



Trabalho de Conclusão de Curso

**Heurística baseada em agrupamentos hierárquicos
para resolver um problema de roteamento de
veículos dinâmico e estocástico.**

Ruan Heleno Correa da Silva
rhcs@ic.ufal.br

Orientador:
Prof. Dr. Rian Gabriel Santos Pinheiro

Maceió, dezembro de 2024

Ruan Heleno Correa da Silva

**Heurística baseada em agrupamentos hierárquicos
para resolver um problema de roteamento de
veículos dinâmico e estocástico.**

Monografia apresentada como requisito parcial para
obtenção do grau de Bacharel em Engenharia de Com-
putação pelo Instituto de Computação da Universidade
Federal de Alagoas.

Orientador:

Prof. Dr. Rian Gabriel Santos Pinheiro

Maceió, dezembro de 2024

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação pelo Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

Prof. Dr. Rian Gabriel Santos Pinheiro - Orientador
Universidade Federal de Alagoas

Prof. Dr. Bruno Costa e Silva Nogueira - Examinador
Instituto de Computação
Universidade Federal de Alagoas

Me. Bruno José da Silva Barros - Examinador
Instituto de Computação
Universidade Federal Fluminense

Maceió, dezembro de 2024

Catálogo na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

P379t Silva, Ruan Heleno Correa da.
Heurística baseada em agrupamentos hierárquicos para resolver um problema de roteamento de veículos dinâmico e estocástico / Ruan Heleno Correa da Silva. – 2024.
42 f. : il.

Orientador: Rian Gabriel Santos Pinheiro.
Monografia (Trabalho de conclusão de curso em Engenharia de Computação) - Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2024.

Bibliografia: f. 38-42.

1. Roteamento de veículos. 2. *Last Mile*. 3. Logística. 4. Otimização combinatória. 5. Inteligência artificial. 6. Aprendizado do computador. 7. Clusters. 8. SDCVRP. 9. ICVRP. I. Título.

CDU: 004.81:159.953.5

Agradecimentos

Gostaria de expressar minha profunda gratidão aos meus pais, cujo esforço, apoio e amor foram fundamentais durante minha jornada na graduação. À minha mãe, Sandra, que sempre dizia que nunca devemos parar de aprender. Tenho a certeza de que, no céu, ela cuida de mim e se orgulha de ver seu menino concluindo mais um ciclo. Ao meu pai, Heleno, que sempre lutou para que tudo desse certo e me ensinou que na vida o mais importante não é o dinheiro, mas os valores que carregamos.

Ao meu amor, Karina Fernanda, pela parceria incondicional, lealdade e por estar sempre ao meu lado nos momentos bons e difíceis. Aos meus irmãos, Allan, Luan e Leon, que também estiveram comigo, oferecendo apoio e incentivo ao longo dessa trajetória.

Aos amigos que compartilharam comigo essa caminhada acadêmica: Derek, João Pedro, John Claudio, Mateus, Luana, Lucas Massa, Hiago, Hugo, Jhonnye e Igor. Cada um de vocês tornou essa jornada mais leve e significativa.

Agradeço também aos professores que compartilharam seus conhecimentos, ensinamentos e experiências que levarei por toda a vida acadêmica e profissional. Em especial, aos professores Prof. Dr. Rian Gabriel Santos Pinheiro e Prof. Dr. Bruno Costa e Silva Nogueira, cujas orientações no projeto de iniciação científica foram determinantes para despertar meu interesse no tema deste trabalho, o Roteamento de Veículos.

Por fim, agradeço à coordenação do curso e ao corpo administrativo do Instituto de Computação da Universidade Federal de Alagoas, cuja dedicação e suporte foram fundamentais para esta etapa da minha formação.

Resumo

Esta monografia aborda o desenvolvimento de algoritmos para resolver um caso específico do Stochastic and Dynamic Capacitated Vehicle Routing Problems (SD-CVRP), denominado Last Mile Incremental Capacitated Vehicle Routing Problem (ICVRP). A proposta envolve a aplicação de uma estratégia fundamentada na organização de modelos de agrupamento em níveis para definição do veículo de cada encomenda analisada. Tal aplicação tem como objetivo a geração de rotas eficientes com intuito de otimizar métricas fundamentais, como a distância a ser percorrida, a quantidade de rotas geradas, o tempo de execução das etapas presentes.

Os resultados experimentais, obtidos a partir de mais de cem mil entregas entre entregadores de onze regiões do Brasil, demonstram a eficácia da abordagem para resolver o ICVRP, tendo demonstrado um desempenho competitivo. Em comparação com a abordagem K-means Greedy, que apresentou a menor distância percorrida total entre os algoritmos de referência, a solução proposta reduziu a distância percorrida em 2.500 km, além de manter um tempo de execução competitivo, com diferença inferior a 2 minutos em relação ao algoritmo QRP Sweep, o mais rápido entre os testados. No entanto, a proposta teve um aumento de quase 52% na distância percorrida em relação ao VRP estático, compensado por uma economia de 75% no tempo de execução. Comparada ao método MAS, da Loggi, a solução mostrou um aumento de 19% na distância total percorrida. Esses resultados indicam a viabilidade e a eficiência da solução para o ICVRP, ao mesmo tempo em que evidenciam oportunidades para aprimoramentos futuros.

Palavras-chave: Roteamento de veículos; Last Mile; Logística; Otimização; Inteligência artificial; Aprendizagem de máquina; Clusterização; SDCVRP; ICVRP.

Abstract

This monograph presents the development of algorithms to solve a specific case of the Stochastic and Dynamic Capacitated Vehicle Routing Problems (SD-CVRP), known as the Last Mile Incremental Capacitated Vehicle Routing Problem (ICVRP). The proposed approach involves applying a strategy based on hierarchical clustering models to determine vehicle assignments for each analyzed delivery. This application aims to generate efficient routes to optimize key metrics such as distance traveled, number of routes generated, and execution time of the various process stages.

Experimental results, obtained from over one hundred thousand deliveries across eleven regions in Brazil, demonstrate the effectiveness of the proposed approach for solving the ICVRP and show competitive performance. Compared to the K-means Greedy approach, which had the lowest total distance traveled among the reference algorithms, the proposed solution reduced the total distance by 2,500 km and maintained a competitive execution time, with a difference of less than 2 minutes compared to the QRP Sweep algorithm, the fastest among those tested. However, the proposed method showed an increase of nearly 52% in the distance traveled compared to the static VRP, offset by a 75% reduction in execution time. When compared to the MAS method from Loggi, the solution showed an increase of 19% in the total distance traveled. These results indicate the feasibility and efficiency of the solution for the ICVRP while highlighting potential opportunities for future improvements.

Key-words: Vehicle routing; Last Mile; Logistics; Optimization; Artificial Intelligence; Machine learning; Clustering; SDCVRP; ICVRP.

Lista de Figuras

2.1	Ciclo de uma entrega. First Mile (à esquerda), Middle Mile (no centro) e Last-Mile (à direita).	6
2.2	Exemplo de como um algoritmo que resolve o TSP funciona. À esquerda estão os locais a serem visitados, enquanto à direita está o caminho para visitar todos os locais.	7
2.3	Exemplo de como um algoritmo que resolve o VRP funciona. À esquerda estão os locais a serem visitados, enquanto à direita estão rotas para que os veículos visitem partindo do centro de expedição (EC).	9
2.4	Exemplo de clusterização com K-means. À esquerda os pontos distribuídos, à direita, os pontos organizados em 3 clusters, com a aplicação do K-means. . . .	11
3.1	Exemplo de 8 UCs e 4 conjuntos de rotas anteriores associadas.	15
4.1	Fluxo geral para solução do Last-Mile ICVRP com HCM. À esquerda, mostra-se a organização da fase offline, e à direita, a fase online.	19
4.2	Exemplo de coordenadas dos dados históricos.	21
4.3	Exemplo da aplicação do K-means para agrupar os dados históricos.	21
4.4	Exemplo de modelos de cluster de segundo nível considerando $k = 3$ e $u = 5$. . .	22
4.5	Exemplo de modelo HCM com $k=3$ e $u=5$	23
4.6	Exemplo da organização das rotas para resolver o TSP, desconsiderando a volta ao EC.	24
5.1	Dados históricos presentes em pa-0.	26
5.2	Dados históricos presentes em pa-1.	26
5.3	Dados históricos presentes em df-0.	27
5.4	Dados históricos presentes em df-1.	27
5.5	Dados históricos presentes em df-2.	27
5.6	Dados históricos presentes em rj-0.	28
5.7	Dados históricos presentes em rj-1.	28
5.8	Dados históricos presentes em rj-2.	29
5.9	Dados históricos presentes em rj-3.	29
5.10	Dados históricos presentes em rj-4.	29
5.11	Dados históricos presentes em rj-5.	30
5.12	Diferença entre a distância total e o tempo de execução entre o método proposto e o CVRP.	35

Lista de Tabelas

5.1	Resultados experimentais em pa-0	32
5.2	Resultados experimentais em pa-1	32
5.3	Resultados experimentais em df-0	32
5.4	Resultados experimentais em df-1	32
5.5	Resultados experimentais em df-2	33
5.6	Resultados experimentais em rj-0	33
5.7	Resultados experimentais em rj-1	33
5.8	Resultados experimentais em rj-2	33
5.9	Resultados experimentais em rj-3	34
5.10	Resultados experimentais em rj-4	34
5.11	Resultados experimentais em rj-5	34
5.12	Resultados experimentais	34

Conteúdo

Lista de Figuras	v
Lista de Tabelas	vii
Lista de Abreviaturas e Siglas	x
1 Introdução	1
1.1 Objetivos	2
1.2 Justificativa	3
1.3 Motivação	3
1.4 Estrutura do Trabalho	3
2 Fundamentação Teórica	5
2.1 Logística	5
2.1.1 Etapas de uma entrega	5
2.2 Otimização combinatória	6
2.2.1 Problema do Caixeiro Viajante - TSP	7
2.2.2 Problema do Roteamento de Veículos - VRP	8
2.2.3 Problema de Roteamento de Veículos Dinâmico (DVRP)	9
2.2.4 Problema do Roteamento de Veículos Estocástico - SVRP	10
2.3 A inteligência artificial e o aprendizado de máquina	10
2.3.1 Método K-means	11
2.4 OpenStreetMap	11
2.4.1 Servidor Open Source Routing Machine - OSRM	12
3 Revisão da Literatura	13
3.1 Roteamento de veículos capacitados dinâmicos e estocásticos	13
3.1.1 Abordagens	14
3.2 Last-Mile Incremental Capacitated Vehicle Routing Problem	15
3.2.1 K-means Greedy e QRP Sweep	16
3.2.2 Greedy TSP e Multiple Scenario Approach	16
3.2.3 Multi-Agent System	16
4 Metodologia	18
4.1 Fluxo de trabalho	18
4.2 Fase de planejamento (offline)	19
4.2.1 Carregando conjunto de dados de treinamento	20
4.2.2 Construção do primeiro nível HCM	20
4.2.3 Alocação das UCs	21

4.2.4	Construção do segundo nível HCM	22
4.2.5	Pré-teste no HCM	22
4.3	Fase de Execução (online)	23
4.3.1	Alocação das entregas	24
4.3.2	Organização das rotas	24
5	Resultados e Discussões	25
5.1	Instâncias de benchmark	25
5.2	Algoritmos de benchmark	30
5.3	Configuração de parâmetros	30
5.4	Protocolo Experimental	31
5.5	Comparação com outros modelos	31
5.5.1	Resultados totais	34
6	Conclusão	36
6.1	Trabalhos Futuros	37
	Referências bibliográficas	38

Lista de Abreviaturas e Siglas

CVRP	Capacitated Vehicle Routing Problem
DC	Distribution Center
DVRP	Dynamic Vehicle Routing Problem
EC	Expedition Center
ICVRP	Incremental Capacitated Vehicle Routing Problem
HCM	Hierarchical Cluster Model
OVRP	Open Vehicle Routing Problem
SD-CVRP	Stochastic and Dynamic Capacitated Vehicle Routing
SVRP	Stochastic Vehicle Routing Problem
TSP	Travelling Salesman Problem
UC	Unidade de Carregamento
UL	Unit Load
VRP	Vehicle Routing Problem

1

Introdução

A Pandemia do novo coronavírus trouxe uma nova realidade para a economia brasileira, impactando significativamente o setor de e-commerce, que registrou um crescimento de 40% em 2020, alcançando um faturamento de R\$ 87,4 bilhões [1], e atingindo R\$ 185,7 bilhões em 2023 [2]. Isso resultou na necessidade de acelerar o fluxo de entregas para atender toda essa demanda, ocasionando desafios substanciais para as empresas de logística, impactando também a experiência do consumidor e a qualidade de vida dos cidadãos em diversas regiões, além de afetar a mobilidade urbana com aumento no congestionamento do tráfego. O meio ambiente também é impactado pelo prolongamento dos tempos de viagem e pelo aumento do consumo de combustível, ocasionando maiores emissões de gases poluentes.

Os problemas de roteamento de veículos (VRP) buscam encontrar a melhor forma de organizar rotas para uma frota de veículos a fim de atender um conjunto de demandas. Eles são conhecidos por sua complexidade computacional, sendo classificados como NP-difíceis na literatura [3]. A crescente demanda por soluções de alocação de entregas para veículos com capacidade limitada, em um ambiente dinâmico em que surgem novas requisições de forma contínua, exige o desenvolvimento de abordagens inovadoras e eficientes. Esse desafio é particularmente relevante para empresas como Loggi, Rappi e iFood, que operam com modelos baseados na economia compartilhada, nos quais motoristas independentes utilizam suas próprias plataformas e veículos para realizar as entregas [4].

Com o aumento constante do volume de dados disponíveis e o avanço das tecnologias de comunicação, a quantidade de informações pertinentes para abordar problemas de roteamento de veículos aumentou. O uso generalizado de serviços de posicionamento e comunicação móvel permite a coleta de dados em tempo real. Como resultado, a comunidade tem se concentrado cada vez mais no estudo e desenvolvimento de algoritmos de otimização eficientes capazes de lidar com desafios complexos, como é o caso dos *Problemas de Roteamento de Veículos Capacitados Dinâmicos e Estocásticos* (SD-CVRP), que se caracteriza pela revelação dinâmica das

informações durante a execução do roteamento, sem conhecimento prévio sobre as futuras demandas, tornando o problema estocástico. Esses avanços oferecem a oportunidade de melhorar a qualidade das soluções por meio de novas técnicas e modelos melhores adaptados às condições reais. Consequentemente, surge uma necessidade premente de algoritmos de otimização eficientes aptos a enfrentar problemas desafiadores.

Um problema específico do SD-CVRP enfrentado por empresas como a Loggi é o *Last-Mile Incremental Capacitated Vehicle Routing Problem* (ICVRP), que tem como objetivo alocar cada entrega de um conjunto de novas entregas em uma unidade de carregamento (UC). As demandas são alocadas individualmente e na ordem de chegada predefinida, tendo como entrada um conjunto de clientes previamente atendidos (histórico) $H = \{p_1, \dots, p_l\}$, um conjunto \mathcal{U} de unidades de carregamento e um conjunto com n novos clientes $D = \{v_{\ell+1}, \dots, v_{\ell+n}\}$. Importante destacar que cada alocação é realizada de acordo com informações extraídas do histórico de clientes atendidos. Tais informações são obtidas do conjunto H .

De maneira geral, a resolução dos problemas relacionados ao SD-CVRP pode ser dividida em duas etapas. A primeira etapa, chamada de planejamento, ocorre de maneira *offline* e utiliza dados históricos de entrega para computar um conjunto de decisões pré-definidas. A segunda etapa, chamada de execução, ocorre de maneira *online* e utiliza as ações pré-definidas na etapa anterior assim como as informações sobre os eventos em tempo real (por exemplo, a chegada de uma nova demanda) para gerar as rotas finais.

Esta pesquisa foca em resolver o ICVRP, um tipo específico de SD-CVRP que envolve a alocação dinâmica de novas entregas a unidades de carregamento com capacidade limitada. Propomos um método baseado em agrupamento hierárquico que explora dados históricos para antecipar demandas futuras e otimizar a alocação de rotas. Nossa abordagem é inspirada por pesquisas anteriores sobre SD-CVRP, otimização antecipatória e técnicas de clustering, além de considerar os desafios específicos do ICVRP no contexto de operações de entrega em grande escala em áreas urbanas.

1.1 Objetivos

Podemos dividir os objetivos deste trabalho em:

Objetivo geral: O objetivo desta pesquisa é desenvolver uma metodologia eficiente para resolver o Last-Mile Incremental Capacitated Vehicle Routing Problem (ICVRP). Ao abordar este problema, espera-se aprimorar a eficiência logística, impactando positivamente a satisfação dos clientes, a mobilidade urbana e a sustentabilidade. Os resultados visam reduzir congestionamentos e emissões de poluentes, contribuindo para uma mobilidade urbana mais eficiente e redução do impacto ambiental.

Objetivos específicos: Apresentar um método para a alocação dinâmica de entregas em unidades de carregamento com capacidade limitada, utilizando uma abordagem baseada em agru-

pamento hierárquico contendo modelos de agrupamento organizados em dois níveis. O foco do trabalho está na otimização da alocação de rotas em operações de grande escala, especialmente em cenários urbanos, nos quais os desafios de logística são mais acentuados.

1.2 Justificativa

Este trabalho é justificado pela rápida expansão do setor de e-commerce e a crescente necessidade de soluções logísticas que atendam à alta demanda de maneira eficiente, reduzam congestionamentos e minimizem o impacto ambiental. A aplicação de uma abordagem eficiente para o ICVRP é relevante para empresas de logística urbana, principalmente as que fazem uso do modelo de economia compartilhada, em que a melhoria nos processos logísticos traz benefícios para a eficiência operacional e a experiência do consumidor.

1.3 Motivação

A principal motivação para este trabalho foi o **Programa de Bolsas de Pesquisa da Loggi (PBP-Loggi)**, uma iniciativa de 11 meses que promoveu o desenvolvimento de novas soluções na área de otimização em logística, por meio de projetos de iniciação científica realizados em parceria com universidades brasileiras. Durante o programa, foi explorado o repositório *Loggi Benchmark for Urban Deliveries (Loggibud)* [5], com enfoque no estudo do *Last Mile Incremental Capacitated Vehicle Routing Problem (ICVRP)*, um dos problemas logísticos enfrentados pela Loggi.

Além disso, este projeto integrou o **Programa Institucional de Bolsas de Iniciação Científica (PIBIC)** no ciclo 2022/2023, tendo sido reconhecido com o **Prêmio de Excelência Acadêmica** pelo trabalho intitulado “*Algoritmos para o Problema de Roteamento de Veículos Online e Estocástico*”. Este reconhecimento ressalta a relevância do estudo para a pesquisa acadêmica e sua aplicação prática na área de logística.

1.4 Estrutura do Trabalho

Este trabalho está organizado da seguinte forma:

- **Capítulo 1:** Introdução - apresenta o contexto, os objetivos e a relevância do problema de roteamento dinâmico, destacando o impacto do e-commerce e os desafios enfrentados pelas empresas de logística.
- **Capítulo 2:** Fundamentação Teórica - apresenta uma análise teórica dos fundamentos aplicáveis à pesquisa.

- **Capítulo 3:** Revisão da Literatura - explora trabalhos e algoritmos relevantes na área de roteamento de veículos, com foco em SD-CVRP e ICVRP, e discute as técnicas de otimização aplicadas.
- **Capítulo 4:** Metodologia - descreve o modelo de clusterização hierárquica proposto, chamado HCM.
- **Capítulo 5:** Resultados e Discussões - apresenta os resultados obtidos com a implementação do método, comparando a eficiência da abordagem com outras técnicas e discutindo as implicações.
- **Capítulo 6:** Conclusão - resume os principais achados e contribuições da pesquisa, além de sugestões para trabalhos futuros.

2

Fundamentação Teórica

ESTE capítulo aborda os principais fundamentos utilizados na elaboração deste trabalho, desde os conceitos de logística e roteamento de veículos que envolvem os principais problemas presentes na literatura até detalhes sobre as ferramentas utilizadas que desempenham papéis essenciais no desenvolvimento da proposta, fornecendo recursos e funcionalidades necessárias para a implementação da abordagem baseada em clusterização hierárquica para resolver o ICVRP.

2.1 Logística

Logística é a área que gerencia o planejamento, a implementação e o controle do fluxo de serviços, mercadorias e informações, em todo o seu trajeto até seu destino final, para atender às necessidades dos clientes. Seu principal objetivo é garantir que o objeto de envio chegue ao destino de maneira ágil, eficiente e econômica, do ponto de origem ao de destino.

2.1.1 Etapas de uma entrega

Nos serviços de logística, a entrega de encomendas é um processo crítico que envolve várias etapas. Desde o momento em que uma encomenda é recebida até seu destino final, é fundamental garantir que ela seja entregue de forma eficaz e eficiente. Para alcançar esse objetivo, é necessário rastrear a encomenda ao longo do seu percurso, que pode ser dividido em três etapas principais: First-Mile, Middle-Mile e Last-Mile [6, 7].

A Figura 2.1 ilustra o processo realizado em uma entrega. Desde o veículo de coleta, passando pelo centro de distribuição (DC) e pelo centro de expedição (EC), até ser direcionado para os entregadores realizarem o percurso final de destino da entrega.

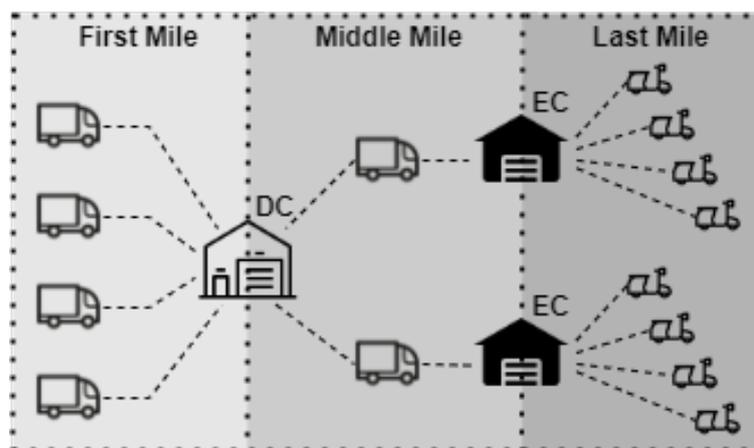


Figura 2.1: Ciclo de uma entrega. First Mile (à esquerda), Middle Mile (no centro) e Last-Mile (à direita).

First-Mile (Primeira milha). A etapa inicial refere-se ao processo de coleta, organização e separação da encomenda, realizado pelo fornecedor no local de origem. Esta etapa encerra-se quando o produto é transportado até um ponto de consolidação de mercadorias, também chamado de Centro de Distribuição (DC).

Middle-Mile (Milha média). A segunda etapa trata da realização do transporte do produto até um Centro de Expedição (EC) que seja o mais próximo do destino final, o cliente.

Last-Mile (Última milha). A última etapa refere-se ao final do processo de logística, quando o cliente recebe seu produto, ou seja, é o momento em que a encomenda sai do EC e vai até o cliente. É a etapa que focaremos no problema proposto.

2.2 Otimização combinatória

Ramo da matemática aplicada e da ciência da computação relacionada à pesquisa operacional que tem como principal objetivo obter a melhor solução entre um conjunto finito de possibilidades, considerando restrições específicas. Abaixo definimos as principais classes de problemas presentes na otimização combinatória, sendo elas: P, NP, NP-difícil e NP-completo [8, 9].

Classe P: Essa classe refere-se a problemas de decisão que podem ser resolvidos em tempo polinomial.

Classe NP: Refere-se a problemas de decisão cuja resposta “Sim” pode ser verificada em tempo polinomial.

Classe NP-difícil: Um problema Π é NP-difícil se todo problema em NP pode ser reduzido em tempo polinomial a Π .

Classe NP-completo: Um problema é NP-completo se estiver em NP e for NP-difícil.

Muitos problemas abordados pela otimização combinatória são NP-difíceis, caracterizados pela elevada complexidade de encontrar a solução ótima, já que não é conhecido algoritmo polinomial para resolvê-los. Embora a verificabilidade de um certificado em tempo polinomial

seja uma propriedade de problemas em NP, os problemas NP-difíceis exigem a exploração de um grande número de possibilidades, tornando a busca pela solução ótima extremamente desafiadora [10].

2.2.1 Problema do Caixeiro Viajante - TSP

Sendo um dos problemas mais conhecidos de análise combinatória e amplamente utilizado na área de logística, o problema do caixeiro viajante envolve encontrar o caminho mais curto que visita um conjunto de cidades exatamente uma vez e retorna à cidade de origem [11, 12]. Podendo ser definido como segue:

Seja $G = (V, E)$ um grafo completo, onde:

- $V = \{v_1, v_2, \dots, v_n\}$ é o conjunto de n vértices (cidades a serem visitadas).
- E é o conjunto de arestas que conecta os vértices ($E = \{(v_i, v_j) \mid i \neq j, i, j \in V\}$).
- $c_{ij} \geq 0$ é o custo (ou distância) associado à aresta $(v_i, v_j) \in E$, representando o custo de viajar diretamente de v_i para v_j .

O objetivo é encontrar uma permutação $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ de V que minimize o custo total do ciclo Hamiltoniano, garantindo como retorno uma sequência de vértices conectados, onde cada vértice é visitado exatamente uma vez, tendo o retorno ao vértice inicial, tal que:

$$\min_{\pi} \sum_{i=1}^n c_{\pi(i), \pi(i+1)},$$

onde $\pi(n+1) = \pi(1)$, indicando que o ciclo retorna ao ponto de partida. A Figura 2.2 ilustra como a solução ao Problema do Caixeiro Viajante pode ser aplicada.

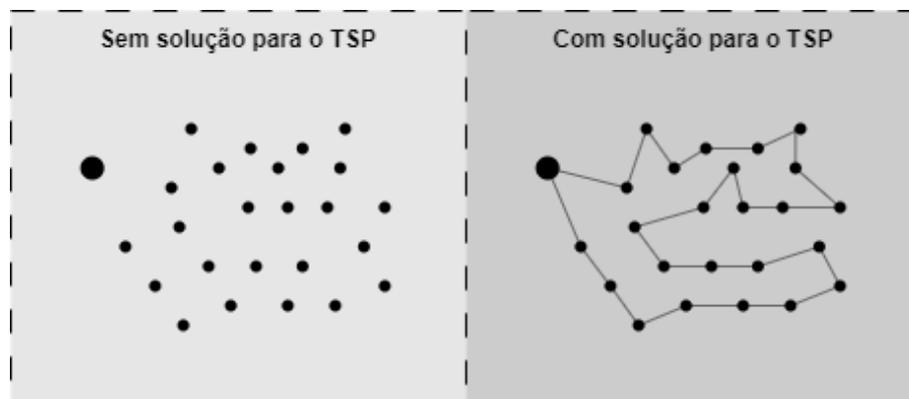


Figura 2.2: Exemplo de como um algoritmo que resolve o TSP funciona. À esquerda estão os locais a serem visitados, enquanto à direita está o caminho para visitar todos os locais.

