Usando Problema do Caixeiro Viajante para Encontrar Rota de Ônibus Público

Marcos Vinícius Tenacol Coêlho [1], Guilherme Almeida da Luz [1]

[1]Departamento de Ciência da Computação
Universidade Federal de Roraima (UFRR) - Boa Vista, RR - Brasil

marcosvinicius.bv@hotmail.com, guilhermealmeidadaluz5@gmail.com

Abstract. *This work addresses the implementation of an approximate algorithm for the multi-objective Traveling Salesman Problem using the Simulated Annealing meta-heuristic technique, applied to bus stops that make up urban public transport lines in the city of Boa Vista (Roraima, Brazil).*

Resumo. *Este trabalho aborda a implementação de um algoritmo aproximado para o problema do Caixeiro Viajante multi-objetivo utilizando a técnica de meta-heurística Simulated Annealing (recozimento simulado), aplicando sobre paradas de ônibus que compõem linhas de transporte público urbano da cidade de Boa Vista (Roraima, Brasil).*

1. Introdução

O Problema do Caixeiro Viajante (TSP - Traveling Salesperson Problem, em inglês) consiste na visitação de vértices de um grafo partindo de um ponto inicial e alcançando todos os outros pontos uma vez e sem repetição, até retornar ao ponto inicial, completando assim um ciclo hamiltoniano. Usado em muitos estudos na área de computação e aplicações de logística, se apresenta como um campo desafiador dada a sua complexidade de otimização, o que o classifica como problema NP-completo, ou seja, não há solução eficiente conhecida em tempo polinomial.

Diversas abordagens aproximadas, como métodos heurísticos e gulosos, buscam soluções próximas do ótimo custo com uma razão de aproximação baixa. Soluções exatas, que testam todas as possibilidades, tornam-se inviáveis para grafos grandes devido ao crescimento exponencial do espaço de busca..

Para este trabalho, foi utilizada a técnica de meta-heurística Simulated Annealing (Recozimento Simulado), um algoritmo de otimização utilizado para designar um custo ideal ou próximo dele, que consegue comportar o problema do Caixeiro Viajante Multiobjetivo.

2. Modelagem do Problema e Dados Utilizados

Com base no artigo “Practical Approach for Solving School Bus Problems” [1] que tem como objetivo o roteamento de ônibus escolares em uma região, bem como, no uso de Goal Programming (Programação de Objetivo), a modelagem do problema se dá em atender os seguintes requisitos simultâneamente: (i) minimização do tempo total de percurso, e (ii) maximização do número de passageiros transportados. Dessa forma, a meta é planejar uma rota que ligue de maneira eficiente o Centro da cidade com a Universidade Federal de Roraima (UFRR), equilibrando os dois pesos.

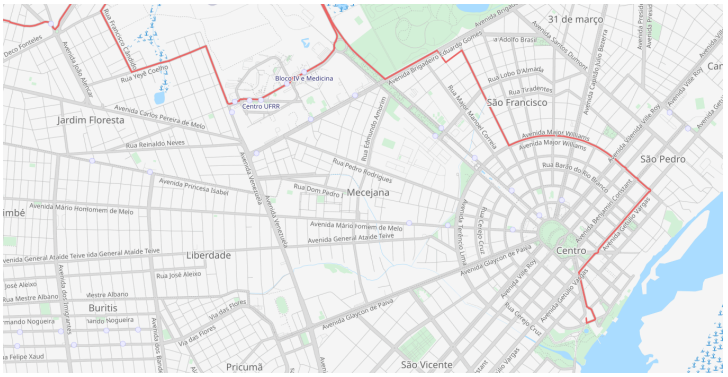

Figura 1. Mapa de Boa Vista - RR via OpenStreetMap

2.1 Paradas de Ônibus Selecionadas

Foram utilizados dados das linhas de ônibus da cidade de Boa Vista - Roraima, encontrados na aplicação web Moovit [2], dando ênfase especial para as linhas: 308, 206, 112 e 104; com base nelas, foram definidos 33 pontos/paradas de ônibus que se encontram entre o Terminal do Centro e o ponto de ônibus ao fundo da UFRR) além de outros pontos próximos/vizinhos.

