

Relatório Técnico: Roteamento e Logística - AED2

Carlos Henrique Gonçalves Batista*

Guilherme Frazão Fernandes[†] Pedro Antônio Maciel Saraiva[‡]

July 2024

Abstract

Este relatório técnico apresenta a modelagem, implementação e avaliação de um sistema de roteamento de múltiplos veículos utilizando grafos. Utilizamos o OpenStreetMap para os dados de mapas, as bibliotecas NetworkX e numpy para a manipulação dos grafos, folium para visualização e algoritmos como A* e Dijkstra, com foco especial no Dijkstra.

1 Introdução

O problema de roteamento de múltiplos veículos (VRP - Vehicle Routing Problem) em diferentes pontos é fundamental na área de logística, pois visa otimizar rotas de entrega, reduzir custos operacionais e melhorar a eficiência dos serviços de transporte. O VRP possui diversas subvariações, como o VRP com janelas de tempo (VRPTW), onde cada ponto de entrega deve ser visitado dentro de um intervalo de tempo específico, e o VRP com capacidade (CVRP), onde cada veículo tem uma capacidade limitada de carga. Escolhemos este tema devido à sua relevância prática e ao desafio técnico envolvido na implementação de algoritmos de roteamento eficientes.

2 Modelagem do Problema

Modelamos o problema utilizando grafos, onde os vértices representam pontos de interesse (e.g., locais de entrega) e as arestas representam as possíveis rotas entre esses pontos. Utilizamos os seguintes elementos:

*UFG

†UFG

‡UFG

- **Vértices:** Pontos de entrega e pontos de início/fim dos veículos.
- **Arestas:** Rotas possíveis entre os pontos, com pesos baseados na distância ou tempo de viagem.

Para a construção e manipulação do grafo, utilizamos as bibliotecas **NetworkX** e **numpy**. O OpenStreetMap foi utilizado como fonte de dados geográficos para obter a rede de ruas e caminhos.

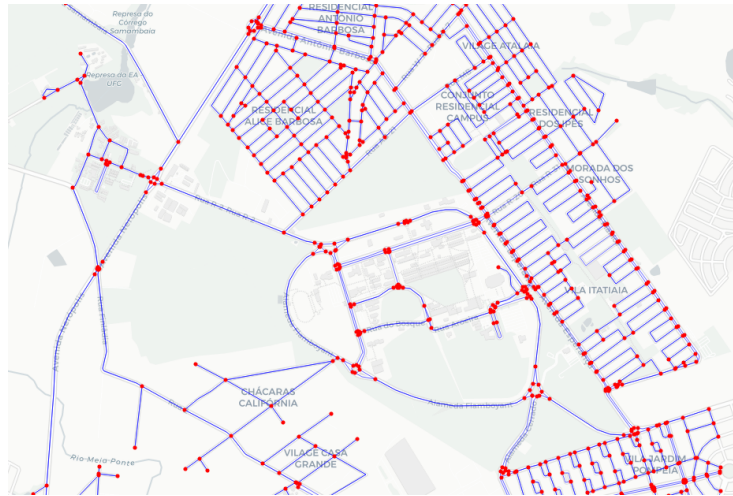


Figure 1: Modelagem do mapa utilizando dados do OpenStreetMap.

3 Implementação

A implementação da solução foi realizada em Python, utilizando as seguintes bibliotecas e ferramentas:

- **OpenStreetMap:** Fonte de dados geográficos.
- **Ortools:** Para operações dentro dos grafos.
- **NetworkX:** Para a criação e manipulação dos grafos.
- **numpy:** Para operações numéricas e manipulação de dados.
- **folium:** Para visualização das rotas.

Implementamos algoritmos de roteamento como A* e Dijkstra, com foco especial no algoritmo de Dijkstra devido à sua eficiência em encontrar o caminho mais curto em grafos com pesos não negativos. A solução foi projetada para calcular rotas ótimas para múltiplos veículos, minimizando a distância total percorrida ou o tempo de viagem.

4 Demonstração da Solução

A seguir, apresentamos um exemplo de entrada e saída do sistema:

- **Entrada:** Coordenadas dos pontos de entrega e pontos de início/fim dos veículos.
- **Saída:** Rotas otimizadas para cada veículo, visualizadas em um mapa utilizando `folium`, que pode ser aberto com o HTML gerado. Disponível no nosso Código-Fonte.

Os resultados demonstram a eficácia do algoritmo de Dijkstra em encontrar rotas otimizadas, mesmo em redes complexas de ruas.



Figure 2: Rotas geradas para múltiplos veículos utilizando o algoritmo de Dijkstra.

5 Conclusão

O desenvolvimento deste sistema de roteamento de múltiplos veículos apresentou vários desafios, como a integração de dados geográficos e a implementação

eficiente de algoritmos de roteamento. Aprendemos sobre a importância de escolher algoritmos adequados para o tipo de grafo e a necessidade de otimizações específicas para lidar com grandes volumes de dados. Futuras melhorias podem incluir a consideração de restrições adicionais, como janelas de tempo e capacidades dos veículos.

6 Código-Fonte e Slide

O código-fonte completo da nossa implementação está disponível no GitHub no seguinte link:

https://github.com/guilhermefraza/AED2_roteamento_de_veiculos

Os slides da nossa apresentação está disponível no link do Canvas abaixo:

https://www.canva.com/design/DAGKczV-YcY/a1INXUnFbWP7AynxAVHriA/edit?utm_content=DAGKczV-YcY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton