for aplicado à criação de rotas para coleta de resíduos, tem-se os vértices de coleta, que são os de recolhimento de material, e os pontos de entrega são os galpões de triagem.

O trabalho de Huang e Ting [50] demonstra um Problema de Roteamento de Veículos chamado *Multi-Vehicle Selective Pickup and Delivery Problem* (MVSPDP). Trata-se de uma extensão do problema SPDP, abordado em [54]. Neste caso, o limite de capacidade de carga não deve ser atingido e existe uma distância máxima a ser percorrida por cada veículo. Nesse sentido, propõe-se um Algoritmo Genético com Religação de Caminho (GAPR), que tem como objetivo minimizar a distância total percorrida por todos os veículos da frota, melhorando a qualidade das soluções geradas pelo algoritmo, em comparação com o Algoritmo Genético (AG) proposto em [54].

O trabalho de Wang et al. [79] propõe um algoritmo de otimização de colônia de formigas chamado *Ant Multiple Rounds* (AMR) para solucionar o Capacitated Vehicle Routing Problem (CVRP). O algoritmo proposto é fundamentado no algoritmo de otimização *Ant Colony System* (ACR), proposto em [33]. No ACR, cada uma das formigas representa um veículo, que inicia seu percurso saindo de um depósito e visitando alguns vértices. Quando os veículos retornam ao depósito, os caminhos são construídos a partir

dos trajetos percorridos pelas formigas. Portanto, as soluções viáveis são construídas cooperativamente, a partir da comunicação entre as formigas. Sendo assim, o AMR funciona de maneira parecida, mas permite que as formigas entrem e saiam do depósito mais de uma vez. Nesse sentido, considera-se o caminho de cada formiga como uma solução viável, simplificando a construção de soluções. No entanto, este trabalho tem como foco produzir soluções viáveis, independente da qualidade dos caminhos gerados.

O trabalho de Rahaman et al. [65] apresenta um algoritmo de geração de rotas acessíveis para pessoas com deficiência e idosas, a partir de métricas como distância vertical e inclinação máxima das vias. Além das rotas acessíveis, sugere-se rotas rápidas, considerando a minimização da distância. Então, os usuários podem escolher a rota mais adequada de acordo com suas preferências. Nesse sentido, propõe-se a implementação da solução *Contour-based Accessible Path Routing Algorithm* (CAPRA), que utiliza o algoritmo de busca A-star.

Niu et al. [61] propõem um modelo de otimização para minimizar a distância, consumo de combustível, emissões de dióxido de carbono (CO₂) e os custos financeiros das rotas de caminhões de logística considerando janelas de tempo. As rotas são configuradas como abertas, pois os veículos não retornam para o depósito após visitarem todos os clientes. Além disso, a carga máxima do caminhão é configurada como parâmetro dependendo do tipo de veículo utilizado na rota. O algoritmo proposto utiliza Busca Tabu (BT) e a heurística VP modificada para identificar as rotas. Os resultados do trabalho demonstram que, em comparação com as rotas fechadas, as rotas abertas reduziram o custo financeiro total em 20%. Ademais, os custos das emissões de combustível e CO₂ diminuíram quase 30%.

O trabalho de Andelmin e Bartolini [5] aborda o *Green Vehicle Routing Problem* (G-VRP). Nesse problema, cada cliente (vértice) do cenário deve ser visitado por pelo menos um veículo de uma frota. Para isso, as rotas devem iniciar e finalizar no depósito, além de proporcionar a menor distância possível, considerando restrições de combustível para cada veículo. Nessa perspectiva, o trabalho propõe uma metaheurística chamada *Multi-Start Local Search* (MSLS). O algoritmo faz uma reformulação do problema G-VRP baseada em multigrafo, que é um tipo de grafo com múltiplas arestas entre nós. No algoritmo, os nós representam clientes e um depósito. Além disso, cada aresta representa uma sequência de paradas consecutivas de reabastecimento entre dois nós.

O trabalho de Altabeeb, Mohsen e Ghallab [4] apresenta uma solução híbrida chamada *Capacitated Vehicle Routing Problem - Firefly Algorithm* (CVRP-FA), baseada no algoritmo do vagalume, para solucionar o problema de roteamento de veículos com limites de capacidade. Para isso, incorpora-se técnicas de busca local, como o método 2-Opt melhorado e 2-h-opt que são descritos no trabalho, para acelerar a convergência em direção à solução ótima. Além de acrescentar um método *crossover* e dois tipos de técnicas de mutação, baseadas em Algoritmo Genético, para equilibrar a intensificação e diversificação. Como resultado, o algoritmo proposto demonstrou rápida taxa de convergência e alta precisão computacional.

Goeke, Roberti e Schneider [40] propuseram uma solução que fornece um serviço que satisfaz as demandas de visita a clientes utilizando o mesmo motorista, característica chamada de consistência de motorista, aproximadamente ao mesmo tempo (consistência

do tempo de chegada). Nesse sentido, cria-se rotas considerando os clientes que solicitaram serviço e uma diferença máxima de tempo de chegada na residência dos clientes em cada dia de trabalho do motorista. Sendo assim, a solução proposta utiliza o método exato amplamente utilizado na literatura, chamado *Column Generation* (CG), juntamente com a heurística *Large Neighborhood Search* (LNS). O algoritmo resolve pequenas e médias instâncias com até cinco períodos de planejamento e 30 clientes.

A Tabela 4.1 apresenta a comparação dos trabalhos relacionados ao roteamento de veículos, considerando o ano de publicação, se considera a capacidade do veículo, minimização da distância percorrida pelo(s) veículo(s), custos financeiros e se utiliza de dados de inclinação das vias para gerar rotas. Sendo assim, as características abordadas nas propostas dos trabalhos relacionados foram marcadas na tabela com o símbolo (✓).

Tabela 4.1: Comparação dos trabalhos relacionados

Referência	Ano	Capacidade	Distância	Custos	Inclinação	Algoritmo
Liao e Ting [54]	2010	✓	✓			AG
Huang e Ting [50]	2011	✓	✓			GAPR
Wang et al. [79]	2016	✓	✓			AMR
Rahaman et al. [65]	2017		✓		✓	A-star
Niu et al. [61]	2018	✓	✓	✓		BT e VP
Andelmin e Bartolini [5]	2019		✓			MSLS
Altabeeb et al. [4]	2019	✓	✓			CVRP-FA
Goeke et al. [40]	2019	✓	✓			CG e LNS

4.2 Roteamento de Veículos para Coleta de Resíduos

O trabalho de Tavares et al. [76] propõe um modelo de 3 Dimensões (3D) baseado no Sistema de Informação Geográfica (SIG) ArcGIS, que gera rotas para a coleta seletiva otimizando o consumo de combustível de um veículo automotor de grande porte. Para tanto, considera-se a inclinação das vias e a massa do veículo na modelagem das rotas. Dessa forma, a pesquisa mostra que a rota mais curta muitas vezes não é a que gera menor custo financeiro. O trabalho foi implementado em dois cenários: cidade de Praia, com 20 pontos de coleta, e na ilha de Santiado, onde se gera uma rota para transporte dos materiais até o ponto de incineração. As rotas propostas proporcionaram uma redução de 8% e 12% do consumo de combustível no primeiro e no segundo cenário, respectivamente.

A pesquisa realizada por Benjamim e Beasley [11] gera rotas de coleta de resíduos para múltiplos veículos automotores considerando janelas de tempo, período de descanso dos condutores e múltiplas instalações de disposição de resíduos. Nesse sentido, os veículos partem do depósito, realizam a coleta em pontos específicos, seguem até a instalação de disposição para esvaziar a carga e, por fim, retornam ao depósito. Cada ponto de coleta possui um período de tempo disponível para visita. Seleciona-se conjuntos de pontos de coleta vizinhos com janelas de tempo parecidas para gerar rotas viáveis. As metaheurísticas utilizadas para solucionar o problema foram BT, *Variable Neighbourhood Search* (VNS) e um algoritmo combinando as duas, chamado *Variable Neighbourhood Tabu Search* (VNTS).

O trabalho de Das e Bhattacharyya [28] apresenta um esquema de geração de rotas de coleta que minimiza a distância e o tempo dos trajetos para diminuir os custos financeiros da gestão de resíduos. Para isso, considera-se quatro fases de coleta, sendo que a primeira é realizada utilizando veículos de tração humana e as demais fases empregam veículos automotores. A solução proposta para este problema calcula soluções ótimas baseadas no Problema do Caixeiro Viajante. Além disso, é avaliada a partir de dados reais geográficos e de descarte de materiais da cidade de Calcutá, na Índia. Como resultado, demonstra-se que é possível reduzir 30% do custo total da coleta utilizando o esquema proposto.

O trabalho de Fujdiak et al. [39] aborda o roteamento de veículo automotor de grande porte para a coleta de resíduos no contexto das Cidades Inteligentes e *Internet of Things* (IoT). Nesse caso, considera-se um cenário onde as lixeiras possuem sensores que indicam a porcentagem de resíduo depositado. Nesse sentido, o trabalho tem como objetivo identificar as rotas de coleta mais eficientes para coletar os resíduos das lixeiras que já atingiram a capacidade máxima. Para tanto, considerou-se que as lixeiras são os vértices e as vias são as arestas. O peso das arestas é calculado a partir da distância, densidade do tráfego e qualidade da via. Um Algoritmo Genético foi implementado para gerar as rotas. Os resultados evidenciaram que, em média, é possível reduzir os custos da coleta em 15%.

Khan e Samadder [51] desenvolveram um método de alocação de pontos de coleta de materiais e otimização de rotas para minimizar os custos da coleta de resíduos na cidade de Dhanbad, na Índia. Para criar as rotas otimizadas, utilizou-se como parâmetros a posição geográfica dos pontos de coleta, a densidade da população, os horários de coleta e a capacidade volumétrica dos caminhões. Além disso, a avaliação da solução foi realizada com base no ArcGIS. No entanto, não foram utilizados dados de inclinação das vias.

Nowakowski, Król e Mrówczyńska [62] apresentam uma solução para otimização de rotas para coletas sob demanda, na qual os residentes de determinada localidade solicitam o recolhimento dos materiais via aplicação móvel/web. Além de calcular rotas otimizadas, a solução proposta seleciona o tamanho apropriado do veículo de coleta, o número de veículos para atender todas as solicitações e considera a carga do veículo como parâmetro a ser otimizado, estabelecendo limites volumétricos e de peso. Este trabalho se concentra em minimizar o custo operacional associado às rotas, reduzindo os recursos necessários para realizar a coleta, como veículos e funcionários. Para gerar as rotas otimizadas, utiliza-se Algoritmo Genético e lógica fuzzy. Porém, a performance do algoritmo proposto não foi avaliada, o que é importante visto que algoritmos genéticos geralmente exigem alto custo computacional. Além disso, essa solução considera apenas a utilização de veículos automotores.

O trabalho de Vu, Ng e Bolingbroke [78] propõe um modelo baseado no SIG ArcGIS para roteamento de veículos de tração humana e automotores, minimizando custos financeiros na coleta de resíduos na cidade de Hai Phong, no Vietnã. A proposta possui três componentes: localização-alocação, para identificar a quantidade e a distribuição dos pontos de coleta no cenário; roteamento de veículos de tração humana, os quais recolhem os materiais das residências e depositam em pontos de coleta, e roteamento de veículos automotores de grande porte (caminhões), que passam pelos pontos de coleta reunindo os materiais coletados. As rotas geradas minimizaram a distância percorrida, tanto para veí-

culos de tração humana, quanto automotores, estabelecendo limites de distância máxima para cada veículo.

O trabalho de Ahmad et al. [2] propõe um sistema ideal de recomendação de rotas para veículos de coleta seletiva, de acordo com o perfil de uma determinada região. O objetivo desta pesquisa é minimizar a distância das rotas, o consumo de combustível do caminhão coletor e maximizar a quantidade de materiais recolhidos. Para tanto, emprega-se o algoritmo BAT, fundamentado no comportamento de caça de morcegos, AG e *Particle Swarm Optimization* (PSO), baseado com um enxame de partículas que se movem para encontrar um máximo global. Nesse sentido, realizou-se o processamento de dados reais de descarte de materiais da cidade de Jeju, na Coreia do Sul, para prever o perfil da região. Nos experimentos, o algoritmo BAT apresentou melhores resultados em termos de tempo de processamento, enquanto que o PSO demonstrou o menor custo total em relação ao consumo de combustível e recursos humanos empregados.

Kiziltas, Alaka e Eren [52] apresentam um modelo matemático para minimizar os custos da coleta de resíduos, diminuindo os custos dos caminhões e funcionários utilizados na coleta de resíduos, considerando o consumo de combustível, a distância total percorrida por todos os veículos automotores de uma frota e o tempo de trabalho dos coletores. Essa solução estabelece limite de capacidade dos veículos e de tempo de trabalho dos coletores, além de considerar a utilização de diferentes tipos de veículos automotores (frota heterogênea). Ademais, os pontos de coleta e os pesos dos materiais coletados em cada ponto são previamente conhecidos.

O trabalho [37] estima as vantagens econômicas e operacionais da inclusão de catadores no planejamento e gestão de RSU na cidade de La Paz, Bolívia. Utiliza-se o SIG QGIS para identificar os custos econômicos, assim como as rotas de coleta e transporte dos resíduos. Além disso, o QGIS permite introduzir a infraestrutura das ruas da área estudada, a localização das lixeiras coletivas e a quantidade de resíduos gerada por habitante. Como resultado, verificou-se que a inclusão dos catadores informais permite reduzir os gastos com gerenciamento de resíduos em cerca de 10%, aumentar a taxa de reciclagem em cerca de 3,5% e reduzir as distâncias percorridas por caminhões compactadores em cerca de 7%.

A Tabela 4.2 apresenta a comparação dos trabalhos relacionados e o algoritmo proposto neste trabalho, o CATADORA, a partir das seguintes características: ano de publicação, se considera a capacidade do(s) veículo(s) como parâmetro, se o trabalho minimiza a distância percorrida pelo(s) veículo(s), custos financeiros da coleta, se utiliza de dados de inclinação das vias para gerar rotas e a utilização de veículos de tração humana (Coleta Primária). As características abordadas nas propostas dos trabalhos relacionados foram marcadas na tabela com o símbolo (✓).

O algoritmo proposto tem como diferencial, em relação aos trabalhos relacionados, associar as informações sobre inclinação das vias e a utilização de veículos de tração humana para minimizar a potência empregada para movimentar o veículo.