O TSP é um problema NP-difícil, o que significa que não há um algoritmo eficiente conhecido para resolvê-lo em tempo polinomial à medida que o número de cidades cresce. Isso implica que, conforme o número de cidades aumenta, a quantidade de possíveis soluções aumenta de forma exponencial, tornando impraticável verificar todas as rotas possíveis para encontrar a solução ótima.

2.2.2 Problema do Roteamento de Veículos - VRP

Um dos principais desafios presente em indústrias e setores, como logística, transporte, distribuição e entrega, é a redução da distância a ser percorrida pelo condutor para realização de seu objetivo, seja ele a distribuição de mercadorias, a entrega de pacotes ou o transporte de passageiros. Esse desafio denominamos como *Problema de Roteamento de Veículos* (VRP), podendo ser definido como segue:

Seja $G = (V, E)$ um grafo completo, onde:

- $V = \{v_0, v_1, \dots, v_n\}$ é o conjunto de vértices, com v_0 representando o depósito e v_1, \dots, v_n os clientes.
- c_{ij} é o custo de viajar da cidade v_i para v_j , $i, j \in V$.

O objetivo do Problema de Roteamento de Veículos (VRP) é determinar as rotas para K veículos, de modo que:

- Cada cliente seja visitado exatamente uma vez.
- O custo total das rotas seja minimizado.

A sua função objetivo busca minimizar o custo total das rotas percorridas pelos veículos. Esse custo pode representar diferentes métricas, como a distância total percorrida, o tempo gasto ou o consumo de combustível, dependendo do contexto do problema. A principal restrição é que cada cliente deverá ser visitado exatamente uma vez.

Assim como o problema do caixeiro viajante, o problema de roteamento de veículos é um problema clássico de otimização combinatória. Ele utiliza diversas variáveis como a localização dos clientes e dos veículos, as demandas dos clientes, as capacidades dos veículos, entre outras, a fim de definir as rotas mais eficientes para uma frota de veículos, que devem atender a um conjunto de clientes com demandas específicas [13].

A Figura 2.3 mostra um exemplo de como as demandas estão organizadas inicialmente e o resultado final com as rotas definidas para os veículos e suas demandas.

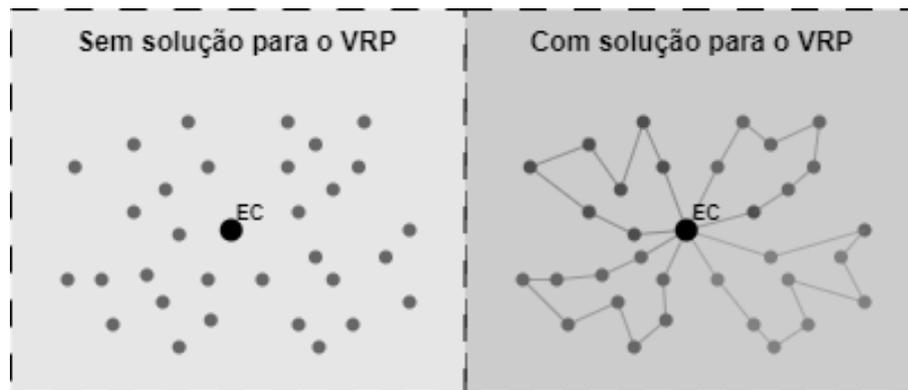


Figura 2.3: Exemplo de como um algoritmo que resolve o VRP funciona. À esquerda estão os locais a serem visitados, enquanto à direita estão rotas para que os veículos visitem partindo do centro de expedição (EC).

Open VRP - OVRP: Empresas de logística, como Loggi, Rappi e iFood, operam com modelos baseados na economia compartilhada, onde condutores independentes utilizam suas próprias plataformas e veículos para realizar as entregas [4]. Por esse motivo, as rotas criadas não precisam voltar à origem. Isso é conhecido como Open VRP (OVRP) [14].

Capacitated VRP - CVRP: O CVRP é uma extensão do VRP onde cada veículo tem uma capacidade máxima Q , e cada cliente $v_i \in V$ possui uma demanda q_i . Além das restrições do VRP, o CVRP inclui a restrição que assegura que a demanda total atendida por cada veículo k não exceda sua capacidade Q .

2.2.3 Problema de Roteamento de Veículos Dinâmico (DVRP)

O Problema de Roteamento de Veículos Dinâmico (DVRP), também conhecido como *online* ou em tempo real, é uma extensão do VRP que incorpora a necessidade de tomada de decisão em um ambiente dinâmico. Neste contexto, as informações relacionadas aos clientes, como suas demandas ou localizações, tornam-se conhecidas apenas no momento em que os clientes são analisados, exigindo adaptações contínuas durante a execução das rotas [15].

As características do DVRP incluem:

- Atualizações nas demandas q_i dos clientes.
- Alterações nas localizações $v_i \in V$ dos clientes.
- Variações nos custos c_{ij} das arestas $(i, j) \in E$.

A função objetivo no DVRP busca minimizar o custo total das rotas planejadas, considerando o cenário dinâmico. O objetivo principal é garantir a eficiência do roteamento em tempo real, adaptando as decisões às informações atualizadas enquanto atende às demandas dos clientes e respeita as restrições do problema.

2.2.4 Problema do Roteamento de Veículos Estocástico - SVRP

O Problema de Roteamento de Veículos Estocástico (SVRP) é uma extensão do VRP clássico em que algumas informações do problema, como demandas q_i , localizações v_i ou custos c_{ij} , são incertas e modeladas como variáveis aleatórias, tendo conhecimento apenas dos dados anteriores (histórico) que podem ser utilizados para cálculos estatísticos [16, 17, 18]. Essas incertezas são reveladas gradualmente durante a execução das rotas. As restrições são as mesmas do VRP, mas as decisões precisam ser robustas para lidar com as incertezas, garantindo soluções viáveis e eficientes para alcançar o objetivo de minimizar o custo total das rotas geradas.

2.3 A inteligência artificial e o aprendizado de máquina

A inteligência artificial é um ramo da computação que tem como objetivo a análise e o desenvolvimento de algoritmos que consigam reproduzir o comportamento humano na realização de tarefas, na tomada de decisões e no pensamento criativo que possam resolver problemas complexos de forma eficaz e autônoma [19]. Dentre as áreas presentes na inteligência artificial destaca-se o aprendizado de máquina.

Aprendizado de Máquina. Tem como objetivo automatizar o processo de modelos capazes de extrair padrões de informações com os dados sem terem sido programados para isso. Neste sentido, o fato do algoritmos aprender com os dados tem impulsionado avanços significativos em diversas áreas, incluindo logística. [20, 21]

Há três tipos de aprendizagem de máquina [22, 21]:

- **Aprendizado supervisionado:** neste tipo de aprendizado, os dados são rotulados, ou seja, possuem respostas associadas. O modelo utiliza essas informações para aprender a mapear entradas para saídas, permitindo prever valores ou categorias de novos dados. Esse tipo de aprendizado inclui modelos de classificação (para prever categorias) e de regressão (para prever valores numéricos).
- **Aprendizado não supervisionado:** neste tipo de aprendizado, os dados não possuem rótulos ou classificações prévias. O objetivo é identificar padrões ou estruturas subjacentes nos dados, como agrupamentos (clusterização) ou relações entre variáveis (modelos de associação). Modelos de aprendizado não supervisionado são amplamente utilizados para organizar dados em grupos ou identificar características relevantes de forma automática.
- **Aprendizado por reforço:** neste tipo de aprendizado, o sistema interage com um ambiente, tomando ações que resultam em recompensas ou penalidades. Com base nesse feedback, o sistema ajusta sua política de tomada de decisão para maximizar as recompensas ao longo do tempo. Esse processo envolve tentativa e erro, permitindo que o modelo aprenda a tomar as melhores decisões para cada estado observado no ambiente.

2.3.1 Método K-means

É uma técnica de clusterização utilizado como algoritmo de aprendizagem de máquina não supervisionado, que busca organizar os dados em uma quantidade definida de grupos (parâmetro K), a partir de suas características. Os dados são agrupados através da menor distância para um valor central presente em cada grupo. Esses valores centrais, denominados centroides, são definidos inicialmente de forma aleatória, mas a cada análise, cada centroide escolhido é reposicionado levando em consideração a média das posições dos seus dados presentes [23, 24, 25].

Sua capacidade de agrupar dados com base em similaridade faz com que o K-means seja considerado uma aplicação interessante ao lidar com dados de localização. Pode-se, a partir dele, explorar padrões espaciais ocultos em conjuntos de dados geográficos, revelando insights valiosos sobre a distribuição e concentração de eventos, pessoas ou lugares.

A Figura 2.4 mostra um exemplo de como pontos (x, y) podem ser organizados em clusters com o algoritmo K-means.

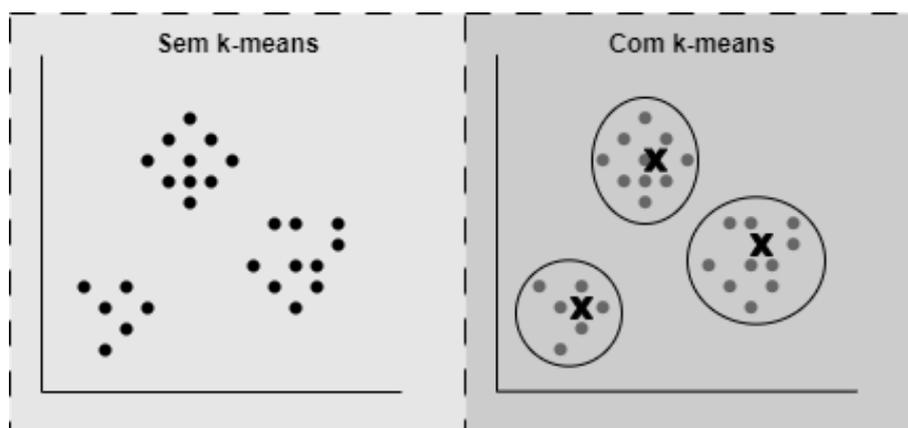


Figura 2.4: Exemplo de clusterização com K-means. À esquerda os pontos distribuídos, à direita, os pontos organizados em 3 clusters, com a aplicação do K-means.

2.4 OpenStreetMap

O OpenStreetMap [26] é um projeto colaborativo de mapeamento online que fornece dados geoespaciais gratuitos e abertos para todo o mundo. Seus dados são armazenados em um formato de banco de dados geoespacial e podem ser acessados através de APIs, eles incluem informações detalhadas sobre estradas, edifícios, pontos de interesse e outras características geográficas, tornando-o uma fonte valiosa para aplicações de roteamento, geolocalização e análise espacial.

Ele desempenha um papel crucial na democratização do acesso às informações geográficas, sendo utilizado em aplicações de diversas áreas, como é o caso do aplicativo de navegação GPS. Seu funcionamento ocorre através da colaboração dos usuários, que têm liberdade na criação, edição e correção de informações geográficas.

2.4.1 Servidor Open Source Routing Machine - OSRM

O OSRM (Open Source Routing Machine) [27] é um servidor de roteamento de código aberto que utiliza dados do OpenStreetMap para calcular rotas otimizadas aplicando o algoritmo de Dijkstra. Ele busca analisar dois ou mais pontos de localização (latitude e longitude) e, a partir disso, fornecer informações sobre distâncias, tempos de viagem e direções para ajudar na otimização de rotas em aplicações de logística, transporte e planejamento urbano.

A implantação de um servidor OSRM local pode ser muito benéfica em casos em que o acesso à rede é restrito, e é preferível manter a conexão dentro da rede local. Outras ferramentas de roteamento também podem ser muito caras ou podem não fornecer dados precisos para sua região.

Após instalar o OSRM, você pode acessar seus serviços de roteamento fazendo solicitações ao servidor por meio da API REST. O OSRM suporta vários perfis de roteamento, como dirigir, andar de bicicleta e caminhar, que você pode escolher com base no seu caso de uso.

3

Revisão da Literatura

ESTE capítulo busca realizar uma revisão da literatura relacionada ao problema proposto, Last Mile ICVRP. O capítulo buscará definir o problema proposto para solução e também a classe que ele faz parte, o SD-CVRP, além de descrever as abordagens presentes na literatura que buscam resolver os problemas desta classe.

3.1 Roteamento de veículos capacitados dinâmicos e estocásticos

Os problemas de roteamento de veículos capacitados dinâmicos e estocásticos (SD-CVRPs) são caracterizados pela necessidade de tomadas de decisões em tempo real, à medida que novas informações se tornam disponíveis, e pela incerteza inerente aos parâmetros operacionais. Esses problemas, denominados *dinâmicos*, lidam com situações em que a informação é revelada progressivamente, exigindo que as rotas sejam continuamente ajustadas [17, 18, 28, 29]. A natureza *estocástica* desses problemas incorpora a incerteza ao modelar parâmetros do problema como variáveis aleatórias, com distribuições de probabilidade conhecidas, resultando em desafios adicionais na construção de rotas e no gerenciamento de restrições de capacidade [30, 31]. Diversos algoritmos e heurísticas têm sido desenvolvidos para lidar com essas complexidades, como abordagens baseadas em reotimização contínua e algoritmos gulosos adaptativos [32, 33, 34, 35, 36].

De maneira geral, a resolução dos problemas relacionados ao SD-CVRP pode ser dividida em duas etapas. A primeira etapa, chamada de *planejamento*, ocorre de maneira *offline* e utiliza dados históricos de entrega para computar um conjunto de decisões pré-definidas. A segunda etapa, chamada de *execução*, ocorre em tempo real e utiliza as ações pré-definidas na etapa anterior assim como as informações sobre os eventos em tempo real (ex.: chegada de uma nova demanda) para gerar as rotas finais.

Historicamente, os SD-CVRPs têm sido abordados utilizando-se processos de Markov e programação estocástica. Processos de Markov são utilizados para modelar a transição entre estados com base em probabilidades condicionais, permitindo a otimização de decisões ao longo de um horizonte de tempo dinâmico [37, 38, 39]. Por outro lado, a programação estocástica busca otimizar uma medida de risco (como o valor esperado, a variância ou o valor em risco condicional) enquanto atende a determinadas restrições operacionais [40, 41, 42]. Essas metodologias, apesar de serem bem estabelecidas, muitas vezes não se adequam completamente às demandas do mundo real, especialmente em ambientes de entrega altamente dinâmicos, como o *e-commerce* e operações logísticas urbanas. Nesses cenários, a localização dos clientes e a demanda por entregas podem mudar rapidamente, exigindo soluções que sejam não apenas otimizadas, mas também adaptáveis e responsivas.

3.1.1 Abordagens

Com o avanço da tecnologia e o aumento da complexidade das operações logísticas, novas abordagens têm sido propostas para tratar dos SD-CVRPs. Duas estratégias se destacam:

Uso de Agentes Autônomos com Mineração de Dados: Uma abordagem recente, explorada por [4], envolve a utilização de agentes autônomos que modelam as rotas de entrega e empregam técnicas de mineração de dados para extrair informações relevantes sobre a distribuição de pacotes. Essa abordagem permite a otimização da alocação de pacotes em tempo real, ajustando as decisões de roteamento com base em padrões históricos e tendências emergentes. Embora promissora, essa estratégia enfrenta desafios relacionados à escalabilidade e à complexidade computacional, especialmente em cenários com grandes volumes de dados e alta variabilidade nas demandas.

Clusterização de Rotas com Métodos de Alocação de Pacotes: Outra abordagem significativa é a combinação de técnicas de clusterização, como o K-means, com métodos dinâmicos de alocação de pacotes. Loggi propôs uma metodologia que utiliza o histórico de entregas para formar clusters de rotas, facilitando a alocação dinâmica de pacotes dentro desses clusters [43]. Esta abordagem se destaca pela simplicidade e eficiência computacional, sendo especialmente útil em cenários com grandes volumes de dados, onde a clusterização pode reduzir a complexidade do problema e melhorar a eficiência operacional.

Apesar dos avanços mencionados, as estratégias existentes ainda enfrentam desafios significativos, especialmente em relação à dinamicidade do problema e à chegada de novas demandas em tempo real. A necessidade de soluções que não apenas respondam rapidamente às mudanças, mas que também sejam robustas em face da incerteza, permanece um campo fértil para a pesquisa. O desenvolvimento de algoritmos que integrem de forma mais eficaz a informação em tempo real e a previsão de demandas futuras pode oferecer avanços significativos para a área. Além disso, a combinação de técnicas tradicionais com abordagens modernas, como a inteligência artificial e o aprendizado de máquina, pode abrir novas perspectivas para o desen-

volvimento de soluções mais adaptáveis e escaláveis.

Em suma, enquanto a literatura sobre SDVRPs e Last-Mile ICVRP oferece uma sólida base teórica e metodológica, há uma necessidade clara de continuar explorando novas técnicas e abordagens que possam lidar melhor com as complexidades e incertezas do mundo real. A integração dessas abordagens com as demandas práticas das operações logísticas modernas é um desafio que, se superado, pode levar a melhorias significativas na eficiência e na qualidade dos serviços de entrega.

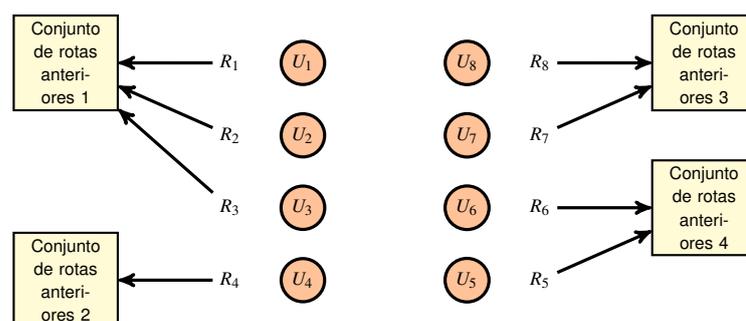
3.2 Last-Mile Incremental Capacitated Vehicle Routing Problem

O Problema de Roteamento de Veículos Capacitados Incrementais de última milha (Last Mile ICVRP) é um tipo específico de SD-CVRP que envolve a alocação dinâmica de novas entregas a UCs com capacidade limitada [44]. Propomos um método baseado em clusterização hierárquico que explora dados históricos para antecipar demandas futuras e otimizar a alocação de rotas. Nossa abordagem é inspirada por pesquisas anteriores sobre SD-CVRP, otimização antecipatória e técnicas de clusterização, ao mesmo tempo em que aborda os desafios específicos do ICVRP no contexto de operações de entrega em larga escala em áreas urbanas.

O problema pode ser descrito como se segue. Dado um conjunto (histórico) de clientes atendidos anteriormente $H = \{v_1, \dots, v_\ell\}$, um conjunto \mathcal{U} de UCs e um conjunto com n novos clientes $D = \{v_{\ell+1}, \dots, v_{\ell+n}\}$, o Last-Mile ICVRP tem como objetivo alocar cada uma das entregas do conjunto D a uma UC.

Cada demanda é alocada, uma por vez, em uma única UC de acordo com a ordem de chegada pré-definida. As UCs tem sua capacidade máxima que, quando atingida, precisa ser despachada para entrega. O problema busca minimizar a distância percorrida pelos veículos ao alocar as entregas às UCs. A Figura 3.1 mostra um exemplo de como as unidades de carregamento podem ser organizadas considerando uma quantidade de rotas anteriores.

Figura 3.1: Exemplo de 8 UCs e 4 conjuntos de rotas anteriores associadas.



Até onde sabemos, apenas os seguintes métodos tratam especificamente do ICVRP na li-

teratura: (i) QRP Sweep (QRPS) e K-means Greedy (KG), ambos disponíveis no repositório Loggibud [5], (ii) o sistema multiagente (MAS) proposto por Fonseca-Galindo et al. [4], e (iii) os algoritmos Greedy TSP (GTSP) e Multiple Scenario Approach (MSA), ambos também propostos em [4]. Com exceção de GTSP e MSA, todos os métodos compartilham uma estrutura comum, composta por uma fase de planejamento (offline) e uma fase de execução (online), que são detalhadas abaixo. No entanto, vale notar que, como será mostrado, tanto QRPS quanto KG violam uma das restrições do ICVRP, que limita o número de UCs no centro de expedição.

3.2.1 K-means Greedy e QRP Sweep

Dado um parâmetro k , na fase de planejamento, ambos os algoritmos dividem a região de entrega em k sub-regiões usando dados históricos, de modo que cada sub-região é associada a um número ilimitado de UCs. Durante a fase de execução, cada pacote recebido é atribuído à sub-região mais próxima. Como pode haver várias UCs associadas a uma sub-região, o pacote recebido é atribuído na UC mais restrita em termos de capacidade. Se nenhuma UC em uma sub-região puder acomodar o pacote, uma nova UC é aberta. Esse processo é repetido até que todos os pacotes sejam alocados. Nesse ponto, o problema do caixeiro-viajante (TSP) é resolvido para determinar a rota de entrega para cada UC.

A principal diferença entre QRPS e KG está em como as sub-regiões são definidas. Em QRPS, um método de varredura é usado para dividir a região de entrega, visando uma distribuição uniforme de pacotes. Em contraste, KG utiliza o algoritmo K-means para definir as sub-regiões.

3.2.2 Greedy TSP e Multiple Scenario Approach

Essas abordagens são baseadas em métodos tradicionais de VRP dinâmico e não possuem uma fase de treinamento. No GTSP, um pacote recebido é alocado de forma gananciosa na UC com o menor custo de inserção, de modo que o custo de inserir um pacote p em uma UC U_i é dado por $cost(p, U_i) = TSP(U_i, p) - TSP(U_i)$, onde $TSP(U_i, p)$ é a distância da rota calculada resolvendo o TSP considerando os pacotes já atribuídos a U_i mais o novo pacote p , e $TSP(U_i)$ é o custo original da rota sem a inclusão do novo pacote p .

De maneira semelhante ao GTSP, no algoritmo MSA, a UC de um pacote recebido é selecionada com base em cenários de previsão, amostrando cenários futuros e computando a decisão que retorna o menor custo total médio entre as amostras. Um cenário de previsão é uma previsão de pedidos futuros baseada em amostragem de padrões geográficos conhecidos de entregas de pacotes, assumindo que a distribuição geográfica permanece consistente ao longo do tempo.

3.2.3 Multi-Agent System

Na fase de planejamento, a abordagem de Fonseca-Galindo et al. [4] utiliza mineração de trajetórias para extrair regras de associação de rotas históricas, identificando padrões frequentes de

entrega representados como sequências de áreas geográficas. Na fase de execução, cada UC é gerenciada por um agente que faz lances para novos pacotes com base nas regras de associação e métricas baseadas em distância geradas na fase de planejamento. Quando uma UC está cheia, o veículo associado à rota é despachado, e a capacidade da UC é renovada. De maneira semelhante a QRPS e KG, as rotas dos veículos despachados são determinadas resolvendo o TSP.



Metodologia

ESTE capítulo apresenta em detalhes o fluxo de trabalho do método proposto, descrevendo cada etapa de sua implementação e aplicação. O objetivo é fornecer uma visão clara e estruturada sobre os procedimentos adotados, destacando como o *Hierarchical Cluster Model* (HCM) foi desenvolvido e aplicado para resolver o *Last-Mile Incremental Capacitated Vehicle Routing Problem* (ICVRP).

4.1 Fluxo de trabalho

A Figura 4.1 ilustra o fluxo de trabalho do método proposto, que chamamos de *Hierarchical Cluster Method* (HCM), compreendendo duas fases distintas: planejamento e execução. A fase de planejamento ocorre de modo *offline* e abrange as seguintes etapas: carregamento do conjunto de dados históricos e organização do modelo de *cluster* hierárquico. Ao fim da primeira etapa, utilizamos o modelo na fase de execução, que é conduzida de modo *online* e envolve a atribuição de novas entregas às UCs na chegada ao centro de expedição mais próximo do destino final. Nas subseções seguintes detalhamos as duas fases.

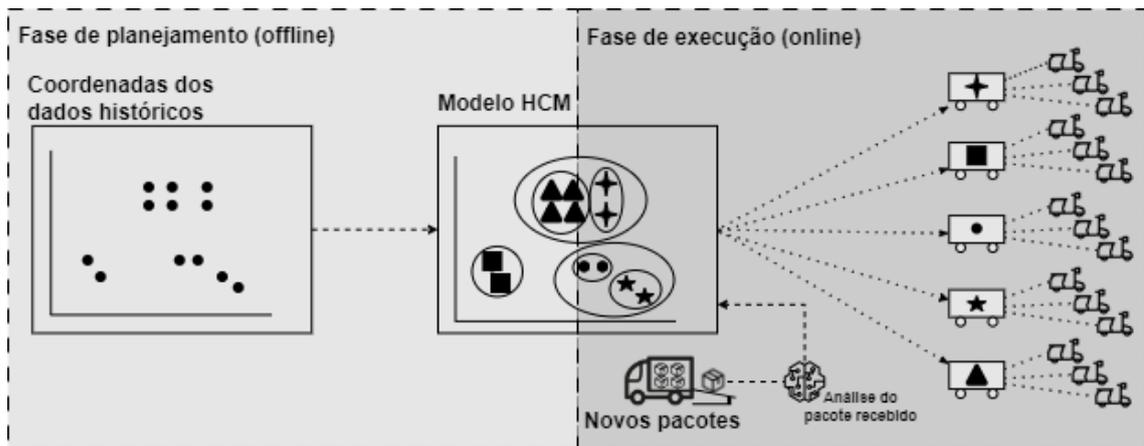


Figura 4.1: Fluxo geral para solução do Last-Mile ICVRP com HCM. À esquerda, mostra-se a organização da fase offline, e à direita, a fase online.

4.2 Fase de planejamento (offline)

O Algoritmo 4.1 apresenta o pseudocódigo da parte *offline* do método proposto, denominado HCM-offline, que tem como principal objetivo gerar um modelo de *cluster* hierárquico com base na análise de dados históricos (conjunto de dados de treinamento). O modelo de *cluster* hierárquico é estruturado em dois níveis: o primeiro nível engloba k *clusters* denotados como C_1, \dots, C_k , e o segundo nível apresenta n_i *sub-clusters* para cada *cluster* C_i . O número total de *sub-clusters* é igual ao número de UCs no centro de expedição, denotado como u , ou seja, $n_1 + n_2 + \dots + n_k = u$.