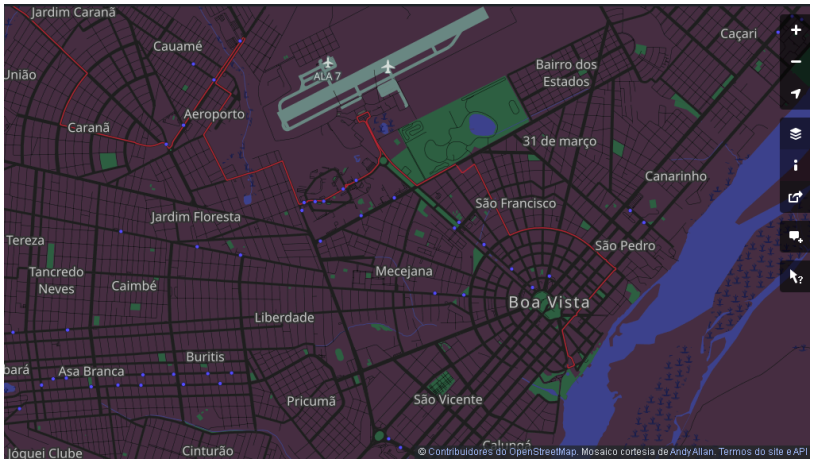

Figura 2. Representação de Pontos de Ônibus no OpenStreetMap

2.2 Captação de Coordenadas Geográficas dos Pontos

A identificação e localização geográfica precisa das paradas foi realizada através da plataforma OpenStreetMap mediante latitude e longitude. Em casos de falhas na

obtenção automática, foi utilizada a biblioteca Geopy, por meio da classe Nominatim, para recuperar as coordenadas de latitude e longitude de cada ponto, garantindo qualidade na modelagem do grafo.

2.3 Atribuição de Passageiros aos Pontos

Cada ponto de parada foi categorizado de acordo com seu tipo, representando o potencial estimado de passageiros no local. Essa categorização foi associada a faixas de valores aleatórios para simular um cenário realista:

Tabela 1. Critério de classificação dos pontos por demanda

Tipo de Parada	Descrição	Faixa de Passageiros
Tipo 1	Alta demanda (terminais, polos)	15 a 40
Tipo 2	Média demanda	7 a 15
Tipo 3	Baixa demanda	1 a 6

Valores estes que foram gerados aleatoriamente por script em linguagem de programação python para os 33 pontos de ônibus selecionados e armazenados em um arquivo .csv de arestas a ser lido pelo algoritmo Simulated Annealing para TSP feito em linguagem de programação C.

2.4 Representação em Grafo

Para estruturar o sistema de transporte como grafo, foram utilizados scripts em Python que processam os dados geográficos e de demanda, produzindo arquivos adequados para visualização e cálculo da rota:

- points_to_graph.py: lê um arquivo points.csv com os campos id, endereço, stop_type e gera:
- vertices.csv: contendo id, endereço, latitude, longitude, passageiros
- arestas.csv: contendo origem, destino, tempo_min, passageiros
- buildGraph.py: utiliza as bibliotecas Pandas, Folium e Requests para construir um grafo interativo, com base nos arquivos vertices.csv e tempo.csv. O grafo é exibido visualmente sobre o mapa de Boa Vista, permitindo análise da conectividade entre os pontos.

A estrutura final é um grafo direcionado, onde:

Vértices representam os pontos de parada com suas localizações e estimativas de demanda;

Arestas representam os tempos estimados entre pontos conectados e os passageiros carregados no trajeto.

3. Solução

Após a estruturação do grafo, o algoritmo implementado em linguagem C (tsp.c) executa cálculos de rota e obteve saídas armazenadas em arquivos .csv com base nos critérios abaixo:

- Menor custo possível;
- Rota com menor tempo total;
- Rota com maior número de passageiros transportados;
- Rota de equilíbrio, calculada por uma função de soma ponderada dos dois objetivos:

Custo Total = (peso1 x tempo_total) - (peso2 x passageiros)

O algoritmo Simulated Annealing, implementado no arquivo tsp.c, é inspirado no processo físico de recozimento e tem como destaque a capacidade de escapar de ótimos locais por meio da aceitação probabilística de soluções piores nas primeiras etapas da busca. Neste contexto, cada solução candidata corresponde a uma sequência de paradas entre um ponto de origem e um destino, incluindo um conjunto fixo de pontos intermediários.