Tabela 4.2: Comparação dos trabalhos sobre rotas de coleta de resíduos

Trabalho	Ano	Capacidade	Distância	Custos	Inclinação	Coleta Primária
Tavares et al. [76]	2009		✓	✓	✓	
Benjamim e B. [11]	2010	✓	✓			
Das e B. [28]	2015	✓	✓	✓		✓
Fujdiak et al. [39]	2016		✓	✓		
Khan e S. [51]	2016	✓	✓	✓		
Nowakowski et al. [62]	2017	✓	✓	✓		
Vu, Ng e B. [78]	2018	✓	✓	✓		✓
Ahmad et al. [2]	2020	✓	✓	✓		
Kiziltas et al. [52]	2020	✓	✓	✓		
Ferronato et al. [37]	2020	✓	✓	✓		
CATADORA	2022		✓		✓	✓

4.3 Resumo Conclusivo

Os trabalhos relacionados ao problema de roteamento de veículos abordam a construção de caminhos entre dois pontos utilizando, na maioria das vezes, métodos heurísticos. No contexto do roteamento de veículos para coleta de resíduos, os trabalhos estão, em sua maioria, focados no roteamento de veículos automotores. Além disso, buscam principalmente minimizar os custos financeiros da gestão de RSU, reduzir o consumo de combustível dos veículos automotores e diminuir a distância das rotas de coleta. Os trabalhos que consideram a utilização de veículos de tração humana não abordam o esforço físico dos catadores para movimentar o carrinho ao longo das rotas sugeridas. É necessário considerar essa métrica, uma vez que o trabalho dos catadores-carrinheiros é fundamental na coleta de Resíduos Sólidos Urbanos, especialmente nos países emergentes como o Brasil.

Capítulo 5

Algoritmo de Otimização de Rotas Personalizadas

Este capítulo descreve o algoritmo de sugestão de rotas CATADORA (CATADORA), desenvolvido a partir da combinação das estratégias de busca Dijkstra bidirecional, *Shortest Path Faster Algorithm* e A-star; heurísticas Vizinho mais Próximo, Inserção do mais Próximo e Inserção do mais Distante e das políticas propostas Política de Menor Trabalho, Política de Menor Impedância e Política de Menor Distância.

5.1 CATADORA

Os catadores-carrinheiros percorrem a rota para recolhimento de materiais recicláveis e reutilizáveis a pé, empurrando o veículo. Sendo assim, o trajeto pode ser extremamente cansativo ou demorado para o trabalhador. Nesse sentido, é importante proporcionar rotas otimizadas que minimizam o esforço físico ao longo do percurso, ou a distância dos trajetos.

O roteamento de veículos com múltiplos pontos de parada se divide em dois subproblemas. O primeiro está relacionado à ordenação dos pontos de coleta que devem ser visitados e o segundo está associado à criação de caminhos otimizados entre estes pontos, no mapa geográfico. Sendo assim, dado um conjunto de coordenadas geográficas que representam os pontos de coleta, o algoritmo CATADORA determina uma rota para que um catador-carrinheiro visite todos os pontos de coleta.

O algoritmo proposto utiliza um grafo no qual cada vértice representa um ponto de coleta, chamado Grafo de Ordenação. Além disso, implementa-se a Rede Geográfica para a construção de caminhos. Uma rede virtualiza cenários do mundo real com nós/vértices e *links*/arestas. No caso da Rede Geográfica, representa-se os segmentos de vias/ruas como *links* e as junções entre esses segmentos, como nós [12].

O algoritmo CATADORA personaliza as rotas de coleta de acordo com três políticas: PMT, PMI e PMD. Essas políticas são utilizadas para atualizar o peso dos *links* da Rede geográfica, de acordo com o trabalho empregado para empurrar o veículo ao longo do *link*, da inclinação e da distância, respectivamente. Nessa perspectiva, a política PMT minimiza o trabalho aplicado para empurrar o veículo ao longo do trajeto, a política PMI

evita vias que possuem aclives acentuados e a política PMD diminui a distância total da rota.

A atualização do peso dos *links* da Rede Geográfica utilizando as políticas influencia na criação de caminhos e na ordenação dos pontos de coleta. Apesar da interdependência da rede e do Grafo de Ordenação para a definição da rota, a implementação da rede representando o cenário geográfico e do grafo representando os pontos de coleta possibilita que abordagens diferentes sejam utilizadas para a definição de uma única rota. Sendo assim, utiliza-se uma heurística no Grafo de Ordenação dos pontos de coleta e uma estratégia de busca para a criação de caminhos na Rede Geográfica. Com isso, é possível fazer diferentes combinações de heurísticas, estratégias de busca e políticas para gerar rotas otimizadas.

5.2 Implementação da Rede Geográfica

O algoritmo CATADORA gera a Rede Geográfica a partir de um conjunto de coordenadas geográficas dos pontos de coleta, inicial e final que fazem parte da rota. Nesse conjunto, a primeira coordenada representa o ponto de início e a última, o destino final (local de disposição dos materiais), que pode ou não ser igual ao ponto inicial. As outras coordenadas representam os pontos de parada. Com base nisso, o algoritmo proposto delimita uma área retangular com os valores mínimos e máximos de latitude e longitude do conjunto de coordenadas, estabelecendo uma margem que acrescenta o valor de 0.01 nas coordenadas.

A área delimitada é utilizada para obtenção de dados geográficos do *Open Street Map*, que é um mapa colaborativo livre para utilização e edição [46]. Os principais elementos do *Open Street Map* são os nós, caminhos e relações. Os nós representam pontos específicos no espaço e possuem dados de latitude e longitude; os caminhos são listas ordenadas de 2 a 2 mil nós e são utilizados para representar recursos lineares, como segmentos de ruas e rios e as relações documentam o relacionamento entre dois ou mais elementos, os quais podem ser nós, caminhos ou outras relações [56].

No *Open Street Map* existem nós que representam estabelecimentos ou residências e vértices que compõem caminhos. Nesse sentido, utiliza-se a ferramenta OSMnx para baixar os dados de vértices que compõem caminhos, descartando o outro tipo de nó. Essa ferramenta possibilita modelar, virtualizar, e analisar redes de ruas do mundo real. Nessa perspectiva, o OSMnx gera a Rede Geográfica definindo os segmentos de ruas como *links* e as junções entre esses segmentos como nós [12]. Além disso, utiliza-se os dados disponíveis no *Open Street Map* para configurar os *links*, tais como velocidade máxima das vias, o tipo de pavimentação e a distância dos segmentos das ruas.

Após baixar os dados do *Open Street Map* e gerar a Rede Geográfica, adiciona-se os pontos de coleta, inicial e final na rede. As coordenadas dos pontos de parada podem representar nós existentes na rede ou pontos próximos a um caminho definido no *Open Street Map*. Nesse sentido, é possível verificar o elemento que cada coordenada representa no mapa a partir de requisições enviadas para a Interface de Programação de Aplicações (API) pública do *Open Street Map*. Se a coordenada representa um nó já existente na rede,

ou seja, se está localizada em uma junção de segmentos de ruas, assume-se o identificador (*id*) do nó como um ponto de parada. Se a coordenada não representa um nó já existente e está localizada próximo a um caminho, a API retorna o identificador desse caminho.

Os pontos de coleta representam a localização do usuário que deseja descartar materiais recicláveis. Então, dificilmente a coordenada representa uma junção de segmentos de ruas do *Open Street Map*. Geralmente, a localização é de uma casa ou edifício. A Figura 5.1(a) apresenta um exemplo de ponto de parada e o respectivo caminho mais próximo.

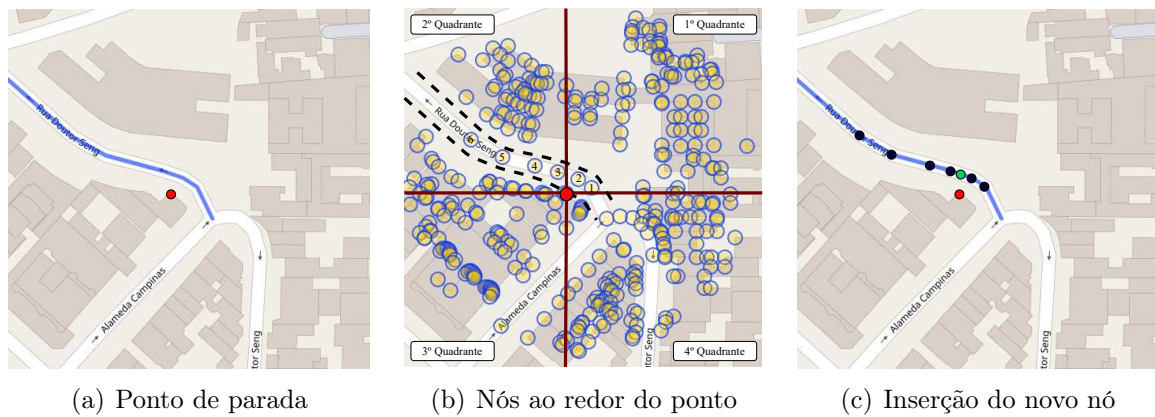


Figura 5.1: Ilustração do processo de inserção do ponto de parada na Rede Geográfica

Quando o ponto de parada representa uma casa ou edifício, é preciso inserir um vértice no caminho, próximo à coordenada do ponto de coleta, o qual representa o local que o(a) catador(a)-carrinheiro(a) deve parar e coletar os materiais. Além de inserir o vértice, é preciso identificar quais são os nós mais próximos, conectá-lo e torná-lo acessível na Rede Geográfica. Para isso, verifica-se os nós que compõem o caminho mais próximo. Em seguida, o algoritmo classifica a localização dos nós que fazem parte do caminho de acordo com 4 quadrantes. Nesse sentido, a Figura 5.1(b) apresenta os nós existentes ao redor do ponto, identifica os nós que fazem parte do caminho mais próximo e demonstra a divisão de quadrantes.

Para identificar os nós adjacentes ao novo vértice do ponto de parada, verifica-se os conjuntos de latitudes e longitudes dos nós que fazem parte do *link* mais próximo. Se o conjunto de latitudes apresenta maior variância do que o conjunto de longitudes, considera-se o *link* como horizontal. Nesse caso, identifica-se os dois nós mais próximos da coordenada do ponto de parada nos lados direito (1° e 4° quadrantes) e esquerdo (2° e 3° quadrantes). Se a variância do conjunto de longitudes for maior, define-se o *link* como vertical e observa-se os dois nós mais próximos da coordenada, considerando o lado de cima (1° e 2° quadrantes) e o lado de baixo (3° e 4° quadrantes).

O método dos quadrantes conecta o novo vértice que representa o ponto de coleta a dois nós, que estão de lados opostos. No exemplo da Figura 5.1(b), considera-se os lados direito e esquerdo para identificar os vértices adjacentes, que são o “2” e o “3”. Sendo assim, a Figura 5.1(c) apresenta a inserção do novo nó, configurado como o ponto de parada, na cor verde.

A Figura 5.2(a) apresenta um exemplo de caminho com formato irregular. Além disso, a Figura 5.2(a) demonstra que o método dos quadrantes não funciona nesse tipo de caso,

pois um dos lados opostos ao ponto de parada não tem vértices. Sendo assim, o algoritmo faz uma busca na área ao redor da coordenada do ponto de coleta e armazena os dois nós mais próximos ao ponto de parada. O novo vértice do ponto de coleta é inserido no caminho mais próximo, entre os dois nós encontrados.

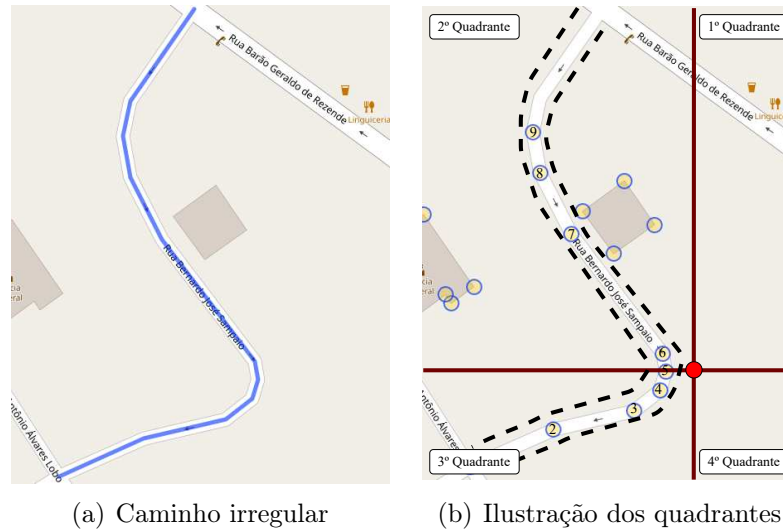


Figura 5.2: Aplicação do método dos quadrantes em um caminho irregular

Após adicionar os pontos de parada na Rede Geográfica, adiciona-se as informações de velocidade máxima das vias que não tem essa informação cadastrada no *Open Street Map*. Esses dados são inseridos baseados no tipo de via (terciária, secundária, primária, dentre outras.), seguindo os padrões de trânsito brasileiro. Além disso, configura-se as vias com velocidades máximas de 40 Km/h ou menos como bidirecionais, para possibilitar que os catadores-carrinheiros percorram esses caminhos nos dois sentidos. Em seguida, os dados de altitude de todos os nós da Rede Geográfica são adicionados na rede.

Os dados de altitude são obtidos a partir do Banco de Dados Geomorfométricos do Brasil. Esse banco de dados faz parte do projeto Topodata, o qual fornece dados abertos de todo território nacional: altitude, declividade, orientação de vertentes, curvaturas, forma do terreno, dentre outros. Os dados de altitude estão estruturados em arquivos GeoTiff com 1° de latitude por 1,5° de longitude, nomeados e documentados de forma padronizada, disponíveis no *site* do Topodata [77]. Sendo assim, o algoritmo CATADORA verifica quantos arquivos GeoTiff são necessários para obter os dados de elevação de toda a área delimitada do cenário geográfico. Se for necessário mais de um arquivo GeoTiff, cria-se um mosaico reunindo os dados de altitude em um único arquivo. Depois, o algoritmo faz a leitura do arquivo GeoTiff, identificando os valores de altitude de cada nó da Rede Geográfica. Com isso, configura-se o eixo z dos nós e os ângulos das arestas.

Depois de adicionar os pontos de parada, completar as informações faltantes sobre velocidade máxima das vias, configurar as ruas de tráfego em baixa velocidade como bidirecionais e determinar a altitude dos nós, a Rede Geográfica está pronta para ser utilizada na definição de rotas. Sendo assim, as estratégias de busca são utilizadas para construir caminhos de menor custo possível com base na Rede Geográfica. Nessa perspectiva, o custo total de uma rota partindo de um ponto a outro na Rede Geográfica se dá pela

soma dos custos dos *links* que compõem esse trajeto.

A Figura 5.3 apresenta o diagrama de atividades da etapa de construção da Rede Geográfica utilizando o algoritmo CATADORA, de acordo com as etapas descritas: identificar latitudes e longitudes máximas e mínimas; construir a rede geográfica; adicionar os pontos de coleta; adicionar informações sobre velocidade máxima; configurar as vias como bidirecionais; configurar altitude dos nós e, por fim, calcular o peso das arestas.