Algoritmo 4.1 HCM-offline(H, k_{min}, k_{max}, u)**Output:** Hierarchical cluster model M^*

```

1 ( $H_1, H_2$ )  $\leftarrow$  load_training( $H$ ) // Carrega dados históricos e os separa em
   treinamento e pré-teste
2  $S^* \leftarrow \infty$  // Inicializa o melhor valor do modelo
3 forall pairs ( $k, u$ ) where  $k_{min} \leq k \leq k_{max}$  do
4    $C \leftarrow$  K-means( $k, H_1$ ) // Constrói modelo de cluster de primeiro nível
5   ( $n_1, \dots, n_k$ )  $\leftarrow$  UL_allocation( $C, u$ ) // Aloca UCs disponíveis para cada cluster
6   forall  $C_i \in C$  do
7      $SC_i \leftarrow$  K-means( $n_i, C_i$ ) // Constrói modelos de cluster de segundo nível
8    $routes \leftarrow$  HCM-online( $k, u, C, (SC_1, \dots, SC_k), H_2$ ) // Testa o modelo de cluster
   hierárquico
9    $S \leftarrow$  dist(routes) // Calcula a distância das rotas
10  if  $S < S^*$  then
11     $M^* \leftarrow \{C, (SC_1, \dots, SC_k)\}$ 
12     $S^* \leftarrow S$ 
13 return  $M^*$ 

```

O Algoritmo 4.1 recebe como entrada: (i) o conjunto de dados de treinamento $H = \{p_1, \dots, p_l\}$, que compreende informações históricas relacionadas às coordenadas geográficas (latitude e longitude) e volume de locais de entrega anteriores, (ii) número máximo e mínimo de *clusters* no centro de expedição, k_{max} e k_{min} , respectivamente, e (iii) número de UCs no centro de expedição, u . O algoritmo retorna como saída um modelo de *cluster* hierárquico, HCM, tal que $k_{min} \leq k \leq k_{max}$, ou seja, escolhe automaticamente o melhor valor para k .

4.2.1 Carregando conjunto de dados de treinamento

O HCM-offline inicia carregando o conjunto de dados de treinamento e separando esses dados em dois conjuntos, H_1 e H_2 (linha 1). Os primeiros 95% dos itens nesse conjunto de dados (H_1) são usados para construir um modelo de *cluster* hierárquico, e os 5% restantes (H_2) para testar a melhor configuração para k e u . A Figura 4.2 mostra um exemplo da organização das coordenadas presentes nos conjunto de dados.

4.2.2 Construção do primeiro nível HCM

No loop principal do Algoritmo 4.1 (linhas 3-12), o HCM-offline constrói um *cluster* hierárquico para cada valor possível de k e u , e seleciona os melhores valores para eles. O loop principal começa usando K-means para criar o primeiro nível do modelo de *cluster* com base



Figura 4.2: Exemplo de coordenadas dos dados históricos.

no conjunto de dados histórico H_1 (linha 4). A Figura 4.3 mostra um exemplo deste modelo de primeiro nível, considerando $k = 3$.

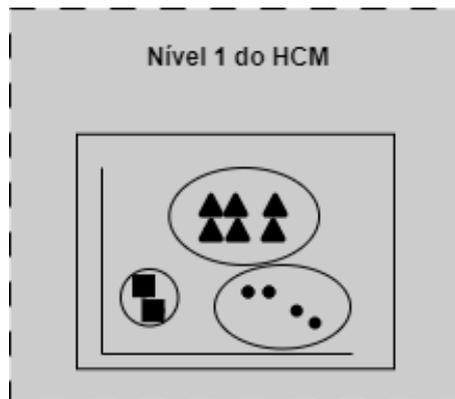


Figura 4.3: Exemplo da aplicação do K-means para agrupar os dados históricos.

4.2.3 Alocação das UCs

Após criar o modelo de primeiro nível, o Algoritmo 4.1 aloca cada uma das UC em um dos k clusters (linha 5). O número de UCs alocadas a um cluster C_i determina o número de sub-clusters n_i nesse cluster. O Algoritmo 4.2 apresenta o pseudocódigo da função `UL_allocation`, que é utilizada para realizar esta alocação. O raciocínio por trás de `UL_allocation` é alocar mais UCs aos clusters que, de acordo com os dados de treinamento, recebem mais entregas. Além disso, precisamos garantir que cada cluster tenha pelo menos uma UC alocada a ele.

Algoritmo 4.2 $UL_allocation(C, u)$ **Output:** Número de sub-clusters n_1, \dots, n_k para cada cluster $C_i \in C$

```

1 forall clusters  $C_i$  in  $C$  do
2    $n_i \leftarrow \left\lceil \frac{|C_i|}{|C|} \times u \right\rceil$  // Determina o número de subclusters para cada cluster
3 while  $n_1 + \dots + n_k > u$  do
4    $i \leftarrow \arg \max(n_1, \dots, n_k)$  // Encontra o cluster com o maior número de
      sub-clusters
5    $n_i \leftarrow n_i - 1$  // Reduz o número de sub-clusters para esse cluster
6 return  $n_1, \dots, n_k$ 

```

4.2.4 Construção do segundo nível HCM

Após definir os valores para n_1, \dots, n_k , HCM-offline cria os modelos de *cluster* de segundo nível (linhas 6 e 7 do Algoritmo 4.1). Ele usa K-means para construir os modelos de segundo nível. A Figura 4.4 mostra um exemplo desses modelos de *cluster* de segundo nível com $k = 3$ e $u = 5$.



Figura 4.4: Exemplo de modelos de cluster de segundo nível considerando $k = 3$ e $u = 5$.

4.2.5 Pré-teste no HCM

Finalmente, nas linhas 8 e 9, o Algoritmo 4.1 testa o modelo com a configuração atual (k, u) . Para testar um modelo hierárquico, aplicamos a segunda parte do conjunto de dados de treinamento, H_2 , no algoritmo *online*, denominado HCM-online, que será explicado na Subseção 4.3. Se a configuração atual (k, u) for melhor que a melhor atual, a melhor atual é atualizada (linhas 11-12). A Figura 4.5 mostra um exemplo do estado final do modelo HCM.

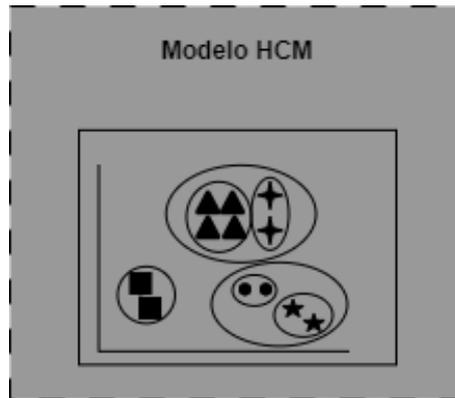


Figura 4.5: Exemplo de modelo HCM com $k=3$ e $u=5$.

4.3 Fase de Execução (online)

O Algoritmo 4.3 apresenta a parte *online* do método proposto, denominada HCM-online, que visa otimizar a alocação de novas entregas às UCs, de forma que a distância das rotas de entrega seja minimizada. Ao receber uma nova entrega, o método utiliza o modelo de *cluster* hierárquico produzido pela parte *offline* do método para estimar a melhor alocação para esta nova entrega.

Algoritmo 4.3 HCM-online($\alpha, u, C, (SC_1, \dots, SC_u), D, W$)

Output: Conjunto de rotas de entrega *routes*

```

1  $(s_1, \dots, s_u) \leftarrow (W, \dots, W)$  // Inicializa a capacidade de cada unidade de
   carregamento
2  $(UL_1, \dots, UL_u) \leftarrow (\emptyset, \dots, \emptyset)$  // Inicializa o conteúdo de cada unidade de
   carregamento
3  $routes \leftarrow \emptyset$  // Inicializa o conjunto de rotas
4 forall  $p \in D$  do
5    $i \leftarrow \text{predict}(p, C)$  // Determina o cluster ao qual  $p$  está associado
6    $j \leftarrow \text{predict}(p, SC_i)$  // Determina o subcluster ao qual  $p$  está associado
7    $UL_j \leftarrow UL_j \cup \{p\}$  // Adiciona o pacote  $p$  à unidade de carregamento  $j$ 
8    $s_j \leftarrow s_j - p.q$  // Atualiza a capacidade da unidade de carregamento  $j$ 
9   if  $s_j \leq \alpha \times W$  then
10     $routes \leftarrow routes \cup \text{TSP\_route}(UL_j)$  // Cria uma nova rota com os pacotes da
       unidade de carregamento  $j$ 
11     $UL_j \leftarrow \emptyset$  // Limpa o conteúdo da unidade de carregamento  $j$ 
12     $s_j \leftarrow W$  // Restaura a capacidade da unidade de carregamento  $j$ 
13 return  $routes$ 

```

4.3.1 Alocação das entregas

O HCM-online retorna uma lista de rotas de veículos e recebe como entrada: (i) o parâmetro do critério de fechamento da UC α , (ii) o número de UCs u , (iii) os modelos de *cluster* de primeiro e segundo níveis C e SC_1, \dots, SC_u , e (iv) a lista de novas entregas D , com sua localização e tamanho. Note que as entradas (ii) e (iii) foram determinadas por HCM-offline.

O Algoritmo 4.3 começa inicializando os conteúdos, bem como a capacidade de cada UC (linhas 1-2). No loop principal (linhas 4-12), ele insere cada pacote recebido $p \in D$ em uma das UC. Especificamente, nas linhas 5 e 6, HCM-online utiliza o modelo de *cluster* hierárquico para determinar a UC para alocar um determinado pacote p . A seleção da UC é baseada na função *predict*, que retorna o *cluster* cujo centróide está mais próximo de um determinado ponto. Em seguida, o novo pacote é inserido na UC escolhida j e a capacidade restante de j é atualizada (linhas 7 e 8).

4.3.2 Organização das rotas

Após a inserção de p , o algoritmo verifica se a capacidade restante de $UL_j (s_j)$ é igual ou inferior a α vezes a capacidade total (linha 9). Se sim, é realizado o despacho dos elementos atuais em UL_j . Isso significa que uma rota considerando os elementos atuais em UL_j é gerada (linha 10) e os elementos em UL_j são limpos (linhas 11 e 12). A geração da rota é realizada resolvendo o TSP para OVRP, pois não há a necessidade do veículo retornar ao centro de expedição. A Figura 4.6 mostra um exemplo de geração de rota. Observamos que a regra de fechamento é verificada após a inserção do pacote porque assume-se que o tamanho de qualquer pacote é inferior a α vezes a capacidade total. O algoritmo retorna na linha 15 uma lista com todas as rotas geradas durante sua execução.

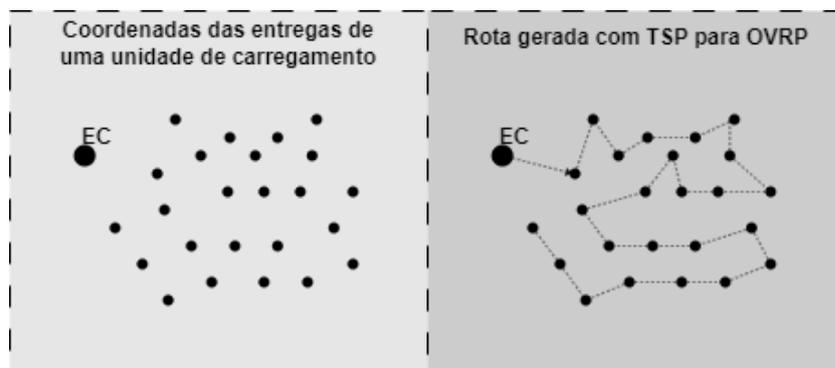


Figura 4.6: Exemplo da organização das rotas para resolver o TSP, desconsiderando a volta ao EC.

5

Resultados e Discussões

Nesta seção apresentamos os resultados obtidos pelos experimentos computacionais realizados para avaliar o método proposto, HCM. Os testes foram realizados em uma máquina Intel i7-10750H com um processador de 2.6 GHz, 16 GB de RAM, executando Ubuntu 20.04.4. Os algoritmos foram desenvolvidos em Python e utilizamos a biblioteca *scikit-learn* [45] juntamente com a biblioteca Numpy [46] para implementar o K-means. As distâncias foram obtidas do OpenStreetMap [26] usando o servidor OSRM [27]. Além disso, adotamos o OR-Tools [47] para resolver problemas de TSP e CVRP.

5.1 Instâncias de benchmark

O método foi testado utilizando o repositório Loggibud [5], que inclui conjuntos de dados e algoritmos de *benchmark* para testar problemas de entrega urbana em larga escala. Este repositório captura a complexidade de cenários operacionais do mundo real e contém informações de entrega em três localidades brasileiras: Pará, Rio de Janeiro e Distrito Federal. As informações de entrega incluem coordenadas de latitude e longitude, peso, capacidade do veículo e localizações dos centros de expedições. Os seguintes datasets do Loggibud [5]¹ foram utilizados para a realização dos experimentos:

- **Pará** (pa-0 e pa-1). Este conjunto de dados consiste em duas regiões: pa-0, que contém 25.967 entregas para treinamento e 9.013 entregas para teste, e pa-1, que contém 372.039 entregas para treinamento e 126.954 para teste, respectivamente. As Figuras 5.1 e 5.2 mostram como os dados históricos estão organizados em cada região de Pará.

¹Loggibud delivery-instances-1.0: <https://loggibud.s3.amazonaws.com/dataset.zip>

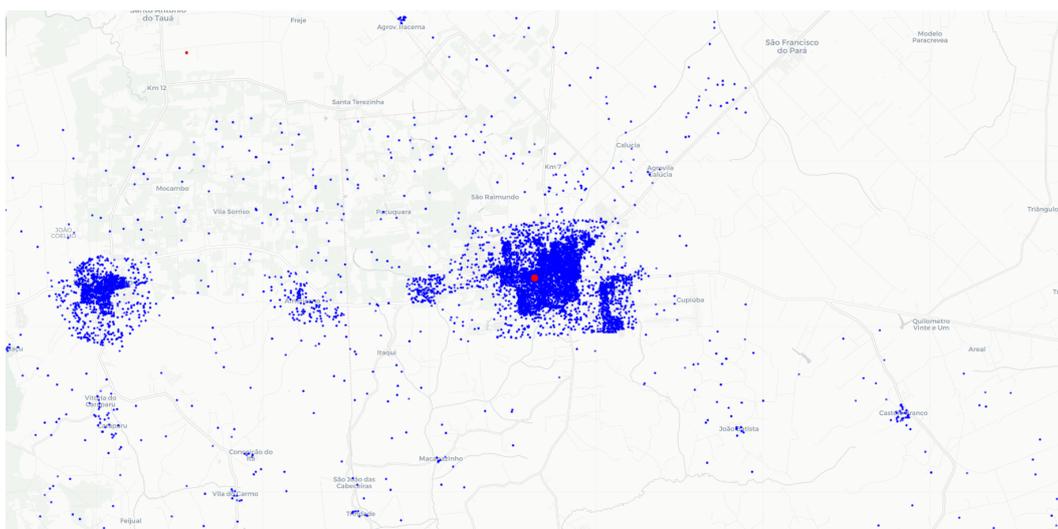


Figura 5.1: Dados históricos presentes em pa-0.

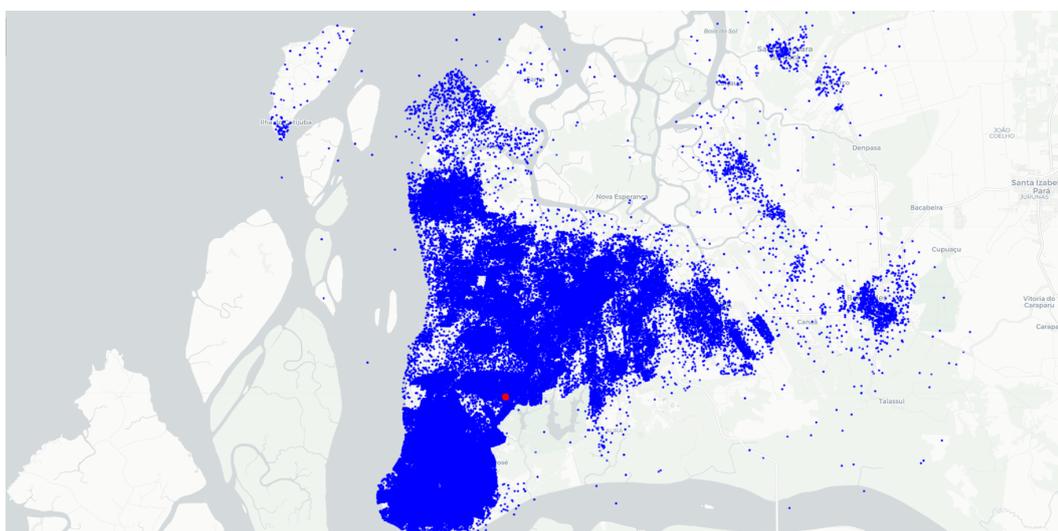


Figura 5.2: Dados históricos presentes em pa-1.

- **Distrito Federal** (df-0, df-1 e df-2). É dividido em três regiões. A primeira região, df-0, compreende 88.552 entregas para treinamento e 28.255 entregas para teste. A segunda região, df-1, consiste em 418.438 entregas de treinamento e 134.601 entregas de teste. A terceira região, df-2, inclui 385.759 entregas de treinamento e 123.802 entregas de teste. As Figuras 5.3, 5.4 e 5.5 mostram como os dados históricos estão organizados em cada região do Distrito Federal.

- **Rio de Janeiro** (rj-0, rj-1, rj-2, rj-3, rj-4 e rj-5). Este conjunto de dados, que representa um dos estados mais proeminentes do Brasil, consiste em seis regiões. O número de entregas de treinamento em cada uma das regiões é, respectivamente, 367.982, 660.734, 191.592, 510.897, 498.743 e 356.175, enquanto o número de entregas de teste é 125.109, 224.935, 65.050, 173.633, 169.745 e 121.497, respectivamente. As Figuras 5.6, 5.7, 5.8, 5.9, 5.10 e 5.11 mostram como os dados históricos estão organizados em cada região do Rio de Janeiro.

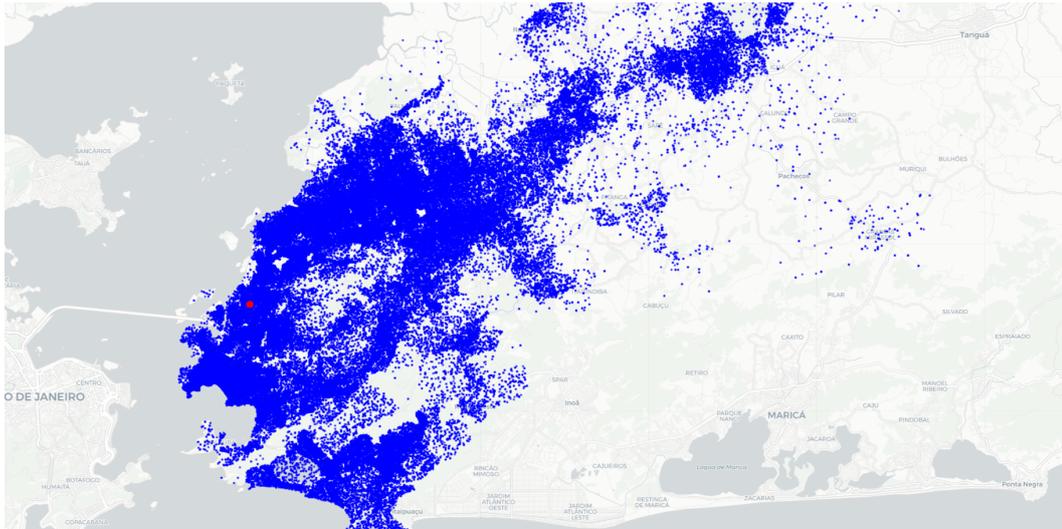


Figura 5.6: Dados históricos presentes em rj-0.

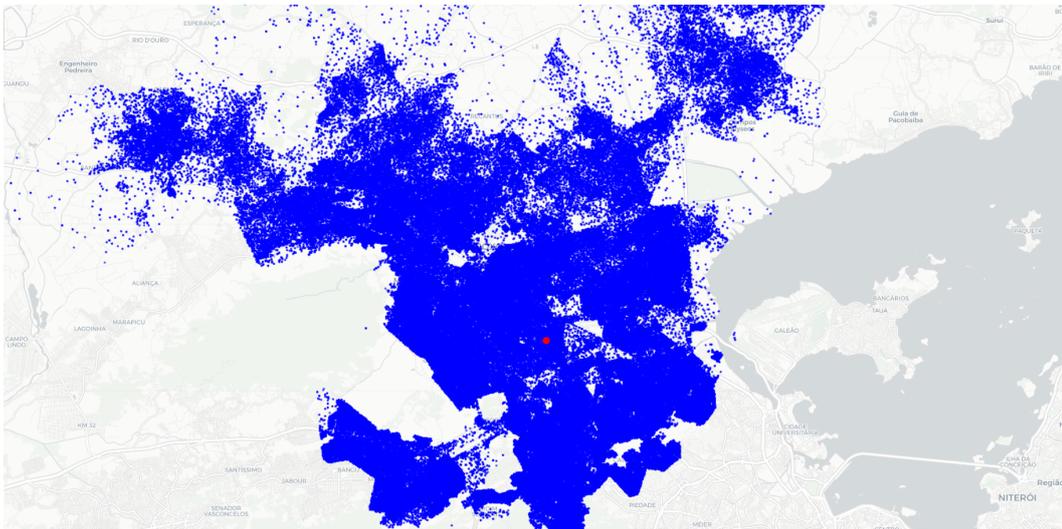


Figura 5.7: Dados históricos presentes em rj-1.

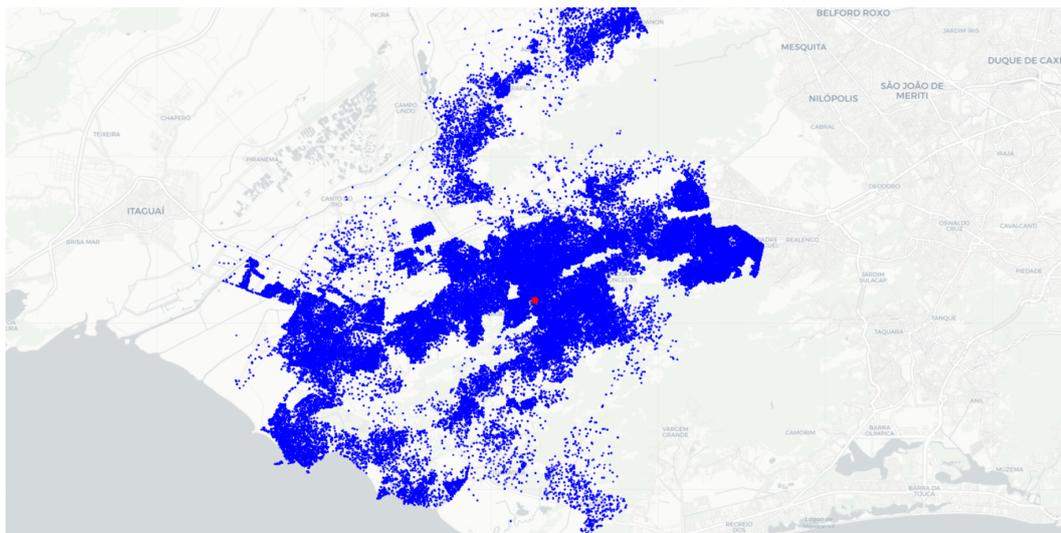


Figura 5.8: Dados históricos presentes em rj-2.

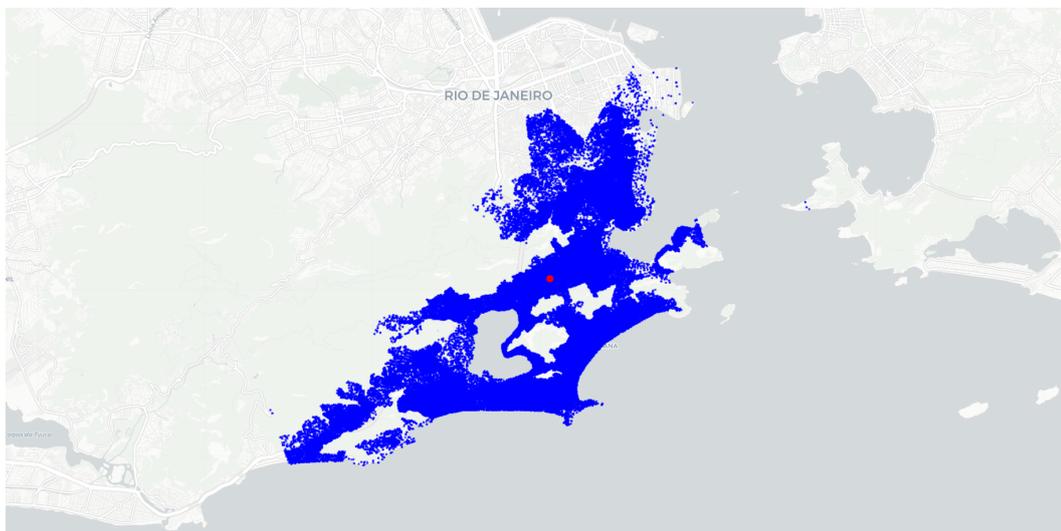


Figura 5.9: Dados históricos presentes em rj-3.