O algoritmo Simulated Annealing (SA) inicia com uma solução aleatória que respeita as restrições de origem, destino e número máximo de intermediários. A cada iteração, uma nova solução é gerada por meio da troca de ordem entre pontos intermediários e avaliada pela função de custo. Se for melhor, é aceita; caso contrário, pode ser aceita com uma probabilidade baseada na diferença de custo e na temperatura atual, que diminui progressivamente.

O processo termina ao atingir uma temperatura mínima, retornando a melhor rota encontrada. Para a rota de equilíbrio, o algoritmo usa pesos ajustáveis para balancear tempo e número de passageiros.

Sua complexidade computacional é dada por O(I*Nt*M*A) onde I representa o número de iterações por temperatura (neste trabalho, um valor fixo), Nt corresponde ao número de ciclos de resfriamento da temperatura ao longo do processo de otimização, M refere-se ao número de vértices na rota considerada (composta pela origem, destino e um número fixo de pontos intermediários), e A é a quantidade de arestas no grafo.

4. Resultados

Rota Máximo - Passageiros	**Tempo total:**	**Passageiros totais:**
Rota de ida:	**25.69 minutos**	**61**
Rota de volta:	**26.23 minutos**	**91**

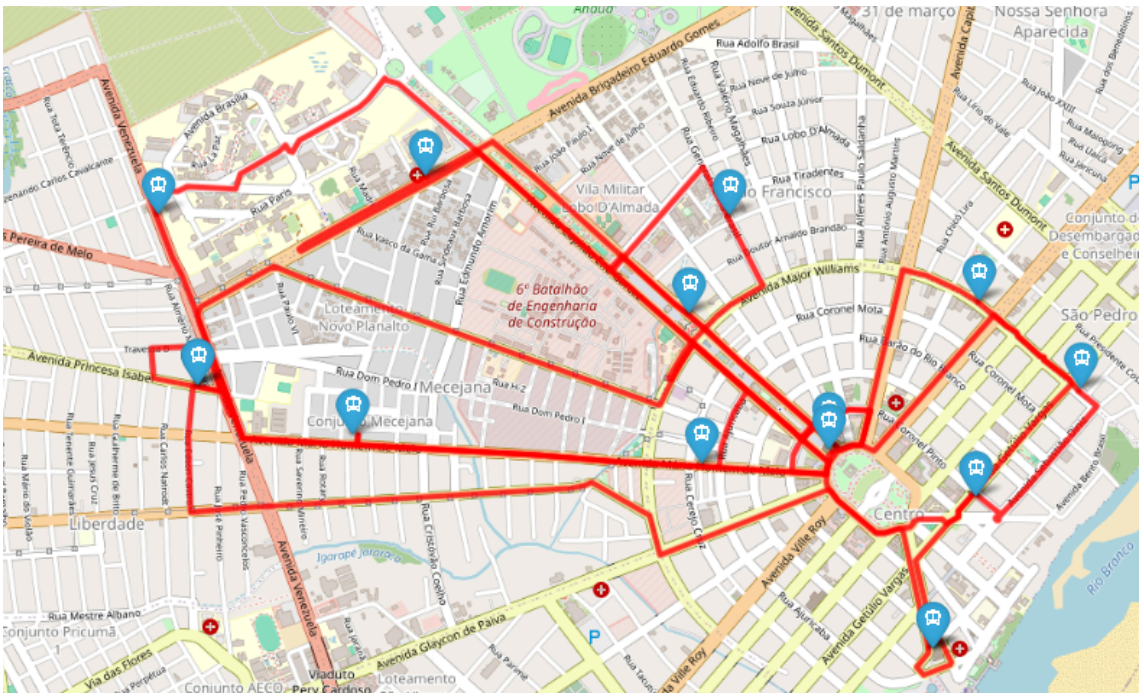

Figura 3. Rota com Maximização de Passageiros

Rota Mínimo - Tempo	Tempo total:	Passageiros totais:
Rota de ida:	15.09 minutos	27
Rota de volta:	18.79 minutos	109