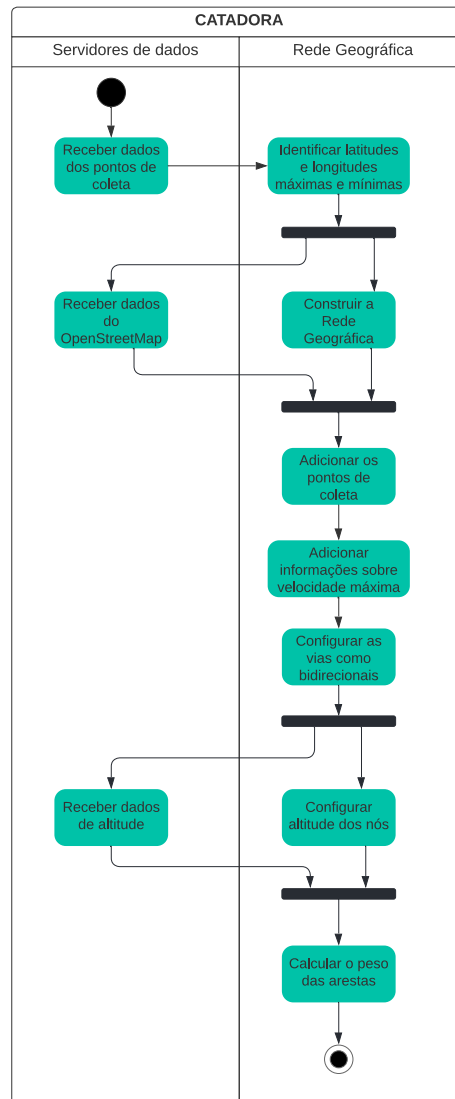


Figura 5.3: Diagrama de atividades da construção da Rede Geográfica

Os custos associados aos *links* da Rede Geográfica são calculados de acordo com as políticas PMT, PMI ou PMD, as quais minimizam o trabalho para empurrar o veículo, evitam aclives e declives acentuados e diminuem a distância dos trajetos, respectivamente.

5.3 Política de Menor Trabalho

A política PMT calcula os pesos associados aos *links* da Rede Geográfica de acordo com o trabalho necessário para empurrar o veículo em cada segmento de rua. O conceito físico

de Trabalho é de transferência de energia para/de um objeto. Nesse sentido, o trabalho total realizado por uma força sobre um objeto é igual à variação da energia cinética, a qual está relacionada ao estado de movimento de um objeto. Quanto maior a velocidade de movimento do corpo, maior sua energia cinética [47].

A energia cinética associada ao movimento de um corpo de massa m e velocidade v , considerando velocidades muito menores do que a velocidade da luz, é calculada por meio da Equação 5.1.

$$K = \frac{mv^2}{2} \quad (5.1)$$

Onde m representa a massa do objeto, v a velocidade e K a energia cinética. De acordo com a segunda lei de Newton, é possível relacionar a força à aceleração de acordo com a Equação 5.2.

$$F = ma \quad (5.2)$$

Onde m representa a massa do objeto e a , a aceleração. Visto que quando a força é constante a aceleração também é constante, é possível utilizar a Equação 5.3 para descrever a velocidade.

$$v^2 = v_0^2 + 2ad \quad (5.3)$$

Onde v representa a velocidade do objeto, v_0 a velocidade inicial, a é a aceleração e d o deslocamento. Reagrupando a Equação 5.3, obtemos a Equação 5.4.

$$ad = \frac{v^2 - v_0^2}{2} \quad (5.4)$$

Considerando que a aceleração é igual a força vezes a massa ($a = F/m$), é possível substituir a aceleração por F/m na Equação 5.4 e obter a Equação 5.5.

$$Fd = \frac{v^2m - v_0^2m}{2} \quad (5.5)$$

A Equação 5.5 demonstra que a variação da energia cinética é igual a Força vezes o deslocamento. Portanto, a Equação 5.6 é utilizada para calcular o Trabalho.

$$W = Fd \quad (5.6)$$

Onde W representa o trabalho, que tem a mesma unidade da energia cinética, o Joule, representado por J . A força aplicada para locomover um veículo a uma determinada velocidade é chamada de força de tração. Nesse caso, para realizar movimento é preciso vencer as resistências sobre uma pista. Dentre as forças resistentes ao movimento do veículo, é possível citar o arrasto aerodinâmico, a resistência ao rolamento e a força devido a ação da gravidade [75, 47].

A Força de arrasto aerodinâmico se dá devido os efeitos inerciais do ar sobre o veículo, uma vez que o ar “colide” com o veículo em movimento. A intensidade dessa força depende da densidade do ar, que é aproximadamente 1.22 Kg/m^3 ao nível do mar; da área frontal

do objeto, perpendicular ao movimento do ar em relação ao corpo; do coeficiente de arrasto, que depende do formato do objeto, e da velocidade do ar [75, 22, 47]. Com isso, tem-se a Equação 5.7.

$$F_a = \left(\frac{1}{2} * \rho * C * S * v^2 \right) \quad (5.7)$$

Onde ρ representa a densidade do ar, C corresponde ao coeficiente de arrasto, S a área frontal do veículo, v a velocidade do ar em relação ao corpo. O valor de C é 1 quando o formato do veículo é totalmente perpendicular ao movimento do ar, como por exemplo os veículos dos catadores-carrinheiros. Além disso, a área frontal desse tipo de carrinho geralmente é $1m^2$. Ademais, é comum considerar que a velocidade do ar em relação ao corpo (v) é igual a velocidade do veículo, consequentemente, o ar está parado em relação à via [75, 22].

Além da força de arrasto, tem-se a resistência ao rolamento, que está relacionada à perda de energia mecânica na roda. Essa resistência ocorre devido a deformação do pneu na região de contato com o solo. Dessa forma, depende da força normal à pista e do coeficiente de resistência adimensional. Visto que a força normal é calculada de acordo com a massa, a aceleração da gravidade e o ângulo de inclinação da via, a Equação 5.8 apresenta a força de resistência ao rolamento [75].

$$F_r = m * g * f_r \cos \theta \quad (5.8)$$

Onde m representa o valor da massa do veículo, g a constante de aceleração da gravidade, f_r o coeficiente de resistência ao rolamento e θ o ângulo de inclinação da via. Além disso, o valor do coeficiente f_r depende do tipo de roda do veículo e da pavimentação [75].

Quando a rua possui aclive/declive, a força peso do veículo é dividida em duas componentes, uma ortogonal que é anulada pela força normal e a componente paralela à via. Esta última atua como resistência em aclives e como adjuvante em declives. Nessa perspectiva, a Equação 5.9 apresenta a componente da força peso que influencia na força de tração [75, 47].

$$P = m * g * \sin \theta \quad (5.9)$$

Onde m representa o valor da massa do veículo, g a constante de aceleração da gravidade e θ o ângulo de inclinação da via. Além disso, destaca-se que essa componente se torna nula quando a rua é horizontal.

De acordo com o princípio fundamental da dinâmica, a força resultante em um corpo é dada pelo somatório de forças que agem sobre ele [47]. Nesse sentido, as forças de arrasto aerodinâmico, resistência ao rolamento e peso compõem a força resultante. Dessa forma, basta somá-las e multiplicá-las pela distância para obter o trabalho, como apresentado na Equação 5.10.

$$W = (F_a + F_r + P) * d \quad (5.10)$$

Onde W representa o trabalho, F_a a força de arrasto aerodinâmico, F_r a força de resistência ao rolamento, P a força peso e d o deslocamento. Além disso, o trabalho

é positivo quando a energia é transferida para o objeto e negativo quando a energia é transferida do objeto. Portanto, se a força possui uma componente vetorial no sentido do deslocamento, o trabalho é positivo. No sentido contrário, o trabalho é negativo [47].

A PMT utiliza o valor obtido por meio da Equação 5.10 como peso associado aos *links* da Rede Geográfica. Utilizar essa política proporciona a busca por caminhos que exigem o menor trabalho possível do catador-carrinheiro, visto que o custo minimizado pelos algoritmos de busca é o valor do trabalho. Como esse valor pode ser negativo e os algoritmos Dijkstra Bidirecional e A-star não aceitam valores negativos, subtrai-se todos os pesos dos *links* da rede pelo menor valor de trabalho obtido. Com isso, os pesos serão sempre positivos.

5.4 Política de Menor Impedância

A política PMI pode ser implementada no algoritmo CATADORA para gerar rotas que permitam aos catadores-carrinheiros evitarem aclives acentuados. Essa política foi idealizada para ser utilizada quando os trabalhadores possuem limitações físicas ou preferem caminhos com poucas subidas. Nesse sentido, propõe-se calcular o valor da Impedância (I) de acordo com a Equação 5.11.

$$I = \begin{cases} \theta^2 * d, & \text{se } \theta > 0 \\ \theta * (-1) * d, & \text{caso contrário} \end{cases} \quad (5.11)$$

Onde θ representa o ângulo da via e d , o tamanho do *link* (distância). A impedância obtida por meio da Equação 5.11 é configurada como peso associado a cada *link* da Rede Geográfica. Nesse caso, o ângulo é elevado ao quadrado quando é positivo, ou seja, quando a rua é uma subida. No entanto, quando o ângulo é negativo, representando uma descida, o ângulo é multiplicado por -1 . Portanto, o valor da impedância sempre será positivo, mas as subidas possuem penalidade maior. Além disso, tem-se um valor constante de peso dos *links*, pois não é preciso atualizar durante o percurso, diferentemente do que ocorre quando se utiliza a política PMT.

5.5 Política de Menor Distância

A política PMD minimiza a distância percorrida. Sendo assim, o peso dos *links* da Rede Geográfica é igual ao valor da distância/tamanho da aresta. Esse valor é obtido por meio do *Open Street Map* e incorporado na rede durante sua construção. Nesse caso, não é preciso fazer qualquer tipo de atualização do peso dos *links* durante a execução do algoritmo.

Após construir a Rede Geográfica e atualizar o peso dos *links*, é possível definir a ordem de visita dos pontos de coleta utilizando o Grafo de Ordenação, descrito a seguir.

5.6 Implementação do Grafo de Ordenação

O Grafo de Ordenação é utilizado para definir a ordem de visitação dos pontos de coleta. Sendo assim, é um grafo completo que configura todos os pontos de parada, além dos pontos inicial e final, como vértices interconectados. Visto que esse grafo é completo, cada vértice é adjacente a todos os outros. A implementação do Grafo de Ordenação foi realizada utilizando o pacote *Networkx*, o qual permite criar, manipular e analisar grafos [44]. Além disso, essa ferramenta possibilita a configuração dos dados associados ao grafo, como por exemplo os custos das arestas e as adjacências dos vértices.

Os custos das arestas do Grafo de Ordenação são calculados a partir dos custos dos trajetos entre os pontos de coleta, na Rede Geográfica. Nesse sentido, a Figura 5.4(a) ilustra um exemplo de Grafo de Ordenação, com dois pontos de coleta destacados. O custo da aresta entre os vértices 8 e 9 destacados é calculado de acordo com o custo total do caminho entre esses dois pontos de coleta, na Rede Geográfica. A Figura 5.4(b) apresenta a rota que conecta os vértices 8 e 9 destacada em vermelho na Rede Geográfica.

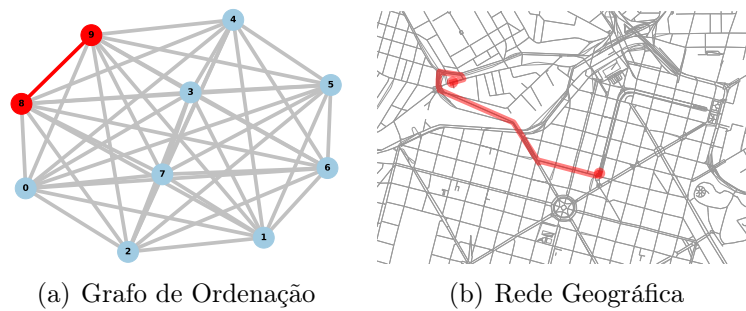


Figura 5.4: Ilustração do grafo e rede utilizados para determinação da rota

O algoritmo de definição de rotas para catadores-carrinheiros utiliza uma heurística para definir a ordem de visitação dos pontos de coleta a partir do Grafo de Ordenação, além de uma estratégia de busca para criar caminhos entre pontos de coleta na Rede Geográfica.

5.7 Algoritmos de Construção das Rotas

O algoritmo CATADORA implementou as heurísticas Vizinho mais Próximo (VP), Inserção do mais Próximo (IP) e Inserção do mais Distante (ID), bem como as estratégias de busca SPFA, Dijkstra Bidirecional e A-star. Os símbolos utilizados nesta Seção são os mesmos apresentados nas Tabelas 3.1, 3.2 e 3.3 do Capítulo 3. No entanto, o conjunto V representa os vértices do Grafo de Ordenação com exceção do vértice final e o conjunto N representa os vértices não visitados do Grafo de Ordenação, exceto vértices inicial e final. Além disso, adiciona-se outros símbolos, descritos na Tabela 5.1.

Considerando a aplicação da heurística VP no Grafo de Ordenação, é possível descrever a construção das rotas nas seguintes etapas: define-se o vértice inicial como o vértice atual e, em seguida, verifica-se o custo desse vértice até cada um dos nós adjacentes que não foram visitados ainda. Esse custo é calculado de acordo com uma estratégia de busca,

Tabela 5.1: Sumário de símbolos e significados

Símbolo	Significado
V	Conjunto de vértices do Grafo de Ordenação exceto v_{final}
N	Conjunto de vértices não visitados do Grafo de Ordenação (exceto v_{final} e $v_{inicial}$)
A_R	Matriz de Adjacências da Rede Geográfica
V_R	Conjunto de vértices da Rede Geográfica
C_R	Matriz de custos associados aos <i>links</i> da Rede Geográfica
C_G	Matriz de custos associados aos <i>links</i> do Grafo de Ordenação
$massa_i$	Massa inicial do veículo
$massa_v$	Massa atual do veículo
M	Conjunto de valores de massa dos materiais dos ponto de parada
$politica$	Política implementada (PMT, PMI ou PMD)
S	Vértices do grafo de Ordenação já visitados
$rota$	Conjunto de nós da Rede Geográfica que fazem parte da rota
n	Variável auxiliar do vértice atual (v_{atual})
K	Caminho mínimo entre dois pontos de coleta na Rede Geográfica
C_k	Custo do caminho K
K_{min}	Caminho mínimo entre o vértice atual e o ponto de coleta mais próximo

a qual escolhe o melhor caminho entre o par de vértices, na Rede Geográfica. Depois, encontra-se o vértice mais próximo do nó atual, ou seja, que possui o menor custo. O nó mais próximo passa a ser o vértice atual e, novamente, calcula-se os custos até os vértices que ainda não foram visitados. Esse processo é realizado até todos que os vértices do Grafo de Ordenação, ou seja, os pontos de coleta, sejam inseridos no caminho. Por fim, o nó final é inserido no caminho.

O Algoritmo 9 apresenta a combinação da heurística Vizinho mais Próximo com uma estratégia de busca, que pode ser SPFA, A-star ou Dijkstra bidirecional, as quais foram descritas no Capítulo 3. Os parâmetros de entrada do Algoritmo 9 são: vértice inicial ($v_{inicial}$), vértice final (v_{final}), conjunto de vértices do Grafo de Ordenação (V); conjunto de vértices não visitados do Grafo de Ordenação (N); matriz de custos do Grafo de Ordenação (C_G); matriz de custos da Rede Geográfica (C_R); conjunto de vértices da Rede Geográfica (V_R), conjunto de vértices adjacentes da Rede Geográfica (A_R); a política (PMT, PMI ou PMD); a massa inicial do veículo ($massa_i$) e o conjunto de massas dos materiais que serão coletados em cada ponto de parada (M).

De acordo com o Algoritmo 9, o conjunto de vértices visitados (S) é inicializado com o vértice inicial ($v_{inicial}$) (linha 1). Além disso, o vértice atual (v_{atual}) é inicializado de acordo com o vértice inicial (linha 2). A rota e o próximo nó a ser visitado (n) são inicializados com valor vazio (linhas 3-4). Ademais, a massa atual do veículo $massa_v$ é inicializada com o valor inicial da massa do veículo (linha 5). Por fim, a matriz de custos da Rede Geográfica (C_R) é atualizada a partir da massa atual do veículo e a política implementada (linha 6).

A função *atualiza_peso_links* calcula o peso das arestas da Rede Geográfica de acordo com a política utilizada. Se a política implementada for PMD, os custos associados aos *links* da rede passam a ser a distância de cada aresta. Se a política implementada for

Algoritmo 9: Combinação da heurística VP e Estratégia de Busca

Entrada: $v_{inicial}$, v_{final} , V , N , C_G , C_R , V_R , A_R , $política$, $massa_i$, M
Saída: rota

```

1  $S \leftarrow \{v_{inicial}\}$ 
2  $v_{atual} \leftarrow v_{inicial}$ 
3  $rota \leftarrow \phi$ 
4  $n \leftarrow \phi$ 
5  $massa_v \leftarrow massa_i$ 
6  $C_R \leftarrow atualiza\_peso\_links(C_R, massa_v, política)$ 
7 enquanto  $|S| < |V|$  faça
8    $C_{min} \leftarrow \infty$ 
9   para cada  $n_i$  em  $N$  faça
10      $K_{v_{atual}, n_i} \leftarrow estrategia\_busca(v_{atual}, n_i, V_R, A_R, C_R)$ 
11      $C_G[v_{atual}][n_i] \leftarrow custo\_caminho(K_{v_{atual}, n_i})$ 
12     se  $C_G[v_{atual}][n_i] < C_{min}$  então
13        $C_{min} \leftarrow C_G[v_{atual}][n_i]$ 
14        $K_{min} \leftarrow K$ 
15        $n \leftarrow n_i$ 
16    $rota \leftarrow rota \cup K_{min}$ 
17    $S \leftarrow S \cup \{n\}$ 
18    $v_{atual} \leftarrow n$ 
19    $N \leftarrow N - \{v_{atual}\}$ 
20   se  $política = PMT$  então
21      $massa_v \leftarrow atualiza\_massa\_veiculo(S, M, massa_i)$ 
22      $C_R \leftarrow atualiza\_peso\_links(C_R, massa_v, política)$ 
23    $K_{min} \leftarrow algoritmo\_busca(R, v_{atual}, v_{final}, massa_v, política)$ 
24    $rota \leftarrow rota \cup K_{min}$ 
25 retorna  $rota$ 

```

PMI, os custos são calculados a partir da Equação 5.11. Ademais, se a política for PMT, o peso das arestas da Rede Geográfica são calculados conforme a Equação 5.10.

No algoritmo, enquanto o número de vértices visitados (S) for diferente do número de vértices do Grafo de Ordenação (V) (linha 7), configura-se o custo mínimo para chegar ao nó mais próximo (C_{min}) igual a infinito (linha 8). Para cada vértice não visitado n_i (linha 9), identifica-se o caminho na Rede Geográfica (K) entre o nó atual (v_{atual}) e n_i . Este caminho é calculado por meio da estratégia de busca (linha 10), que pode ser Dijkstra Bidirecional, A-star e SPFA, descritos nos Algoritmos 6, 7 e 8, respectivamente. Em seguida, o custo total do caminho (K) é calculado por meio da função *custo_caminho*, que soma o peso de todas as arestas que compõem K . Se o custo do caminho for menor do que o custo mínimo (C_{min}), atualiza-se o custo mínimo (C_{min}), o menor caminho até o próximo nó (K_{min}) e o nó mais próximo (n) (linhas 13-15).

Depois de identificar o vértice do Grafo de Ordenação mais próximo do nó atual (n), adiciona-se o menor caminho (K_{min}) na rota (linha 16). Em seguida, o nó mais próximo (n) é acrescentado ao conjunto de vértices visitados (S) (linha 17), torna-se o vértice atual (linha 18) e é retirado do conjunto de vértices não visitados (N) (linha 19). Além disso, se a política utilizada for a PMT, a massa do veículo ($massa_v$) é atualizada de acordo com a função *atualiza_massa_veiculo* (linha 21). Esta função soma a massa inicial do veículo

($massa_i$) aos valores de massa dos materiais correspondentes a cada ponto de coleta (M), baseado nos pontos já visitados (S). Ademais, atualiza-se os pesos dos *links* da Rede geográfica, a partir do novo valor de massa do veículo (linha 22).

Após todos os vértices contidos em V estarem em S , cria-se o caminho do último nó adicionado (v_{atual}) até o nó final (linha 23). Acrescenta-se este caminho na rota (linha 24) e, por fim, o algoritmo retorna essa rota. Dessa forma, o Algoritmo 9 apresentou o processo de geração de rotas combinando a heurística VP e uma estratégia de busca. Destaca-se que a política PMT atualiza a massa do veículo e o peso dos *links* da Rede Geográfica cada vez que um nó é visitado no Grafo de Ordenação.

Além da heurística Vizinho mais Próximo, implementou-se também a Inserção do mais Próximo. O Algoritmo 10 apresenta a combinação da heurística IP e uma estratégia de busca. Esse algoritmo tem as mesmas entradas que o Algoritmo 9, apresentado anteriormente.

No Algoritmo 10, o vértice atual (v_{atual}) é inicializado a partir da função *proximo_vertice* (linha 1), descrita no Algoritmo 11. Essa função retorna o vértice mais próximo do início.

Após identificar o vértice mais próximo do vértice inicial, os vértices inicial e atual são inseridos no conjunto de vértices visitados (S) e a massa do veículo é atualizada (linha 2). A matriz de custos da Rede Geográfica (C_R) é atualizada de acordo com a massa atual do veículo (linha 3). Enquanto o número de nós do conjunto S for diferente do número de nós do conjunto V , o custo mínimo do menor caminho até o próximo vértice (C_{min}) é configurado como infinito (linha 4). Além disso, verifica-se qual nó pertencente ao conjunto dos não visitados (N) é mais próximo de qualquer nó já visitado, configurando este nó como v_{atual} (linhas 6-12).

Após definir o vértice atual, é preciso encontrar o índice para inserção desse vértice no conjunto S . Para isso, cria-se um laço de repetição que percorre todos os vértices já visitados (linha 14). Considerando s_i um nó pertencente ao conjunto S , o algoritmo gera uma rota na Rede Geográfica que inicia em s_i e finaliza no vértice atual ($K_{s_i, v_{atual}}$) (linha 15). O custo dessa rota é armazenado em $C_G[s_i][v_{atual}]$ (linha 16). Além disso, cria-se um caminho entre o vértice atual e o nó s_{i+1} , que é posterior a s_i no conjunto S ($K_{v_{atual}, s_{i+1}}$) (linha 17). Outra rota entre s_i e s_{i+1} também é calculada ($K_{s_i, s_{i+1}}$) (linha 19). Os custos desses caminhos são utilizados para calcular o custo total (linha 20), possibilitando encontrar o índice (p) para inserir o vértice atual no conjunto S (linhas 23-24).

Depois de encontrar o índice p , o vértice atual é inserido no conjunto de nós visitados (S) (linha 25) e retirado do conjunto de nós não visitados (N) (linha 26). Em seguida, se a política empregada é a PMT, atualiza-se a massa do veículo e a matriz de custos (C_R) dos links da Rede Geográfica (linhas 27-29). Após todos os vértices de V estarem no conjunto S , adiciona-se o vértice final (v_{final}) no conjunto S (linha 30).

A definição dos caminhos entre os vértices é realizada no final, quando os índices não mudarão mais, visto que a inserção de vértices nessa heurística acontece de forma dinâmica e os índices dos nós podem mudar durante a execução do algoritmo. Sendo assim, a função *define_caminhos* constrói a rota de acordo com os caminhos mínimos entre os vértices ordenados no conjunto S (linha 31). Por fim, o Algoritmo 10 retorna a rota.

Algoritmo 10: Combinação da heurística IP e Estratégia de Busca

Entrada: $v_{inicial}$, v_{final} , V , N , C_G , C_R , V_R , A_R , $politica$, $massa_i$, M
Saída: rota

```

1  $v_{atual} \leftarrow proximo\_vertice(N, C_G, v_{inicial})$ 
2  $S \leftarrow S \cup \{v_{inicial}, v_{atual}\}$ 
3  $C_R \leftarrow atualiza\_peso\_links(C_R, massa_i, politica)$ 
4 enquanto  $|S| < |V|$  faça
5    $C_{min} \leftarrow \infty$ 
6   para cada  $s_i$  em  $S$  faça
7     para cada  $n_i$  em  $N$  faça
8        $K_{s_i, n_i} \leftarrow algoritmo\_busca(s_i, n_i, V_R, A_R, C_R)$ 
9        $C_G[s_i][n_i] \leftarrow custo\_caminho(K_{s_i, n_i})$ 
10      se  $C[s_i][n_i] < C_{min}$  então
11         $C_{min} \leftarrow C[s_i][n_i]$ 
12         $v_{atual} \leftarrow n_i$ 
13    $C_{min} \leftarrow \infty$ 
14   para cada  $s_i$  em  $S$  faça
15      $K_{s_i, v_{atual}} \leftarrow algoritmo\_busca(s_i, v_{atual}, V_R, A_R, C_R)$ 
16      $C_G[s_i][v_{atual}] \leftarrow custo\_caminho(K_{s_i, v_{atual}})$ 
17      $K_{v_{atual}, s_{i+1}} \leftarrow algoritmo\_busca(v_{atual}, s_{i+1}, V_R, A_R, C_R)$ 
18      $C_G[v_{atual}][s_{i+1}] \leftarrow custo\_caminho(K_{v_{atual}, s_{i+1}})$ 
19      $K_{s_i, s_{i+1}} \leftarrow algoritmo\_busca(s_i, s_{i+1}, V_R, A_R, C_R)$ 
20      $C_G[s_i][s_{i+1}] \leftarrow custo\_caminho(K_{s_i, s_{i+1}})$ 
21      $C_{total} \leftarrow C[s_i][v_{atual}] + C[v_{atual}][s_{i+1}] - C[s_i][s_{i+1}]$ 
22     se  $C_{total} < C_{min}$  então
23        $C_{min} \leftarrow C_{total}$ 
24        $p \leftarrow i_s$ 
25    $S[p] \leftarrow v_{atual}$ 
26    $N \leftarrow N - \{v_{atual}\}$ 
27   se  $politica = PMT$  então
28      $massa_v \leftarrow atualiza\_massa\_veiculo(S, M, massa_i)$ 
29      $C_R \leftarrow atualiza\_peso\_links(C_R, massa_v, politica)$ 
30  $S \leftarrow S \cup \{v_{final}\}$ 
31  $rota \leftarrow define\_caminhos(S)$ 
32 retorna  $rota$ 

```

A heurística Inserção do mais Distante também foi implementada. Esta heurística é muito semelhante à Inserção do mais Próximo. Porém, modifica-se a linha 10 do Algoritmo 10 para escolher o vértice mais distante de qualquer nó do ciclo, ao invés do vértice mais próximo.

5.8 Implementação do Algoritmo em Aplicações

O algoritmo CATADORA pode ser utilizado em aplicações reais de geração de rotas utilizando servidor de mapas e de ofertas, como ilustra o esquemático da Figura 5.5. Como exemplos de servidores de mapas, tem-se o *Open Street Map* e o *Google Maps*, os quais fornecem dados de coordenadas geográficas, além de informações adicionais como distância

Algoritmo 11: função `proximo_vertice()`

Entrada: $N, C_G, v_{inicial}$
Saída: v_{atual}

```

1  $C_{min} \leftarrow \infty$ 
2 para cada  $n_i$  em  $N$  faça
3    $K_{v_{inicial}, n_i} \leftarrow \text{algoritmo\_busca}(v_{inicial}, n_i, V_R, A_R, C_R)$ 
4    $C_G[v_{inicial}][n_i] \leftarrow \text{custo\_caminho}(K_{v_{inicial}, n_i})$ 
5   se  $C_G[v_{inicial}][n_i] < C_{min}$  então
6      $C_{min} \leftarrow C_G[v_{inicial}][n_i]$ 
7      $n \leftarrow n_i$ 
8  $v_{atual} \leftarrow n$ 
9 retorna  $v_{atual}$ 

```

dos segmentos de vias, tipo de pavimentação, tipo de via (primária, secundária, terciária, dentre outras). Os servidores de mapa implementados no algoritmo CATADORA foram o *Open Street Map* e o Banco de Dados Geomorfométricos do Brasil.



Figura 5.5: Ilustração do esquema de aplicação do algoritmo

Os dados vinculados ao servidor de ofertas são utilizados para definir os pontos de coleta a partir da disponibilidade de resíduos recicláveis, de acordo com as plataformas de anúncios de materiais. Como exemplo de plataforma de anúncio que pode possuir servidor de ofertas, é possível mencionar o Cataki [6], descrito no Capítulo 2. Visto que essa plataforma permite que os usuários descrevam as características dos materiais que querem descartar, os catadores podem verificar previamente a quantidade, massa, tipo e localização dos materiais recicláveis disponíveis para recolhimento na agenda de coleta. A partir desses dados, é possível calcular a carga dos resíduos que serão coletados e definir rotas utilizando o algoritmo CATADORA.

5.9 Resumo Conclusivo

O algoritmo de sugestões de rotas CATADORA constrói a Rede Geográfica e o Grafo de Ordenação utilizando as ferramentas livres Networkx e OSMnx; implementa as estratégias de busca SPFA, Dijkstra bidirecional e A-star; elabora as heurísticas de geração de rotas VP, IP e ID. Além disso, propõe e implementa as políticas que definem o peso associado aos *links* da Rede Geográfica PMD, PMT e PMI. O código do algoritmo CATADORA foi desenvolvido a partir da linguagem Python e está disponível no GitHub ¹.

¹<https://github.com/vivirodrigues/rota-de-coleta>

Capítulo 6

Avaliação de Desempenho do Algoritmo

Este capítulo apresenta os cenários das simulações realizadas para avaliar o algoritmo proposto, os resultados das simulações de acordo com a potência média e instantânea empregada para empurrar o veículo nas rotas geradas, a distância percorrida e o tempo de trajeto. Além disso, avalia-se o tempo de processamento para gerar as rotas a partir das estratégias de busca e heurísticas aplicadas.

6.1 Cenários de Simulação

A avaliação das heurísticas, estratégias de busca e políticas implementadas no algoritmo proposto se dá por meio de simulações computacionais. Sendo assim, os cenários de simulação utilizam dados geográficos de áreas das cidades de São Paulo e Belém, obtidos por meio do *Open Street Map* [56]. Ilustra-se a imagem do mapa correspondente à área da cidade de São Paulo na Figura 6.1(a) e a Rede Geográfica construída com base nesta área, na Figura 6.1(b), demonstrando informações de altitude normalizada. A cartografia é disponibilizada pelo *Open Street Map* sob a licença CC-BY-SA [56].

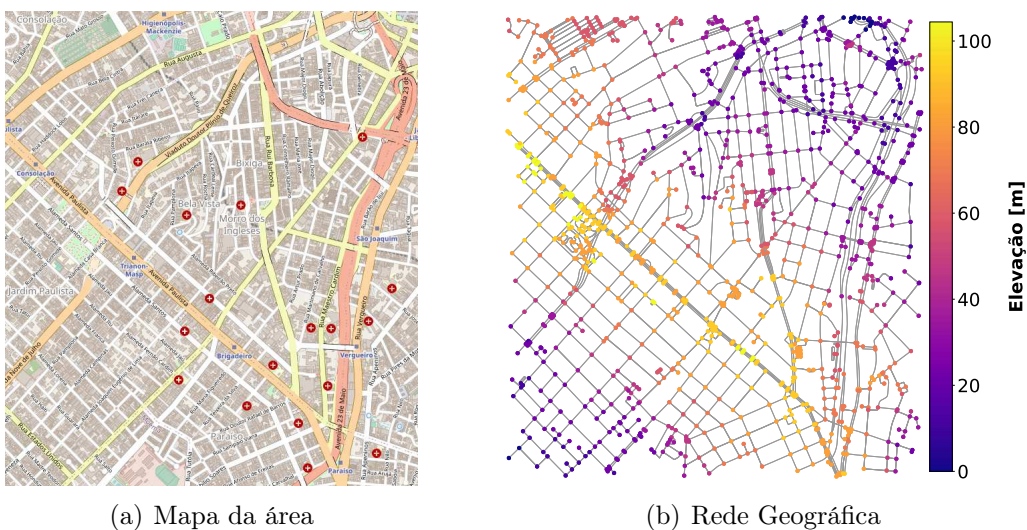


Figura 6.1: Área da cidade de São Paulo utilizada nos cenários de simulação

Observa-se na Figura 6.1(b) que a altitude da Rede Geográfica de São Paulo varia de

0 a 104 metros. Além disso, as diversas cores na ilustração demonstram que a variação da elevação está dispersa na rede, sendo que os pontos mais altos dessa área estão na Avenida Paulista.

A Figura 6.2(a) apresenta a imagem do mapa que corresponde à área de Belém. Além disso, a Figura 6.2(b) ilustra a Rede Geográfica dessa região, de acordo com a altitude normalizada, que varia de 0 a 28 metros. Comparando a área de Belém com a da cidade de São Paulo, esta última possui maior variação de altitude dos nós. Essas cidades foram escolhidas para avaliar e comparar os resultados do algoritmo proposto em solo plano (Belém) e com aclives acentuados (São Paulo).

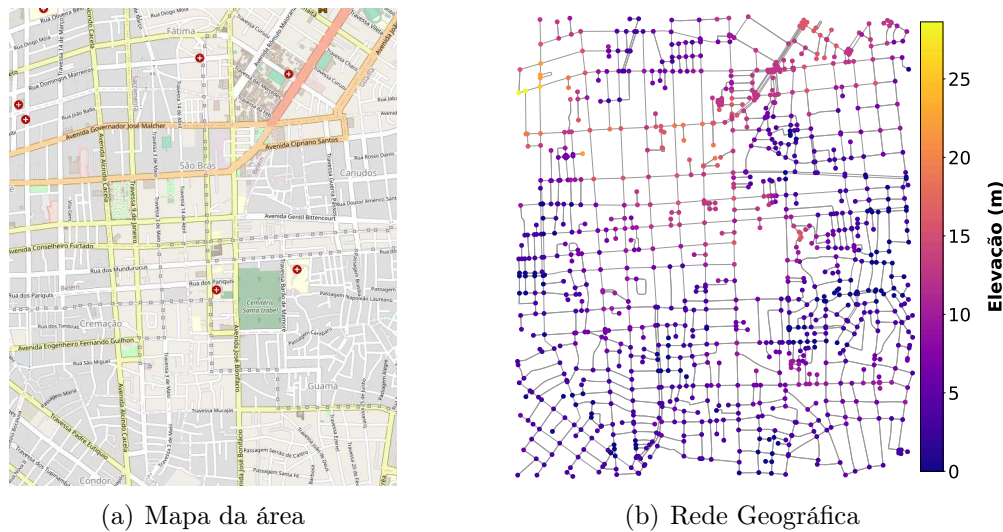


Figura 6.2: Área da cidade de Belém utilizada nos cenários de simulação

6.2 Simulador

As simulações computacionais para obtenção dos resultados foram implementadas em um computador com Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz, 4 núcleos, 2 *threads* por núcleo, memória 8 GB e Linux Ubuntu 20.04.1 LTS. Além disso, utilizou-se o simulador *Simulation of Urban MObility*, versão 1.8.0. Este simulador de código aberto permite criar cenários, analisá-los, gerar padrões de mobilidade e simular estratégias de gerenciamento de tráfego [55].

O SUMO possui um editor gráfico de redes chamado *NETEDIT*, que permite criar cenários adicionando vias, nós, sinais de trânsito, pontos de ônibus, dentre outras características. Outra ferramenta para criar cenários no SUMO é o *NETCONVERT*, que possibilita importar redes de diferentes bancos de dados como o *Open Street Map* e é baseado em linhas de comando. Ademais, o *NETCONVERT* possibilita adicionar dados de altitude nos cenários de simulação por meio de mapas de altura como GeoTiff, *shapefiles*, dentre outros.

Os arquivos GeoTiff aceitos pelo simulador devem ter a etiqueta Int16, a qual significa que os dados armazenados no arquivo são do tipo inteiro de 16 bits. No entanto, os arquivos GeoTiff disponibilizados pelo Banco de Dados Geomorfométricos do Brasil possuem

dados do tipo ponto flutuante de 32 bits. Sendo assim, utilizou-se a ferramenta *Geospatial Data Abstraction Library* (GDAL) para converter os dados dos planos de informação em formato GeoTiff para inteiros de 16 bits.

Os dados do *Open Street Map* para criação dos cenários de São Paulo e Belém no simulador foram baixados em arquivos no formato *eXtensible Markup Language* (XML), a partir da API *Overpass*. Para isso, utilizou-se as informações de latitudes e longitudes máximas e mínimas das Redes Geográficas. Após o *download* dos dados geográficos dos cenários e conversão do formato de dados dos arquivos GeoTiff, a aplicação *NET-CONVERT* foi implementada para gerar os cenários, que ficaram muito parecidos com as Redes Geográficas. É importante destacar também que essa aplicação configura não apenas a altitude dos nós, como também os ângulos de inclinação das ruas do cenário.

Os pontos de parada foram adicionados nos cenários do SUMO para torná-los idênticos as redes. Além disso, o algoritmo proposto fez uma análise dos cenários e as respectivas Redes Geográficas associadas, para diagnosticar possíveis diferenças e corrigi-las, tendo como base as Redes Geográficas. Sendo assim, a Figura 6.3(a) apresenta o cenário de São Paulo implementado no simulador e a Figura 6.3(b), o cenário de Belém.



Figura 6.3: Implementação dos cenários de simulação no SUMO

De acordo com a Figura 6.3(a) e a Figura 6.3(b), é possível identificar os cenários implementados, bem como a interface gráfica do simulador. O SUMO também permite executar as simulações sem interface gráfica, por meio de linhas de comando. Além disso, esse simulador possibilita controlar a simulação por meio da *Traffic Control Interface* (TraCI), que é uma ferramenta baseada em arquitetura cliente/servidor. A TraCI estabelece uma comunicação que permite utilizar a linguagem Python para interagir com o SUMO: ler dados da simulação, manipular estados de objetos, veículos e rotas. Nessa perspectiva, utilizou-se essa ferramenta para gerenciar e acompanhar as simulações nos cenários descritos.

6.3 Parâmetros das Simulações

As simulações possuem dois cenários geográficos e diferentes números de pontos de coleta. Para todos os cenários, considerou-se a pré-definição da localização dos pontos de coleta e da massa dos materiais recolhidos em cada ponto, simulando coletas agendadas previamente. Sendo assim, foram implementados 12, 18 e 24 pontos de parada, considerando as áreas de Belém e São Paulo. Nesse caso, o primeiro e último pontos de parada representam os pontos inicial e final da rota. A Tabela 6.1 demonstra todos os parâmetros utilizados nas simulações.

Tabela 6.1: Parâmetros utilizados nas simulações

Parâmetro	Valor
Número de pontos de parada por simulação	12, 18 e 24
Massa inicial do veículo	100 Kg
Incremento máximo de massa do veículo	5 Kg
Média das coordenadas (São Paulo)	(−46, 64842, −23, 56363)
Média das coordenadas (Belém)	(−48, 47000, −1, 46000)
Desvio padrão das coordenadas	0,002
Velocidade máxima do(a) catador(a)-carrinheiro(a)	3,6 Km/h
Aceleração máxima do(a) catador(a)-carrinheiro(a)	$1,5m/s^2$
Desaceleração máxima do(a) catador(a)-carrinheiro(a)	$2m/s^2$
Área frontal do veículo	$1m^2$
Densidade do ar	$1,2\text{ Kg}/m^3$
Coeficiente aerodinâmico	1
Aceleração da gravidade	9,80665
Número de sementes de aleatoriedade	30
Intervalo de confiança	95%

De acordo com a Tabela 6.1, a massa inicial do veículo foi configurada como 100 Kg e esta massa é incrementada com valor máximo de 5 Kg em cada ponto de coleta, de forma pseudo-aleatória. Essa atualização da massa do veículo representa os materiais recicláveis recolhidos e armazenados no carrinho. Além disso, as coordenadas x e y dos pontos de coleta seguem uma distribuição gaussiana com desvio padrão 0,002, média (−46, 64842, −23, 56363) nos cenários de São Paulo e média (−48, 47000, −1, 46000), nos cenários de Belém.

A velocidade máxima do veículo nas simulações é de 3,6 Km/h, que representa velocidade de uma pessoa caminhando. A velocidade instantânea do veículo é definida de acordo com o algoritmo KraussPS do simulador SUMO, o qual considera os aclives e declives para defini-la. Além disso, utilizou-se parâmetros estabelecidos pelo simulador para a classe pedestre, tais como a aceleração máxima de $1,5m/s^2$ e desaceleração máxima de $2m/s^2$.

As simulações consideraram a área frontal do veículo de $1m^2$, densidade do ar sendo $1,2\text{ Kg}/m^3$, coeficiente aerodinâmico 1 (a área frontal do veículo faz um ângulo de 90 graus em relação ao solo), e aceleração da gravidade 9,80665. Cada simulação foi executada 30 vezes utilizando diferentes sementes de aleatoriedade. Por fim, utilizou-se intervalo de confiança de 95% para gerar os resultados. A avaliação dos resultados obtidos

foi realizada a partir das métricas: potência aplicada para empurrar o veículo, distância total do trajeto, tempo de percurso e tempo de processamento para gerar as rotas.

6.4 Análise da Potência Aplicada no Veículo

Esta Seção responde as questões de pesquisa 1. *Qual o impacto de evitar aclives e declives acentuados na otimização da potência empregada para empurrar o veículo de tração humana?*, 2. *Como a minimização da distância das rotas afeta a potência empregada para empurrar o veículo ao longo do percurso?* e 3. *Como as heurísticas utilizadas para ordenação dos pontos de coleta influenciam na otimização da potência empregada para empurrar o veículo de tração humana?*

A potência aplicada no veículo depende da força de tração e da velocidade empregada pelos catadores-carrinheiros. A força de tração pode ser calculada somando a força de arrasto aerodinâmica, a resistência ao rolamento e a força de ação da gravidade [75]. Sendo assim, a Equação 6.1 é utilizada para calcular a potência instantânea em cada segundo de simulação.

$$P = \left[(m * g * f_r \cos \theta) + (m * g * \sin \theta) + \left(\frac{1}{2} * \rho * C * S * v^2 \right) \right] * v \quad (6.1)$$

Onde P representa a potência instantânea, m o valor da massa do veículo, g a constante de aceleração da gravidade, f_r o coeficiente de resistência ao rolamento, θ o ângulo de inclinação da via, ρ a densidade do ar, C corresponde ao coeficiente de arrasto, S a área frontal do veículo e v a velocidade instantânea do veículo. Os coeficientes de resistência ao rolamento foram definidos de acordo com o tipo de pavimentação das vias (asfalto, concreto, pedras, dentre outros) [22].

A primeira componente da equação corresponde à força de resistência ao rolamento, a segunda componente representa a força de ação da gravidade e a terceira componente, a força de arrasto aerodinâmico. Considera-se que a velocidade do ar em relação ao corpo (v) é igual a velocidade do carrinho, consequentemente, o ar está parado em relação à via [75].

Os trajetos de coleta são mais cansativos quando se emprega altas potências, visto que é necessário aplicar esforço físico para locomover os veículos. Sendo assim, a Figura 6.4(a) apresenta a potência média aplicada para locomover o veículo nos cenários de São Paulo e Belém. A potência média de cada simulação foi calculada de acordo com a média aritmética simples dos valores de potência instantânea. Além disso, as rotas geradas pelo algoritmo foram configuradas com 12 pontos de parada. Nesse caso, compara-se as rotas geradas aplicando as heurísticas VP, IP e ID, assim como as políticas PMT, PMI e PMD.

No cenário de São Paulo, a menor média de potência aplicada no veículo foi 38 W, resultado da rota gerada pelo algoritmo aplicando a heurística Inserção do mais Próximo e a política PMI. No cenário de Belém, a política PMT proporcionou rotas com os menores valores de potência média, enquanto os maiores valores foram obtidos utilizando a política PMD. Além disso, a menor média de potência aplicada foi 11,9 W, obtida utilizando a política PMT e as heurísticas IP e ID. Considerando ambos os cenários de São Paulo e Belém, as heurísticas VP, IP e ID alcançaram resultados semelhantes. As políticas

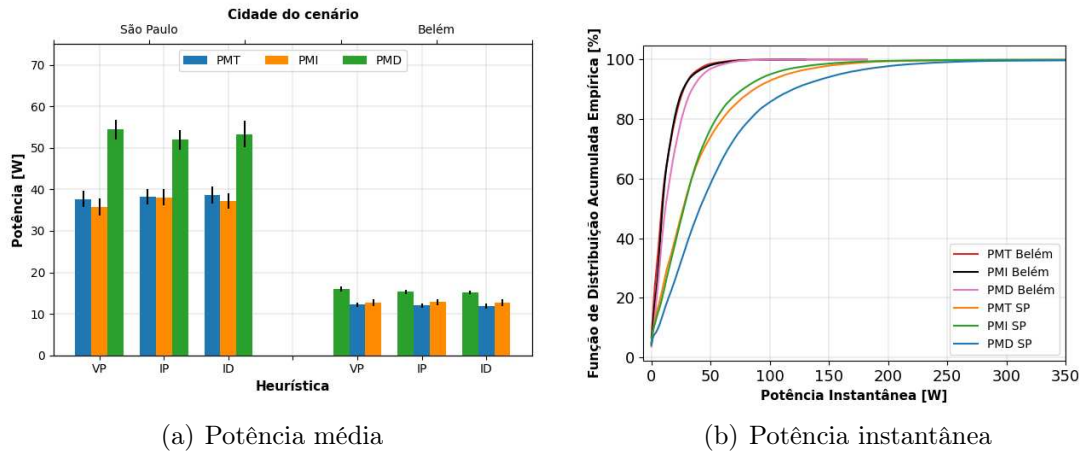


Figura 6.4: Potência aplicada no veículo nas rotas com 12 pontos de parada

PMT e PMI obtiveram valores de potência média parecidos entre si e a política PMD teve o pior desempenho quanto à minimização de potência. Isso se dá porque o objetivo da PMD é minimizar a distância, independentemente se o caminho escolhido tem aclives e/ou declives acentuados.