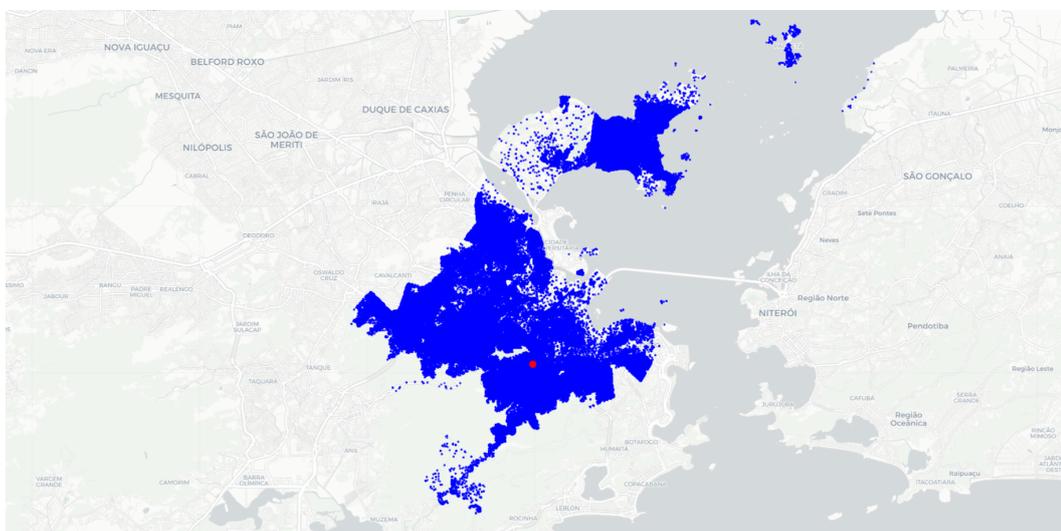


Figura 5.10: Dados históricos presentes em rj-4.

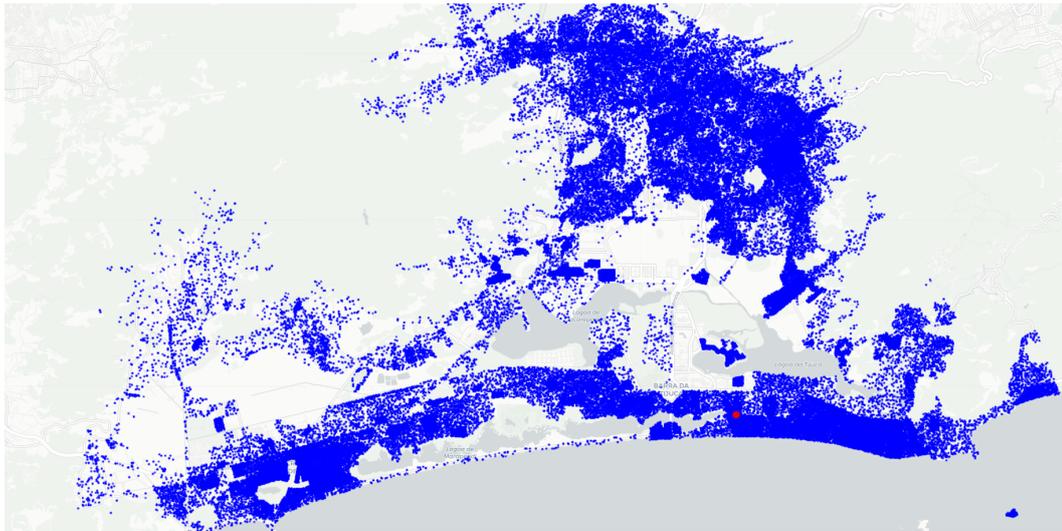


Figura 5.11: Dados históricos presentes em rj-5.

Seguindo [4], assumimos que a capacidade do veículo é restrita a 110 litros, e a UC é fechada quando a carga atinge 80% da capacidade do veículo.

5.2 Algoritmos de benchmark

Comparamos a nova abordagem com outros três métodos dinâmicos: QRPS, KG e GTSP, que foram descritos nas Seções 3.2.1 e 3.2.2. Os códigos-fonte para QRPS e KG foram obtidos do repositório Loggibud, enquanto GTSP foi implementado com base nas descrições fornecidas por seus autores [4].

Também comparamos a abordagem com um método *offline* completo, o *solver* VRP estático disponível no OR-Tools. Seguindo a configuração em [4], o OR-Tools foi configurado para empregar o *Path Most Constrained Arc* para encontrar uma solução inicial e *Guided Local Search* como meta-heurística de busca local.

Infelizmente, uma comparação direta com MAS e MSA não foi possível, pois seus códigos-fonte não estavam disponíveis e os detalhes fornecidos nas publicações dos autores foram insuficientes para permitir a reprodução precisa. Segundo [4], MAS teve em sua distância percorrida uma diferença média de 33% acima dos resultados encontrados pelo VRP estático, CVRP.

5.3 Configuração de parâmetros

O único parâmetro no método proposto são uma tupla, (k_{min}, k_{max}) , e um número inteiro, u . Para u , foi utilizado o valor 28 [4], tendo como base as restrições de operação da Loggi. Os parâmetros restantes foram configurados da seguinte forma: $k_{min} = 4$ e $k_{max} = 25$. Com esta configuração de parâmetros, conforme descrito na Subseção 4.2, o método determina automati-

amente a melhor configuração para o número de *clusters* k , considerando o número de unidades de carga $u = 28$.

No caso de QRPS e KG, o único parâmetro é o número de sub-regiões k . Avaliamos esses algoritmos com valores de k de $\{24, 26, 28\}$, e apenas os resultados usando as melhores configurações são apresentados nesta seção. Para QRPS, o único parâmetro é o número de unidades de carga u , que definimos como 28.

5.4 Protocolo Experimental

Como mencionado anteriormente, no problema de roteamento considerado, as rotas são geradas dinamicamente, assumindo que os pacotes chegam em um fluxo contínuo. Para fins de avaliação, utilizamos o algoritmo estático CVRP do OR-Tools como linha de base para comparação com os outros algoritmos dinâmicos. Para gerar soluções OR-Tools, agrupamos as entregas de teste dentro de cada região 5 *batches*, com cada *batch* contendo 2.500 entregas selecionadas aleatoriamente do subconjunto de dados. Cada lote foi então resolvido usando o OR-Tools. Para cada região, chamamos de resultado offline a distância cumulativa percorrida pelas soluções em todos os *batches*. A escolha de um tamanho de *batch* de 2.500 segue a mesma abordagem adotada por [4] e, como em [4], usamos um limite de tempo de 700 segundos para o OR-Tools.

Para avaliar o desempenho da abordagem e de outros métodos dinâmicos, procedemos da seguinte forma: Primeiro, para cada região, a fase de treinamento de cada método foi executada usando as entregas de treinamento. Em seguida, a fase *online* de cada algoritmo foi executada usando os *batches* descritos anteriormente para cada região. Como os métodos dinâmicos utilizam o OR-Tools para resolver o TSP, definimos um limite de tempo de 1 segundo para esta resolução.

5.5 Comparação com outros modelos

Para exibir e comparar os resultados de cada região, foram utilizadas tabelas contendo cinco colunas: a primeira coluna refere-se à identificação do modelo, a segunda fornece a distância total percorrida pelas rotas, a terceira coluna informa a quantidade de rotas geradas, assim como a quarta coluna refere-se a quantidade média de pacotes por rota, na quinta coluna há o tempo de execução em minutos. Os parâmetros necessários para os modelos foram definidos para 28 (valor de k no modelo KG, quantidade de sub-regiões no modelo QRP, quantidade de unidades de carregamento no modelo GTSP), referente ao número total de unidades de carregamento disponíveis.

Pará. Este estado brasileiro contém duas regiões onde há a presença de um centro de expedição, pa-0 e pa-1. No modelo HCM, o parâmetro k escolhido foi: 20 para pa-0 e 13 para pa-1. As Tabelas 5.1 e 5.2 apresentam os resultados relacionados à pa-0 e pa-1, respectivamente.

Tabela 5.1: Resultados experimentais em pa-0

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	19.188,055	824	14,08
KG	19.644,122	820	14,17
QRPS	33.445,479	832	14,08
GTSP	41.084,173	880	5889,95

Tabela 5.2: Resultados experimentais em pa-1

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	19.730,051	822	14,57
KG	20.167,421	825	14,60
QRPS	33.072,365	812	14,10
GTSP	45.445,578	879	5894,48

Distrito Federal. Contém três regiões que foram usados para gerar os resultados: df-0, df-1 e df-2. No modelo HCM, o parâmetro k escolhido foi: 8 para df-0, 14 para df-1 e 4 para df-2. As Tabelas 5.3, 5.4 e 5.5 apresentam os resultados relacionados à df-0, df-1 e df-2, respectivamente.

Tabela 5.3: Resultados experimentais em df-0

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	23.644,622	820	14,18
KG	24.806,384	824	14,38
QRPS	32.974,783	818	13,97
GTSP	49.868,080	877	5895,45

Tabela 5.4: Resultados experimentais em df-1

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	19.811,402	820	14,28
KG	19.731,822	815	15,08
QRPS	29.988,618	818	13,91
GTSP	48.915,356	881	5848,07

Tabela 5.5: Resultados experimentais em df-2

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	22.059,766	820	14,21
KG	21.860,915	817	14,78
QRPS	31.425,947	816	13,97
GTSP	53.854,424	876	5896,07

Rio de Janeiro. O estado possui o maior número de instâncias no banco de dados. O parâmetro k escolhido para o modelo HCM das seis regiões, foi: 14 para rj-0, 17 para rj-1, 10 para rj-2, 23 para rj-3, 4 para rj-4 e 7 para rj-5. Os resultados experimentais das regiões estão presentes na Tabela 5.6, 5.7, 5.8, 5.9, 5.10 e 5.11, respectivamente.

Tabela 5.6: Resultados experimentais em rj-0

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	21.054,704	827	14,30
KG	21.481,774	823	15,16
QRPS	26.970,245	821	14,05
GTSP	52.185,929	883	5895,07

Tabela 5.7: Resultados experimentais em rj-1

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	26.412,342	831	14,35
KG	26.367,136	832	15,50
QRPS	33.137,182	826	14,08
GTSP	70.784,179	888	5895,81

Tabela 5.8: Resultados experimentais em rj-2

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	20.074,350	823	14,19
KG	20.173,107	824	14,84
QRPS	27.675,392	826	14,15
GTSP	50.311,294	882	5896,84

Tabela 5.9: Resultados experimentais em rj-3

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	9.544,659	828	14,24
KG	9.445,797	826	15,17
QRPS	11.944,115	825	14,25
GTSP	23.117,018	887	5892,87

Tabela 5.10: Resultados experimentais em rj-4

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	14,966,971	819	13,97
KG	14,975,044	818	13,91
QRPS	18.199,428	819	14,08
GTSP	35.535,766	876	5849,69

Tabela 5.11: Resultados experimentais em rj-5

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	17.268,582	827	13,88
KG	17.662,689	828	14,25
QRPS	20.491,119	816	13,91
GTSP	43.981,242	882	5846,17

5.5.1 Resultados totais

Dessa forma, os resultados totais dos algoritmos são mostrados na Figura 5.12.

Tabela 5.12: Resultados experimentais

Modelo	Distância (km)	Rotas	Tempo de execução (min)
HCM	213.755,504	9.061	156,38
KG	216.316,211	9.052	163,74
QRPS	299.324,673	9.029	154,53
GTSP	515.083,039	9.691	64706,28

Abaixo, a figura 5.12.a e a figura 5.12.b apresentam o tempo de execução e a distância total para o método proposto e para o VRP Estático dos experimentos realizados. Além disso, a figura 5.12.c mostra a relação Proposto/CVRP, ou seja, para cada cenário testado, a proporção de resultados obtidos pelo modelo proposto em comparação com a metaheurística estática.

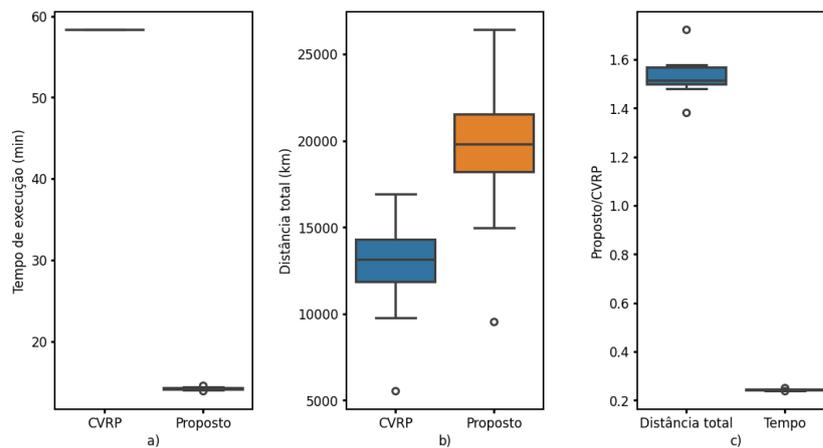


Figura 5.12: Diferença entre a distância total e o tempo de execução entre o método proposto e o CVRP.

O HCM apresentou uma diferença média de 52% a mais na distância percorrida em comparação aos resultados do VRP estático, utilizando aproximadamente 25% do tempo de execução requerido pelo VRP estático. Conforme mencionado na Subseção 5.2, não foi possível realizar uma comparação direta com os algoritmos MAS e MSA. Entretanto, de acordo com [4], o MAS apresentou uma diferença média de 33% a mais na distância percorrida em relação ao VRP estático (CVRP). Assim, o HCM teve, em média, uma diferença de 19% a mais na distância percorrida em relação ao MAS.

6

Conclusão

ESTE trabalho foi motivado pela crescente demanda por soluções eficientes de roteamento de veículos em um cenário de entregas em larga escala, especialmente no contexto do *e-commerce*. Foi apresentado o problema do *Last-Mile Incremental Capacitated Vehicle Routing Problem* (ICVRP), um caso particular do *Stochastic and Dynamic Capacitated Vehicle Routing Problems* (SDVRP), e um grande desafio para empresas de logística que operam em ambientes urbanos utilizando o modelo de economia compartilhada. O ICVRP exige a alocação eficiente de novas entregas a unidades de carregamento (UCs) com capacidade limitada, considerando a chegada contínua de novas demandas e a necessidade de minimizar a distância total percorrida.