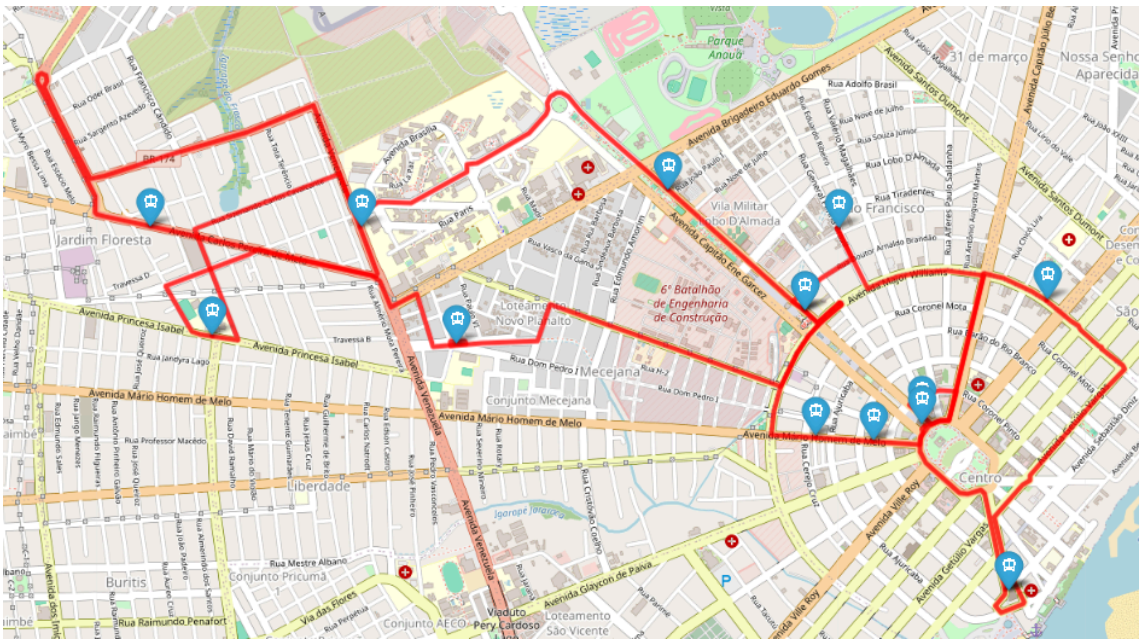

Figura 4. Rota com Minimização de Tempo

Rota Equilíbrio	Tempo total:	Passageiros totais:
Rota de ida:	12.19 minutos	23
Rota de volta:	16.97 minutos	55