Verifica-se nos resultados apresentados na Figura 6.4(a) que os aclives e declives influenciaram na potência aplicada no veículo, pois os valores de potência média do cenário de Belém são inferiores em comparação com os resultados no cenário de São Paulo, considerando as rotas com 12 pontos de parada. Portanto, o cenário com maior variação de altitude exigiu maior potência aplicada no veículo e, consequentemente, esforço físico do(a) catador(a)-carrinheiro(a) para empurrar o veículo.

Além da potência média, é preciso analisar os valores de potência instantânea aplicada no veículo durante as simulações. Para isso, a Figura 6.4(b) apresenta a Função de Distribuição Acumulada (FDA) Empírica da potência instantânea aplicada no veículo. A FDA fornece a probabilidade de produzir um valor menor ou igual a x de potência instantânea para empurrar o veículo. A probabilidade de um valor igual a x ocorrer é calculada dividindo o número de vezes que x ocorre pelo tempo de simulação. Então, a FDA empírica de x é a soma das probabilidades de valores menores ou igual a x ocorrerem.

De acordo com os resultados da Figura 6.4(b), no cenário de São Paulo, as políticas PMT e PMI proporcionaram rotas com maior probabilidade de empregar potências inferiores a 100 W. No cenário de Belém, os valores de probabilidade acumulada se aproximam de 100% considerando as políticas PMT, PMI e PMD para valores acima de 50 W de potência aplicada no veículo. Comparando os resultados dos dois cenários, o de São Paulo proporcionou rotas com menor probabilidade da potência aplicada no veículo ser inferior a 100 W.

A Figura 6.5(a) apresenta a potência média aplicada para empurrar o veículo de tração humana nos cenários de São Paulo e Belém nas rotas de 18 pontos de parada. De acordo com os resultados apresentados, o algoritmo gerou rotas com média de potência mais alta utilizando a política PMD, em comparação com PMT e PMI. No cenário de São Paulo, a combinação da heurística IP e da política PMI aplicada no algoritmo gerou o melhor resultado, 38 W. No cenário de Belém, a rota gerada por meio da combinação da

heurística IP e da política PMT alcançou a menor média de potência aplicada no veículo, 12,4 W.

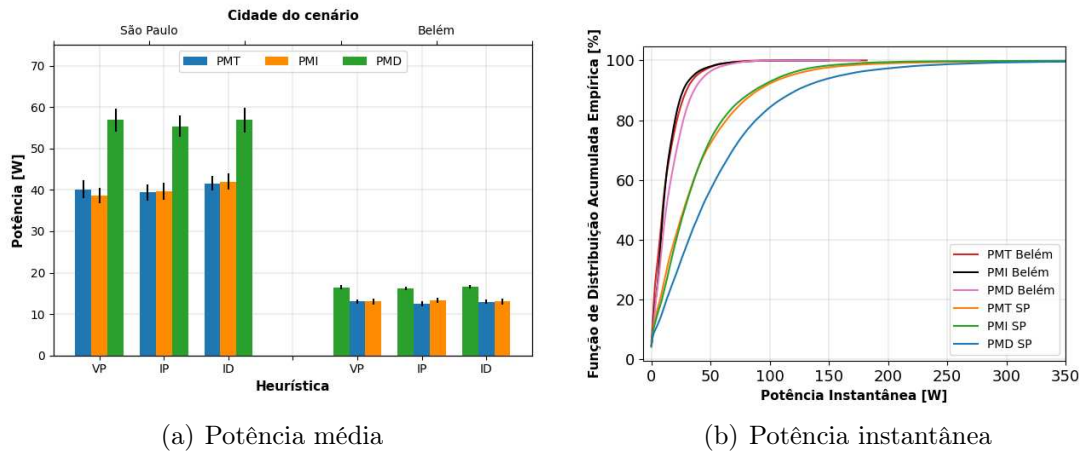


Figura 6.5: Potência aplicada no veículo nas rotas com 18 pontos de parada

A partir da Figura 6.5(b), é possível verificar que a FDA das rotas geradas a partir das políticas PMT, PMI e PMD está próxima de 100% considerando valores de potência instantânea menores ou iguais a 50 W. Ademais, no cenário de São Paulo, a política PMD proporcionou rotas com a menor probabilidade de aplicar potência instantânea menor ou igual a 150 W. Sendo assim, a utilização desta política pode gerar rotas mais cansativas para os catadores-carrinheiros.

As rotas com 24 pontos de parada alcançaram resultados com tendências semelhantes aos apresentados pelas rotas de 12 e 18 pontos de parada, de acordo com a Figura 6.6(a). Nesse caso, a aplicação da política PMD no algoritmo gerou rotas com os maiores valores de potência média. Além disso, os resultados das rotas geradas pelas políticas PMI e PMT foram semelhantes. Nesse sentido, o melhor desempenho quanto à minimização da média de potência no cenário de São Paulo foi de 41,7 W, obtido por meio da combinação da heurística IP e da política PMI. No cenário de Belém, o melhor resultado foi 13,3 W, a partir da combinação da heurística IP e da política PMT.

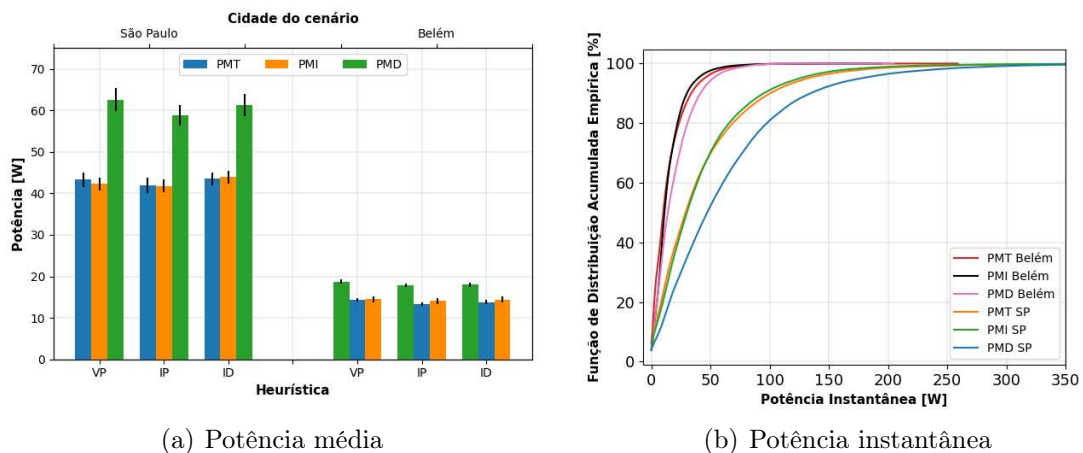


Figura 6.6: Potência aplicada no veículo nas rotas com 24 pontos de parada

A Figura 6.6(b) apresenta a FDA empírica das rotas com 24 pontos de parada. Nesse caso, as rotas geradas pelo algoritmo utilizando as heurísticas PMT, PMI e PMD tem 100% de probabilidade de exigirem potências instantâneas menores ou iguais a 100 W no cenário de Belém. No entanto, no cenário de São Paulo, as heurísticas PMT, PMI e PMD apresentam 89%, 91% e 80% de probabilidade de empregarem valores de potência instantânea menores ou iguais a 100 W. Demonstrando, assim, que o cenário com maior variação de altitude tem maior probabilidade de exigir potências instantâneas maiores do que 100 W, do que o cenário mais plano.

A FDA se aproxima de 100% para valores menores ou iguais a 200 W utilizando as políticas PMT e PMI no cenário de São Paulo, enquanto a PMD se aproxima de 100% a partir de 250 W. Com isso, os resultados da Figura 6.6(b) confirmam que as rotas geradas pela política PMD tem maior probabilidade de exigir o emprego de altas potências no veículo, visto que esta política não tem como objetivo minimizar o esforço físico desempenhado para empurrar veículos de tração humana.

A partir dos resultados apresentados, é possível verificar que a política PMI, a qual permite criar rotas evitando aclives e declives acentuados, minimizou a potência empregada para empurrar o veículo nas simulações, em comparação com a política PMD, que otimiza a distância das rotas. Nesse caso, a minimização da distância das rotas fez com que a potência aplicada no veículo fosse maior, nos dois cenários considerados. Além disso, a heurística IP apresentou predominantemente o melhor resultado nos dois cenários. No entanto, a diferença entre os resultados das heurísticas VP, IP e ID é pequena.

6.5 Tempo de Trajeto e Distância Total das Rotas

Esta Seção apresenta resultados que permitem responder à questão de pesquisa 4. *De que forma a otimização do trabalho empregado para empurrar os veículos nas rotas influencia na distância total dos percursos?*

Apesar da minimização do esforço físico empregado para empurrar os veículos de tração humana ser importante, o algoritmo propõe personalizar as rotas de acordo com os objetivos dos catadores. Sendo assim, a avaliação da distância e tempo total do trajeto permitem identificar a combinação de política e heurística para o algoritmo que possibilita gerar trajetos mais curtos e rápidos para os catadores-carrinheiros. Com base nos resultados das simulações, a Figura 6.7(a) apresenta a distância total das rotas de 12 pontos de parada e a Figura 6.7(b) demonstra os resultados do tempo necessário para percorrer todos os pontos.

De acordo com a Figura 6.7(a) e a Figura 6.7(b), nos cenários de São Paulo e Belém, a política PMD proporcionou a geração de rotas com menor tempo e distância de percurso, enquanto que as rotas construídas utilizando a política PMI foram as mais longas e demoradas. Além disso, o melhor desempenho quanto à minimização da distância e tempo se deu a partir da combinação da heurística ID e da política PMD: 3,4 quilômetros e 1,3 horas no cenário de São Paulo, 3,2 quilômetros e 1,2 horas no cenário de Belém. Sendo assim, a política PMD alcançou o objetivo de minimizar a distância percorrida nas rotas.

A Figura 6.8(a) apresenta a distância total percorrida nas rotas de 18 pontos de parada

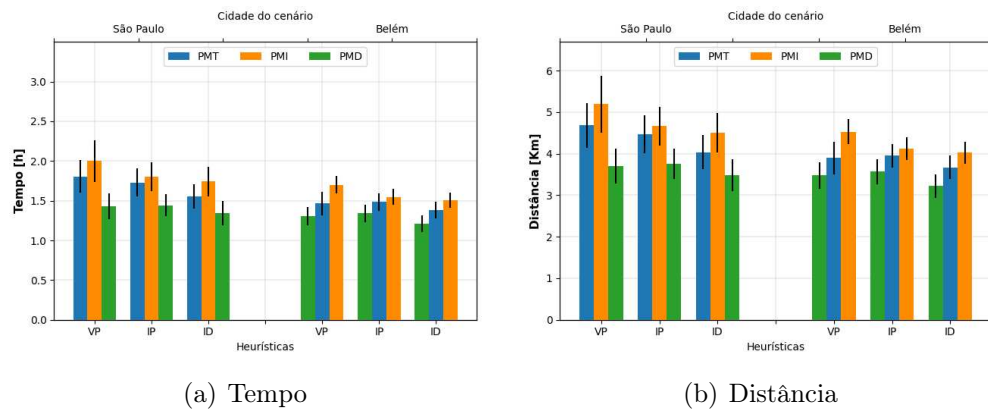


Figura 6.7: Tempo e distância percorridos nas rotas com 12 pontos de parada

e a Figura 6.8(b) demonstra o tempo necessário para percorrer todos os pontos.

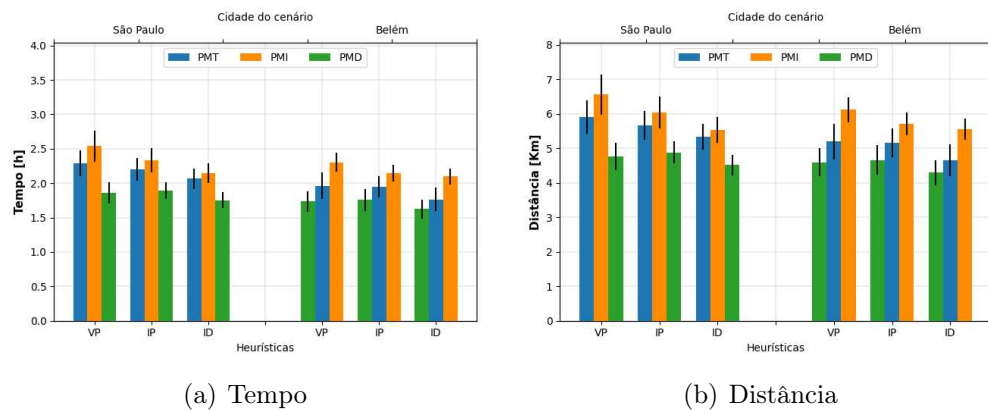


Figura 6.8: Tempo e distância percorridos nas rotas com 18 pontos de parada

Os resultados da Figura 6.8(a) e da Figura 6.8(b) demonstram que as rotas de 18 pontos de coleta geradas a partir da política PMD foram as mais rápidas e finalizaram com a menor distância total. As rotas geradas a partir da política PMI apresentaram maior distância percorrida e menor tempo de percurso. Ademais, os trajetos mais curtos e rápidos foram gerado utilizando a combinação da heurística ID e da política PMD: 1,3 horas de duração e 3,4 quilômetros de distância no cenário de São Paulo; 1,2 horas e 3,21 quilômetros no cenário de Belém.

A Figura 6.9(a) e a Figura 6.9(b) apresentam os resultados de distância e tempo de percurso, os quais seguiram a tendência apresentada nos resultados das rotas com 12 e 18 pontos de parada. Nos trajetos de 24 pontos, os melhores resultados quanto à minimização da distância e tempo foram alcançados utilizando a heurística ID e a política PMD. No cenário de São Paulo, a rota mais rápida foi de 2,2 horas, com distância percorrida de 5,6 quilômetros. No cenário de Belém, a rota mais rápido durou 2 horas, percorrendo 5,4 quilômetros.

Considerando os resultados de distância e tempo médio das rotas com 12, 18 e 24 pontos de parada, é possível verificar que o algoritmo do SUMO manteve a velocidade média próxima a 2,6 Km/h nas simulações. Isso acontece porque, apesar de levar em

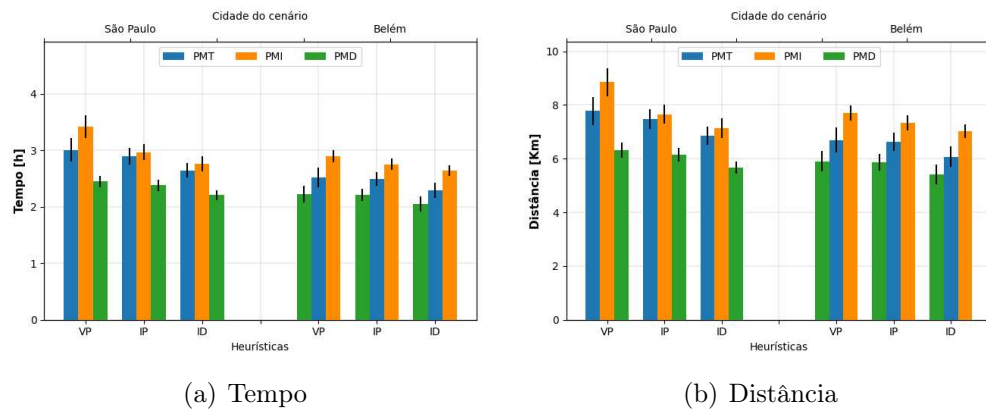


Figura 6.9: Tempo e distância percorridos nas rotas com 24 pontos de parada

consideração os aclives e declives para escolher a velocidade instantânea do veículo, o simulador escolhe velocidades instantâneas próximas à velocidade máxima configurada (3,6 Km/h). O comportamento do simulador na definição da velocidade do veículo de acordo com os aclives e declives foi analisado no trabalho de Sanches et al. [69].