O método tradicional de resolução do *Capacitated Vehicle Routing Problem* (CVRP), utilizado como linha de base neste estudo, também chamado de VRP estático, apresenta limitações significativas para o ICVRP. O CVRP assume que todas as entregas são conhecidas com antecedência, o que não se aplica a cenários dinâmicos. Além disso, a busca por soluções ótimas no CVRP pode ser computacionalmente custosa, exigindo tempo de processamento considerável.

Para superar essas limitações, foi realizado um estudo na literatura para melhor compreender o problema e identificar algoritmos que visem sua resolução. A partir de tal análise, esse trabalho propõe o *Hierarchical Cluster Method*, (HCM), um método que utiliza *cluster* hierárquico para antecipar demandas futuras e otimizar a alocação de rotas. A metodologia descreve a implementação do modelo. O HCM é composto por duas fases: planejamento *offline*, onde um modelo de *cluster* hierárquico é construído com base em dados históricos, e execução *online*, onde o modelo é utilizado para alocar novas entregas às unidades de carregamento (UCs) e gerar rotas de entrega.

Por fim, foram apresentados os resultados dos experimentos computacionais com conjuntos de dados disponibilizados pelo repositório *Loggibud*, da Loggi. Os experimentos evidenciaram a eficácia do HCM, que apresentou um desempenho competitivo em comparação com outras abordagens existentes (QRPS, KG, GTSP, MAS, MSA), destacando-se pela eficiência no roteamento. O HCM obteve resultados expressivos nos modelos que foram testados com os mesmos

conjuntos de dados, como o *K-means Greedy* (KG), o *QRP Sweep* (QRPS) e o *Greedy TSP* (GTSP). O modelo proposto apresentou superioridade quanto à distância percorrida, além de tempos de execução e quantidade de rotas geradas bastante satisfatórios.

De modo geral, o modelo proposto teve em sua distância percorrida uma diferença média de 52% acima dos resultados encontrados pelo VRP estático, CVRP, enquanto os outros algoritmos de comparação tiveram uma diferença maior, demonstrando a capacidade do HCM em gerar soluções de qualidade em um tempo considerável, sendo o segundo algoritmo mais rápido em termos de tempo total de execução, com um tempo de execução equivalente a aproximadamente 25% do VRP estático.

Embora o modelo proposto tenha apresentado resultados competitivos em relação às abordagens comparadas diretamente, ele obteve uma diferença de 19% maior na distância percorrida em comparação ao MAS, indicando que há espaço para melhorias na abordagem.

O HCM, com sua capacidade de lidar com a complexidade do ICVRP, demonstra ser uma abordagem promissora para otimizar a alocação de entregas em cenários de Last-Mile em constante evolução. A sua aplicação em cenários reais pode contribuir para a redução de custos, a otimização do tempo de entrega e a melhoria da experiência do cliente.

Conclui-se que o HCM se apresenta como uma opção válida para resolver o ICVRP, oferecendo uma abordagem eficiente e escalável para lidar com a dinamicidade e incerteza inerentes a esse problema. Os resultados obtidos demonstram a capacidade do modelo em gerar soluções de qualidade em um tempo considerável, superando outros algoritmos da literatura. Com as melhorias sugeridas, como um melhor gerenciamento das UCs e uma análise mais detalhada dos dados históricos, o método pode alcançar um desempenho ainda mais elevado e se estabelecer como uma ferramenta fundamental para aplicações em logística.

6.1 Trabalhos Futuros

Embora o HCM tenha demonstrado resultados promissores, algumas áreas de aprimoramento podem ser exploradas para otimizar ainda mais a solução proposta. As seguintes direções podem ser sugeridas com objetivo de aumentar a adaptabilidade do HCM a cenários mais complexos e dinâmicos, tornando-o ainda mais robusto e eficaz.

- Aperfeiçoamento do gerenciamento das unidades de carregamento (UCs) e sua alocação em relação aos *sub-clusters* (clusters dos modelos do segundo nível), o que pode impactar significativamente a eficiência do sistema.
- Análise mais detalhada dos dados históricos pode melhorar a previsão de padrões de demanda, otimizando a alocação de entregas e auxiliando na escolha do parâmetro k .
- Análise para adição de novos níveis na clusterização hierárquica.

Referências bibliográficas

- [1] Eduardo Magossi. Receita de vendas on-line sobe 41% em 2020. valor.globo.com, 2021. [Online; acessado em 28 Março 2021].
- [2] eDRONE. Dados do ecommerce no brasil, 2024. URL <https://edrone.me/pt/blog/dados-ecommerce-brasil>. Acesso em 3 julho 2024.
- [3] Marcone Jamilson Freitas Souza. Inteligência computacional para otimização. *Notas de aula, Departamento de Computação, Universidade Federal de Ouro Preto, disponível em <http://www.decom.ufop.br/prof/marcone/InteligenciaComputacional/InteligenciaComputacional.pdf>*, 6, 2008.
- [4] J. Camilo Fonseca-Galindo, Gabriela de C. Surita, J. Maia Neto, C. Leite de Castro, and A. Paim Lemos. A multi-agent system for solving the dynamic capacitated vehicle routing problem with stochastic customers using trajectory data mining. *Expert Systems with Applications*, 195:116602, 2022. DOI 10.1016/j.eswa.2022.116602.
- [5] Loggi. Loggibud: Loggi’s vehicle routing problem solver, 2021. URL <https://github.com/loggi/loggibud>. [Online; acessado em 26 Fevereiro 2021].
- [6] Onfleet. First mile, middle mile, last mile: What’s the difference?, 2024. URL <https://onfleet.com/blog/first-mile-middle-mile-last-mile/>. Accessed: 2024-11-15.
- [7] Vinícius César Sacramento. 50 tons de first, middle e last mile, 2024. URL <https://pt.linkedin.com/pulse/50-first-middle-e-last-mile-vin%C3%ADcius-c%C3%A9sar-sacramento-blz9f>. Publicado no LinkedIn. Acesso em: 2024-11-15.
- [8] S. Skiena. Graph problems: Np-hard. 2020. DOI 10.1007/978-3-030-54256-6_19.
- [9] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

- [10] Aziz Ouabarab. Solving combinatorial optimization problems. *Discrete Cuckoo Search for Combinatorial Optimization*, 2020. DOI [10.1007/978-981-15-3836-0_3](https://doi.org/10.1007/978-981-15-3836-0_3).
- [11] Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi. The traveling salesman problem. *Handbooks in operations research and management science*, 7:225–330, 1995.
- [12] Karla L Hoffman, Manfred Padberg, Giovanni Rinaldi, et al. Traveling salesman problem. *Encyclopedia of operations research and management science*, 1:1573–1578, 2013.
- [13] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.
- [14] Yufei Cao, Jianhua Song, and Xiaoping Zhang. Heuristic algorithms for open vehicle routing problem with min-max objective. *Journal of Software*, 9(11):2929–2934, 2014.
- [15] Harilaos N. Psaraftis, Min Wen, and Christos A. Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 2016. DOI [10.1002/net.21628](https://doi.org/10.1002/net.21628).
- [16] Michael R. Swihart and Jason D. Papastavrou. A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *European Journal of Operational Research*, 1999. DOI [10.1016/s0377-2217\(98\)00260-4](https://doi.org/10.1016/s0377-2217(98)00260-4).
- [17] Ulrike Ritzinger, Jakob Puchinger, and Richard F Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231, 2016.
- [18] Harilaos N. Psaraftis. *Dynamic Vehicle Routing Problems*, pages 223–248. North-Holland, 1988.
- [19] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.
- [20] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*.*[Internet]*, 9(1):381–386, 2020.
- [21] Susmita Ray. A quick review of machine learning algorithms. In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 35–39. IEEE, 2019.
- [22] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3(19-48):5–1, 2010.
- [23] Hans-Hermann Bock. Clustering methods: a history of k-means algorithms. *Selected contributions in data analysis and classification*, pages 161–172, 2007.

- [24] Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in neural information processing systems*, 16, 2003.
- [25] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. ISSN 0167-8655. DOI <https://doi.org/10.1016/j.patrec.2009.09.011>. URL <https://www.sciencedirect.com/science/article/pii/S0167865509002323>. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).
- [26] Openstreetmap: Collaborative mapping project, 2020. URL <https://www.openstreetmap.org/#map=4/-15.13/-53.19>. [Online; acessado em 28 Setembro 2020].
- [27] Open source routing machine (osrm), 2020. URL <https://project-osrm.org/>. [Online; acessado em 28 Setembro 2020].
- [28] M. Bernardo and J. Pannek. Robust solution approach for the dynamic and stochastic vehicle routing problem. *Journal of Advanced Transportation*, 2018:1–11, 2018. DOI [10.1155/2018/9848104](https://doi.org/10.1155/2018/9848104).
- [29] D. Barbucha and P. Jędrzejowicz. Agent-based approach to the dynamic vehicle routing problem. In *Advances in Intelligent and Soft Computing*, pages 169–178. Springer Berlin Heidelberg, 2009. DOI [10.1007/978-3-642-00487-2_18](https://doi.org/10.1007/978-3-642-00487-2_18).
- [30] Warren B. Powell and Huseyin Topaloglu. Stochastic programming in transportation and logistics. In *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 555–635. Elsevier, 2003. DOI [10.1016/S0927-0507\(03\)10009-6](https://doi.org/10.1016/S0927-0507(03)10009-6).
- [31] Warren B. Powell and Huseyin Topaloglu. 12. fleet management. In *Applications of Stochastic Programming*, pages 185–215. SIAM, 2005.
- [32] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. *European J. Operational Res.*, 225(1):1–11, 2013. DOI [10.1016/j.ejor.2012.08.015](https://doi.org/10.1016/j.ejor.2012.08.015).
- [33] H. N. Psaraftis. Dynamic vehicle routing: Status and prospects. *Ann. Operations Res.*, 61(1):143–164, 1995. DOI [10.1007/bf02098286](https://doi.org/10.1007/bf02098286).
- [34] A. Larsen, O. B.G. Madsen, and M. M. Solomon. *Recent Developments in Dynamic Vehicle Routing Systems*, pages 199–218. Springer US, 2008. ISBN 978-0-387-77778-8. DOI [10.1007/978-0-387-77778-8_9](https://doi.org/10.1007/978-0-387-77778-8_9).

- [35] P. Jaillet and M. R. Wagner. Online vehicle routing problems: A survey. In *Operations Research/Computer Science Interfaces*, pages 221–237. Springer US, 2008. DOI [10.1007/978-0-387-77778-8_10](https://doi.org/10.1007/978-0-387-77778-8_10).
- [36] S. Schorpp. Dynamic transportation problems. In *Dynamic Fleet Management for International Truck Transportation*, pages 11–29. Gabler, 2011. DOI [10.1007/978-3-8349-6675-9_2](https://doi.org/10.1007/978-3-8349-6675-9_2).
- [37] Warren B Powell. A comparative review of alternative algorithms for the dynamic vehicle allocation problem. In B. Golden and A. Assad, editors, *Vehicle routing: methods and studies. (Studies in Management Science and Systems, Volume 16)*, pages 249–291. Amsterdam ; New York : North-Holland, 1988.
- [38] Barrett W Thomas and Chelsea C White Iii. Anticipatory route selection. *Transportation Science*, 38(4):473–487, 2004.
- [39] Marlin W. Ulmer, Justin C. Goodson, Dirk C. Mattfeld, and Marco Hennig. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 2019. DOI [10.1287/trsc.2017.0767](https://doi.org/10.1287/trsc.2017.0767).
- [40] Dimitris Bertsimas, Philippe Chervi, and Michael Peterson. Computational approaches to stochastic vehicle routing problems. *Transportation science*, 29(4):342–352, 1995.
- [41] Chrysanthos E. Gounaris, Panagiotis P. Repoussis, Christos D. Tarantilis, Wolfram Wiesemann, and Christodoulos A. Floudas. An adaptive memory programming framework for the robust capacitated vehicle routing problem. *Transportation Science*, 2016. DOI [10.1287/trsc.2014.0559](https://doi.org/10.1287/trsc.2014.0559).
- [42] P. Van Hentenryck and R. Bent. *Online Stochastic Combinatorial Optimization*. EBSCO ebook academic collection. MIT Press, 2009. ISBN 9780262513470. URL https://books.google.ca/books?id=vEI_PwAACAAJ.
- [43] Gabriela Surita and José C. Maldonado. Evolução das Soluções e Experiência de Roteirização na Loggi. <https://www.youtube.com/watch?v=IDmaN7ima7k>, 2020. [Online; acessado em 28 Março 2021].
- [44] Loggi. Task 2, loggibud. <https://github.com/loggi/loggibud?tab=readme-ov-file#task-2---last-mile-incremental-capacitated-vehicle-routing-problem>, 2019. Acessado em: 12 nov. 2021.
- [45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,

- M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [46] NumPy developers. Numpy: The fundamental package for scientific computing with Python, 2021. URL <https://numpy.org/>. [Online; acessado em 28 Abril 2021].
- [47] Google Optimization Tools Contributors. Or-tools: Google optimization tools, 2021. URL <https://developers.google.com/optimization?hl=pt-br>. [Online; acessado em 18 Março 2021].