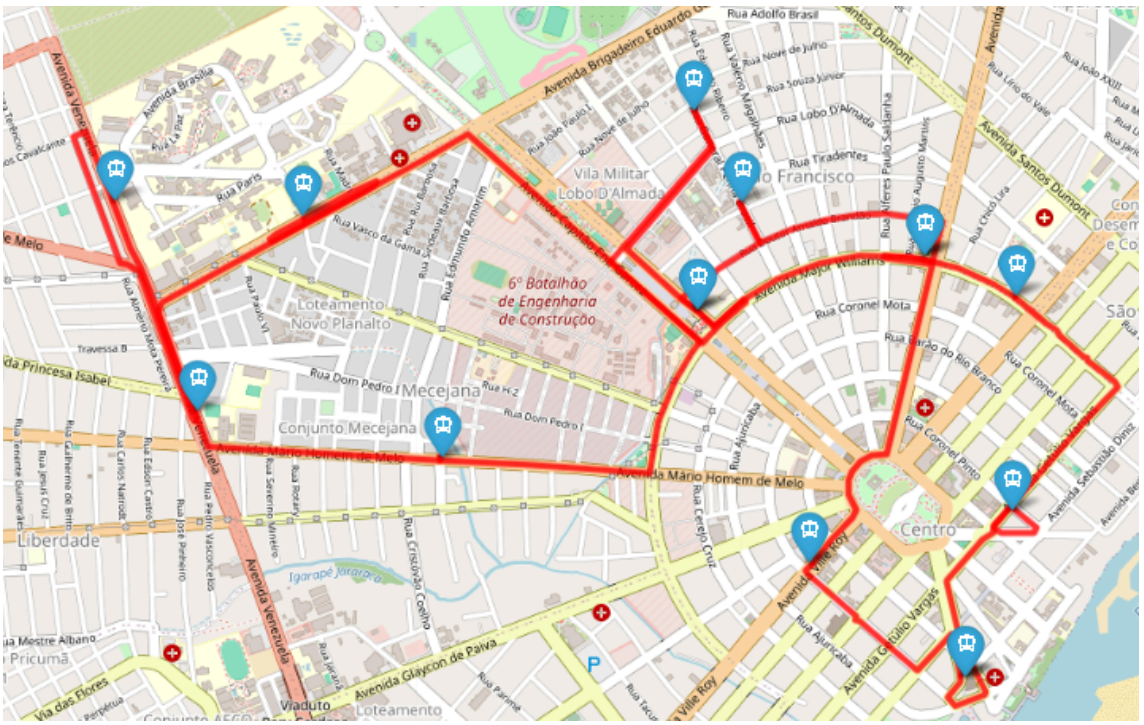

Figura 5. Rota equilibrada

5. Conclusão

Neste trabalho, foi proposto e implementado um modelo de otimização de rotas para transporte urbano baseado no Problema do Caixeiro Viajante com múltiplos objetivos, considerando simultaneamente a minimização do tempo de percurso e a maximização do número de passageiros transportados. A abordagem adotada envolveu a construção de um grafo direcionado, utilizando dados reais extraídos da malha de transporte público da cidade de Boa Vista - RR, apoiada por ferramentas geográficas e simulações baseadas em dados públicos.

Para resolver o problema, foi aplicada a meta-heurística Simulated Annealing, cujas características de exploração do espaço de soluções e capacidade de escapar de ótimos locais mostraram-se adequadas ao contexto proposto.

Além disso, este trabalho demonstrou que técnicas de otimização heurística, como o Simulated Annealing, podem ser eficazmente aplicadas para resolver problemas reais de transporte urbano com múltiplos objetivos. No caso estudado, buscou-se simultaneamente reduzir o tempo de viagem e aumentar o número de passageiros transportados. A forma de equilibrar esses dois fatores, por meio de uma função de custo com pesos ajustáveis, mostrou-se eficiente na geração de rotas variadas, conforme as prioridades estabelecidas.

A construção do grafo com base em dados reais e o uso de ferramentas como o OpenStreetMap conferiram maior realismo à simulação. Contudo, uma limitação relevante está na forma de estimar o fluxo de passageiros, que foi feita apenas com base

na frequência das linhas de ônibus. Isso pode não refletir o comportamento real da população ao longo do dia, impactando os resultados obtidos.

A aplicação prática do modelo evidenciou que o uso de heurísticas, como o Simulated Annealing, é viável para problemas de roteamento urbano com múltiplos critérios. O desafio principal foi encontrar rotas que equilibrassem o tempo de deslocamento e o volume de passageiros. A estratégia de combinar esses dois objetivos em uma única função de custo, com pesos ajustáveis, possibilitou a geração de diferentes tipos de rotas, adaptadas a variados cenários de planejamento.

Futuras melhorias incluem a incorporação de dados mais detalhados sobre o fluxo real de passageiros, além da avaliação do modelo em outras cidades para validar sua aplicabilidade geral.

6. Referências

[1] H. Tsay and J. D. Fricker, "Practical Approach for Solving School Bus Problems", 1988.

[2] Moovit. “Cidade de Boa Vista - Ônibus Horários, Rotas e Atualizações”, Moovit. Disponível em: https://moovitapp.com/index/pt-br/transporte_p%C3%BAblico-lines-Boa_Vista-4307-924570

[3] OpenStreetMap. Disponível em: https://www.openstreetmap.org/?#map=14/2.82997/-60.69869&layers=T

[4] M. V. Tenacol Coêlho e G. A. da Luz, Repositório no GitHub. Disponível em: https://github.com/GuilhermeAlmeidadaLuz/FinalProject_DCC606_Tema_CaixeiroViajante_RR_2025.git