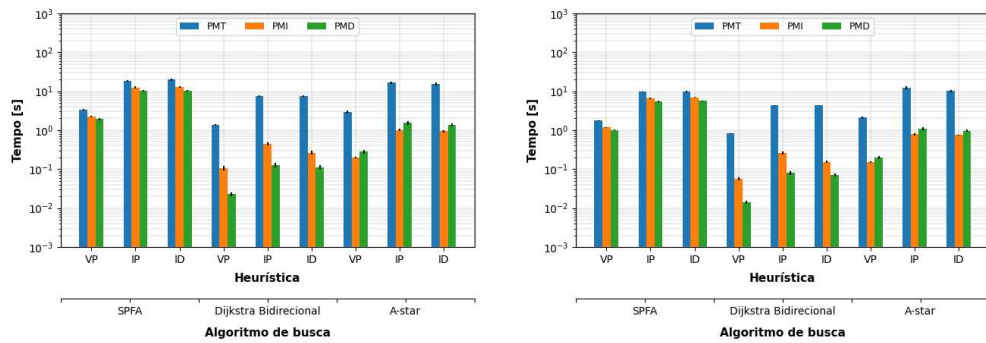
Os resultados apresentados nesta Seção permitem verificar que a implementação da política que minimiza o trabalho, PMT, tem como resultado rotas mais longas do que quando se implementa a política PMD. Porém, gera rotas mais curtas do que quando se emprega a política PMI.

6.6 Análise de Desempenho das Heurísticas, Estratégias e Políticas Utilizadas

Os resultados apresentados nesta Seção possibilitam responder à questão de pesquisa 5. *O quão computacionalmente eficientes são as combinações de estratégias de busca, heurísticas e políticas empregadas no algoritmo para solucionar o problema?*

O algoritmo de geração de rotas precisa ter o mínimo tempo de processamento possível para ser implementado em aplicações reais. Nessa perspectiva, o desempenho do algoritmo proposto depende da configuração da estratégia de busca, heurística e política implementadas. Sendo assim, a Figura 6.10(a) e a Figura 6.10(b) apresentam o tempo de processamento da geração de rotas de 12 pontos de parada considerando os cenários de São Paulo e Belém, respectivamente.

Os resultados do tempo de processamento para gerar rotas de 12 pontos de parada nos cenários de São Paulo e Belém demonstram que a PMT obteve o maior tempo de processamento entre as políticas. Isso acontece porque a PMT atualiza os pesos associados aos *links* da Rede Geográfica a cada iteração. Considerando a estratégia A-star, a implementação da política PMD fez com que esta estratégia precisasse de mais iterações para encontrar o caminho mínimo entre dois pontos. Por isso, o tempo de processamento da política PMD foi maior do que da PMI. A utilização da estratégia de busca SPFA exigiu maior tempo para gerar as rotas, em comparação com Dijkstra bidirecional e A-star. Ademais, o melhor desempenho foi alcançado utilizando a combinação da estratégia



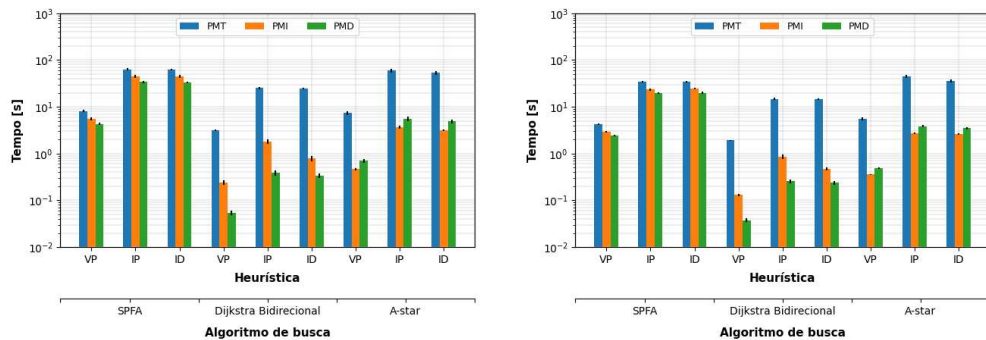
(a) São Paulo

(b) Belém

Figura 6.10: Tempo de processamento 12 pontos de coleta

de busca Dijkstra bidirecional, heurística Vizinho mais Próximo e política PMD: 0,023 segundos no cenário de São Paulo e 0,014 segundos no cenário de Belém.

A Figura 6.11(a) e a Figura 6.11(b) apresentam os tempos de processamento para geração de rotas de 18 pontos de parada nos cenários de São Paulo e Belém, respectivamente.



(a) São Paulo

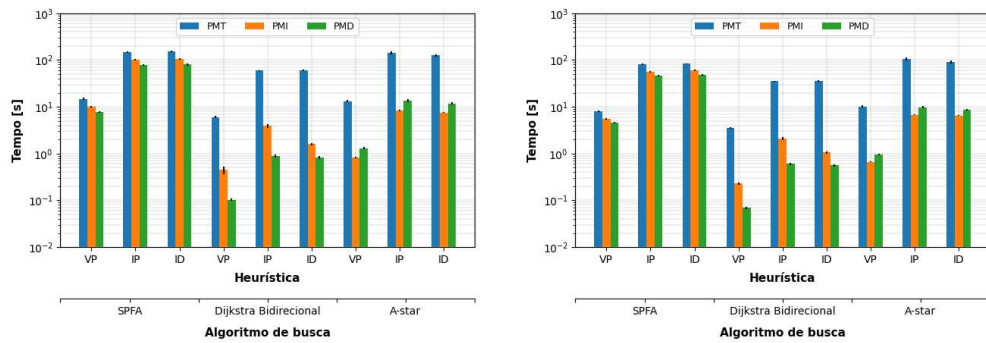
(b) Belém

Figura 6.11: Tempo de processamento 18 pontos de coleta

De acordo com os resultados apresentados na Figura 6.11(a) e na Figura 6.11(b), o emprego da política PMT proporcionou os maiores tempos de processamento. Ademais, o melhor desempenho foi alcançado utilizando a estratégia de busca Dijkstra bidirecional, a heurística VP e a política PMD, cujo valor é três ordens de magnitude menor do que o tempo de processamento mais alto, produzido a partir do emprego da estratégia de busca SPFA, heurística Inserção do mais Próximo e política PMT.

A Figura 6.12(a) e a Figura 6.12(b) evidenciam os tempos de processamento do algoritmo para geração de rotas de 24 pontos de parada nos cenários de São Paulo e Belém, respectivamente.

Os tempos de processamento do algoritmo proposto considerando rotas de 24 pontos de parada apresentados na Figura 6.12(a) e na Figura 6.12(b) seguiram tendências semelhantes aos resultados demonstrados anteriormente, na geração de rotas de 12 e 18 pontos de parada. Nos cenários de São Paulo e Belém com 24 pontos de parada, o emprego



(a) São Paulo

(b) Belém

Figura 6.12: Tempo de processamento 24 pontos de coleta

da política PMT e da estratégia de busca SPFA no algoritmo proporcionou os maiores tempos de processamento, que seriam impraticáveis em uma aplicação real. Isso ocorreu porque a estratégia SPFA finaliza a busca depois que a fila de prioridades está vazia e as possibilidades de atualização de caminhos se esgotaram. Portanto, não há um critério que permita parar o algoritmo quando o vértice final é encontrado, pois isso permitiria retornar soluções não ótimas. Além disso, a combinação Dijkstra bidirecional, VP e PMD proporcionaram os melhores desempenhos.

6.7 Resumo Conclusivo

A personalização das rotas de coleta proposta pelo algoritmo permite encontrar caminhos que minimizam a potência aplicada para empurrar o veículo ou diminuem a distância e tempo de trajeto. Nesse sentido, catadores-carrinheiros que desejam fazer rotas menos cansativas podem escolher a política PMT, que tem a proposta de minimizar a potência aplicada no veículo e, conseqüentemente, o esforço físico. Os resultados demonstraram que essa política, além de minimizar a potência, também encontra caminhos que exigem caminhadas não muito longas, em comparação com a política PMI. A PMI permite buscar caminhos que evitam aclives e declives, além de minimizarem a potência aplicada no veículo. Além disso, essa política apresentou tempo de processamento uma ordem de magnitude inferior em comparação com a PMT. Para os catadores-carrinheiros que preferem rotas rápidas, a política PMD possibilita encontrar caminhos com as menores distâncias e o mais rápido tempo de processamento, em comparação com as outras políticas. Considerando as estratégias de busca implementadas, o melhor desempenho foi alcançado utilizando a Dijkstra bidirecional e a heurística VP.

Capítulo 7

Conclusão

Este trabalho apresentou uma abordagem original de sugestões de rotas para catadores-carrinheiros que utilizam veículos de propulsão humana na coleta seletiva, desenvolvida a partir das tecnologias livres Networkx, OSMnx e *Open Street Map*. O algoritmo proposto gera rotas utilizando as heurísticas Vizinho mais Próximo, Inserção do mais Próximo e Inserção do mais Distante para ordenar os pontos de coleta, estratégias de busca *Shortest Path Faster Algorithm*, Dijkstra Bidirecional e A-star para determinar o melhor caminho entre pontos de coleta. Além disso, as políticas PMT, PMI e PMD foram propostas para personalizar as rotas de acordo com os objetivos de minimização dos catadores-carrinheiros.

A política PMT obteve resultados satisfatórios quanto à diminuição do esforço físico nas simulações porque, além de minimizar a potência aplicada para empurrar o veículo, proporcionou rotas com menor distância e tempo de trajeto do que a política PMI. Além disso, as rotas mais curtas e rápidas foram obtidas a partir da política PMD. No entanto, estas rotas apresentaram os maiores valores de potência aplicada. Demonstrando, assim, que os caminhos de menor distância não foram os que diminuíram o esforço físico dos catadores-carrinheiro nos cenários considerados.

Para avaliar o desempenho das heurísticas, políticas e estratégias de busca, verificou-se o tempo de processamento para geração de rotas. A aplicação da combinação da estratégia Dijkstra bidirecional e da heurística Vizinho mais Próximo no algoritmo alcançou o melhor desempenho. Ademais, a política PMT aumentou o tempo de processamento em relação às outras políticas, visto que atualiza os pesos associados aos *links* da Rede Geográfica cada vez que a massa do veículo é modificada nos pontos de parada.

Com base nos resultados obtidos, é possível afirmar que para aplicar o algoritmo em aplicações reais utilizando a PMT, deve-se combinar esta política com a heurística VP e Dijkstra bidirecional, a fim de desempenhar um tempo de processamento viável. No entanto, se outra política for empregada, as heurísticas IP e ID também proporcionam tempos de processamento viáveis utilizando a estratégia Dijkstra bidirecional.

A relevância social deste trabalho é possibilitar melhorar a qualidade de vida dos catadores-carrinheiros por meio de sugestões de rotas que minimizam a potência empregada para empurrar o veículo e, conseqüentemente, o esforço físico desses trabalhadores. Além disso, a atualização da massa do veículo e aplicação da política PMT durante a roteirização de veículos de tração humana é uma inovação científica.

7.1 Limitações do trabalho

O trabalho apresenta limitações em relação às heurísticas aplicadas, abrangência geográfica, dados e metodologias de teste. Essas limitações são descritas a seguir:

- *O Grafo de Ordenação precisa ser completo*: O algoritmo CATADORA cria rotas considerando que todos os pontos de coleta são interconectados geograficamente. Por isso, utiliza-se grafos completos. Porém, se existirem pontos de coleta desconectados, que não podem ser alcançados por meio de um caminho direto entre si, as heurísticas implementadas no algoritmo CATADORA podem não conseguir ordená-los.
- *Obtenção dos dados de altitude por meio de shapefiles*: O algoritmo CATADORA obtém os dados de altitude dos nós por meio de arquivos do tipo *shapefile*. Esses arquivos são extraídos do banco de dados geomorfométricos do Brasil (Topodata). No entanto, restringem-se ao território brasileiro. Sendo assim, não é possível considerar os dados de altitude para criar rotas de coleta em regiões fora do Brasil a partir da implementação atual do algoritmo CATADORA.
- *Os dados de ofertas de materiais nos experimentos não são reais*: Utilizou-se dados pseudo-aleatórios para definir os pontos de coleta e o peso dos materiais em cada ponto.
- *O algoritmo proposto não foi testado*: nenhuma metodologia de teste foi aplicada para, por exemplo, verificar diferentes entradas e suas respectivas saídas nas funções do código a fim de encontrar falhas.

7.2 Trabalhos Futuros

A pesquisa desenvolvida pode ser ampliada em diversos trabalhos futuros, apresentados a seguir:

- *Algoritmo de geração de rotas para múltiplos veículos com restrições de tempo*: Na coleta primária com múltiplos veículos, as rotas são divididas entre os catadores da cooperativa, para que todas as casas de uma área sejam visitadas por pelo menos um coletor. Sendo assim, propõe-se o desenvolvimento de um algoritmo de geração de rotas para múltiplos veículos com restrições de tempo, de modo que a duração dos percursos seja parecida e os veículos possam retornar ao ponto de chegada ao mesmo tempo.
- *Restrições de massa e/ou capacidade do veículo*: Apesar do veículo de tração humana geralmente ser aberto na parte superior, permitindo encaixar muitos materiais amontoados, restrições de capacidade podem ser aplicadas para que os materiais não atrapalhem a visão dos catadores, uma vez que estes trabalhadores geralmente estão posicionados atrás do veículo. Uma alternativa é permitir que os usuários configurem a capacidade volumétrica do veículo. Além disso, é possível estabelecer

restrições de massa para não prejudicar a saúde física devido esforço exagerado dos catadores-carrinheiros.

- *Roteamento dos veículos de tração humana de forma dinâmica:* Muitas vezes a rota sugerida não pode ser rigorosamente cumprida por causa de intempéries, caminhos obstruídos ou outros fatores externos. Nesses casos, é preciso refazer a rota com base nos pontos de coleta que estão agendados, mas ainda não foram visitados.
- *Utilização de dados da velocidade máxima das ruas:* Os dados de velocidade máxima das vias estão disponíveis no *Open Street Map* e podem ser utilizados para criar rotas mais seguras, evitando vias que tem tráfego em alta velocidade. Diminuindo, assim, os riscos de acidentes.
- *Avaliação do protótipo da aplicação real em campo:* O algoritmo CATADORA pode ser aplicado no desenvolvimento e avaliação de protótipo, associado ao aplicativo Cataki ou outro semelhante, de modo que seja possível identificar e exibir as rotas otimizadas por meio de aplicação web ou móvel.
- *Análise de desempenho a partir de experimentos:* A análise de desempenho do algoritmo CATADORA comparando as combinações de heurísticas, estratégias de busca e políticas permitiu identificar a combinação computacionalmente eficiente para otimizar as rotas. No entanto, em uma aplicação real, é necessário decidir se o processamento das rotas será realizado nos dispositivos ou no servidor. Para isso, é preciso fazer experimentos em dispositivos móveis, como *smartphones* ou máquinas com processamento e memória limitados, e em servidores.
- *Aprendizado de máquina para identificar padrões:* Bases históricas de ofertas de materiais recicláveis associadas ao aprendizado de máquina podem ser utilizadas para identificar padrões e prever a disponibilidade de resíduos, bem como a massa e localização dos materiais.

Referências Bibliográficas

- [1] ABRELPE. Panorama dos resíduos sólidos no brasil 2020, 2020. Acessado em: 10 de setembro de 2021.
- [2] Shabir Ahmad, Imran, Faisal Jamil, Naeem Iqbal, and Dohyeun Kim. Optimal route recommendation for waste carrier vehicles for efficient waste collection: A step forward towards sustainable cities. *IEEE Access*, 8:77875–77887, 2020.
- [3] Ravindra K Ahuja, Kurt Mehlhorn, James Orlin, and Robert E Tarjan. Faster algorithms for the shortest path problem. *Journal of the ACM (JACM)*, 37(2):213–223, 1990.
- [4] Asma M. Altabeeb, Abdulqader M. Mohsen, and Abdullatif Ghallab. An improved hybrid firefly algorithm for capacitated vehicle routing problem. *Applied Soft Computing*, 84:105728, 2019.
- [5] J. Andelmin and E. Bartolini. A multi-start local search heuristic for the Green Vehicle Routing Problem based on a multigraph reformulation. *Computers and Operations Research*, 109:43–63, 2019.
- [6] Sem autor. Cataki. <https://www.cataki.org/pt/>, 2020. Acessado em: 2020-04-04.
- [7] Sem autor. Destino sustentável. <https://www.destinosustentavel.org>, 2020. Acessado em: 16 de setembro de 2021.
- [8] Sem autor. Instituto gea. <https://institutogea.org.br>, 2021. Acessado em: 16 de setembro de 2021.
- [9] Patrícia Belfiore and Luiz Paulo Fávero. *Pesquisa Operacional para cursos de Engenharia*, volume 1. Elsevier Brasil, 2013.
- [10] P. F. Benevides. *Aplicação de heurísticas e metaheurísticas para o problema do caixeiro viajante em um problema real de roteirização de veículos*. PhD thesis, Universidade Federal do Paraná, 2011.
- [11] A. M. Benjamin and J. E. Beasley. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers and Operations Research*, 37(12):2270–2280, 2010.

- [12] Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers Environment and Urban Systems*, 65:126–139, 07 2017.
- [13] BRASIL. Política nacional de cooperativismo, lei nº 5.764, 16 de dezembro de 1971. *Diário Oficial da República Federativa do Brasil*, 1971.
- [14] BRASIL. Política nacional de resíduos sólidos, lei nº 12.305, 2 de agosto de 2010. *Diário Oficial da República Federativa do Brasil*, 2010.
- [15] Rafael Antonio Bruno. *Veículo de Tração Humana para Coleta de Resíduos Recicláveis*. Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica, São Paulo, SP, 2006.
- [16] PIMP MY CARROÇA. Relatório de atividades 2020, 2021. Acessado em: 19 de setembro de 2021.
- [17] Casa.com.br. Celulares são doados a catadores do aplicativo cataki em ação social. <https://casa.abril.com.br/news/celulares-sao-doados-a-catadores-do-aplicativo-cataki-em-acao-social/>, 2019. Acessado em: 17 de setembro de 2021.
- [18] Armando Borges de Castilhos Junior, Naiara Francisca Ramos, Clarissa Martins Alves, Fernando Antônio Forcellini, and Odacir Dionísio Graciolli. Catadores de materiais recicláveis: análise das condições de trabalho e infraestrutura operacional no sul, sudeste e nordeste do brasil. *Ciência & saúde coletiva*, 18(11):3115–3124, 2013.
- [19] CEMPRE. Política nacional de resíduos sólidos—a lei na prática. *São Paulo*, 2012.
- [20] Boris V Cherkassky, Andrew V Goldberg, and Tomasz Radzik. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical programming*, 73(2):129–174, 1996.
- [21] Manus Coffey and Adrian Coad. *Collection of municipal solid waste in developing countries*. UN-Habitat, United Nations Human Settlements Programme, 2010.
- [22] Fernando Augusto Menon Costa. Proposta de mecanismo: motor pneumático para veículo de coleta de recicláveis. In *Monografia do Bacharelado em Engenharia Mecânica, Universidade de Brasília*, pages 1–105, July 2016.
- [23] Bruna Cristina, Jaquetto Pereira, and Fernanda Lira Goes. *Catadores de Materiais Recicláveis Um encontro nacional*. IPEA, 2016.
- [24] Gabriel Altafini Neves da Silva, Francisco Assis da Silva, Daniela Tereza Ascencio Russi, Mário Augusto Pazoti, and Robson Augusto Siscoutto. Algoritmos Heurísticos Construtivos Aplicados Ao Problema Do Caixeiro Viajante Para a Definição De Rotas Otimizadas. *Colloquium Exactarum*, 5(2):30–46, 2013.

- [25] George Dantzig. *Linear programming and extensions*. Princeton University Press, 1962.
- [26] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.
- [27] Subhasish Das, S.-H. Lee, Pawan Kumar, Ki-Hyun Kim, Sang Soo Lee, and Satya Sundar Bhattacharya. Solid waste management: Scope and the challenge of sustainability. *Journal of Cleaner Production*, 228:658–678, 2019.
- [28] Swapan Das and Bidyut Kr Bhattacharyya. Optimization of municipal solid waste collection and transportation routes. *Waste Management*, 43:9–18, 2015.
- [29] Organização das Nações Unidas. Transformando nosso mundo: a agenda 2030 para o desenvolvimento sustentável. *BRASIL*, 2015.
- [30] Grupo de Trabalho da Sociedade Civil para a Agenda 2020. Relatório luz 2020, 2021. Acessado em: 19 de setembro de 2021.
- [31] Grupo de Trabalho da Sociedade Civil para Agenda 2030 do Desenvolvimento Sustentável. ODS, 2021. Acessado em: 03 de setembro de 2021.
- [32] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [33] Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [34] Stuart E Dreyfus. An appraisal of some shortest-path algorithms. *Operations research*, 17(3):395–412, 1969.
- [35] Wolfgang Ertel. *Introduction to Artificial Intelligence (Undergraduate Topics in Computer Science)*. Springer, 2017.
- [36] Duan Fanding. A faster algorithm for shortest-path–spf. *Journal of Southwest Jiaotong University*, 2(9):2, 1994.
- [37] Navarro Ferronato, Gianluca Preziosi, Marcelo Antonio Gorritty Portillo, Edith Gabriela Guisbert Lizarazu, and Vincenzo Torretta. Assessment of municipal solid waste selective collection scenarios with geographic information systems in bolivia. *Waste Management*, 102:919–931, 2020.
- [38] Michael L Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

- [39] Radek Fujdiak, Pavel Masek, Petr Mlynek, Jiri Misurec, and Ekaterina Olshannikova. Using genetic algorithm for advanced municipal waste collection in Smart City. *10th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2016*, 2016.
- [40] Dominik Goeke, Roberto Roberti, and Michael Schneider. Exact and heuristic solution of the consistent vehicle-routing problem. *Transportation Science*, 53(4):1023–1042, 2019.
- [41] Andrew V Goldberg, Haim Kaplan, and Renato F Werneck. Reach for a*: Shortest path algorithms with preprocessing. In *The shortest path problem*, pages 93–139. Citeseer, 2006.
- [42] Andrew V. Goldberg, Haim Kaplan, and Renato F. Werneck. Better landmarks within reach. In Camil Demetrescu, editor, *Experimental Algorithms*, pages 38–51, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [43] Gregory Gutin and Abraham P Punnen. *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media, 2007.
- [44] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
- [45] MF Haisman, FR Winsmann, and RF Goldman. Energy cost of pushing loaded handcarts. *Journal of Applied Physiology*, 33(2):181–183, 1972.
- [46] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 7(4):12–18, 2008.
- [47] D Halliday, R Resnick, and J Walker. *Fundamentos de Física, volume 1: Mecânica*. Ed. LTC, Rio de Janeiro, 2016.
- [48] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [49] Martin Holzer, Frank Schulz, Dorothea Wagner, and Thomas Willhalm. Combining speed-up techniques for shortest-path computations. *Journal of Experimental Algorithmics (JEA)*, 10:2–5, 2005.
- [50] Yu Hsuan Huang and Chuan Kang Ting. Genetic algorithm with path relinking for the multi-vehicle selective pickup and delivery problem. *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pages 1818–1825, 2011.
- [51] D Khan and SR Samadder. Allocation of solid waste collection bins and route optimisation using geographical information system: A case study of dhanbad city, india. *Waste Management & Research*, 34(7):666–676, 2016.

- [52] Şafak Kızıltaş, Hacı Mehmet Alakaş, and Tamer Eren. Collection of recyclable wastes within the scope of the zero waste project: heterogeneous multi-vehicle routing case in kirikkale. *Environmental Monitoring and Assessment*, 192(8):1–22, 2020.
- [53] Fernando Iagares Távora. A herança da rio+ 20. *Temas e Agendas para o Desenvolvimento Sustentável*, page 17, 2012.
- [54] Xin Lan Liao and Chuan Kang Ting. An evolutionary approach for the selective pickup and delivery problem. *IEEE Congress on Evolutionary Computation*, 2010.
- [55] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.
- [56] Open Street Map. Open street map. <https://www.openstreetmap.org/copyright>, 2021. Acessado em: 29 de setembro de 2021.
- [57] James A McHugh. *Algorithmic graph theory*, volume 68056. Citeseer, 1990.
- [58] MNCR. Movimento nacional dos catadores de materiais recicláveis. <http://www.mncr.org.br/>, 2019. Acessado em: 2020-05-21.
- [59] Patricia Takaki Neves. *Variações e aplicações do algoritmo de Dijkstra*. Dissertação digital, Universidade Estadual de Campinas, Campinas, SP, 2007.
- [60] T. A. J. Nicholson. Finding the Shortest Route between Two Points in a Network. *The Computer Journal*, 9(3):275–280, 11 1966.
- [61] Yunyun Niu, Zehua Yang, Ping Chen, and Jianhua Xiao. Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost. *Journal of Cleaner Production*, 171:962–971, 2018.
- [62] Piotr Nowakowski, Aleksander Król, and Bogna Mrówczyńska. Supporting mobile weee collection on demand: A method for multi-criteria vehicle routing, loading and cost optimisation. *Waste Management*, 69:377–392, 2017.
- [63] Fernando Portela. *O Catador Eletrônico*, volume 1. Loqüi, 2015.
- [64] Álvaro Nunes Prestes. *Uma análise experimental de abordagens heurísticas aplicadas ao problema do caixeiro viajante*. Dissertação, Universidade Federal do Rio Grande do Norte, 2006.
- [65] Mohammad Saiedur Rahaman, Yi Mei, Margaret Hamilton, and Flora D Salim. Capra: A contour-based accessible path routing algorithm. *Information Sciences*, 385:157–173, 2017.
- [66] Deyvid Fernando Reis, Eduardo Marandola Junior, and Lúcia Helena Batista Grato. O sentido de lugar para os catadores-carrinheiros: habitar entre a casa e a rua na cidade de londrina. *ENTRE-LUGAR*, 3(5):117–136, 2012.

- [67] Julia Robinson. On the hamiltonian game (a traveling salesman problem). Technical report, RAND PROJECT AIR FORCE ARLINGTON VA, 1949.
- [68] S. Russell and P. Norvig. *Inteligência artificial*. CAMPUS - RJ, 2004.
- [69] Matheus F Sanches, Maria Vitória R Oliveira, Oscar J Ciceri, Lucas Z Ladeira, Islene C Garcia, Nelson LS da Fonseca, and Leandro A Villas. Efis-ecological fuel-consumption intelligent system. *7th International Conference on Distributed Computing in Sensor Systems*, 2021.
- [70] René P Schwarzenbach, Thomas Egli, Thomas B Hofstetter, Urs Von Gunten, and Bernhard Wehrli. Global water pollution and human health. *Annual review of environment and resources*, 35:109–136, 2010.
- [71] SEBRAE. Minha empresa sustentável: Cooperativa de reciclagem. *Cuiabá*, 2017.
- [72] Ayesha Shabbir, Muhammad Faisal Nadeem, and Tudor Zamfirescu. The Property of Hamiltonian Connectedness in Toeplitz Graphs. *Complexity*, 2020:2–7, 2020.
- [73] Sandro Pereira Silva. A organização coletiva de catadores de material reciclável no brasil: dilemas e potencialidades sob a ótica da economia solidária. *IPEA*, 2017.
- [74] Sandro Pereira Silva, Fernanda Lira Goes, and Albino Rodrigues Alvarez. Situação social das catadoras e dos catadores de material reciclável e reutilizável: Brasil. *IPEA*, 2013.
- [75] Fernando Lang da Silveira. Potência de tração de um veículo automotor que se movimenta com velocidade constante. *Revista brasileira de ensino de física. São Paulo. Vol. 33, n. 1 (jan./mar. 2011), 1304, 7 p.*, 2011.
- [76] G. Tavares, Z. Zsigraiova, V. Semiao, and M. G. Carvalho. Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. *Waste Management*, 29(3):1176–1185, 2009.
- [77] Márcio de Morisson Valeriano and Dilce de Fátima Rossetti. Topodata: Brazilian full coverage refinement of srtm data. *Applied Geography*, 32(2):300–309, 2012.
- [78] Hoang Lan Vu, Kelvin Tsun Wai Ng, and Damien Bolingbroke. Parameter interrelationships in a dual phase gis-based municipal solid waste collection model. *Waste Management*, 78:258–270, 2018.
- [79] Xinyu Wang, Tsan Ming Choi, Haikuo Liu, and Xiaohang Yue. Novel Ant Colony Optimization Methods for Simplifying Solution Construction in Vehicle Routing Problems. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3132–3141, 2016.
- [80] Christoph Weiler, Benjamin Biesinger, Bin Hu, and Günther R Raidl. Heuristic approaches for the probabilistic traveling salesman problem. In *International Conference on Computer Aided Systems Theory*, pages 342–349. Springer, 2015.

- [81] Zhenghao Xi, Heping Liu, Huaping Liu, and Bin Yang. Multiple object tracking using the shortest path faster association algorithm. *The Scientific World Journal*, 2014, 2014.
- [82] Yanyan Xue, Zongguo Wen, Hans Bressers, and Ning Ai. Can intelligent collection integrate informal sector for urban resource recycling in china? *Journal of cleaner production*, 208:307–315, 2019.
- [83] Wei Zhang, Hao Chen, Chong Jiang, and Lin Zhu. Improvement and experimental evaluation bellman-ford algorithm. In *2013 International Conference on Advanced ICT and Education (ICAICTE-13)*, pages 138–141. Atlantis Press, 2013.