



**Ketlyn Sara ALves Ribeiro
Luana Peixoto Borges**

Relatório do Trabalho Prático

**LAVRAS - MG
2025**

Introdução

A otimização de rotas é um desafio fundamental no transporte público. Este trabalho aborda um problema envolvendo n pontos de parada de uma linha de ônibus, modelado e resolvido com foco na minimização da maior distância entre pontos consecutivos no percurso. Para isso, foi implementado um algoritmo baseado em heurísticas e busca local, comparando os resultados obtidos com valores de referência.

Formulação

Modelagem do Grafo:

O problema abordado consiste em encontrar uma solução eficiente para uma variação do Problema do Caixeiro Viajante, mais especificamente, é a otimização de rotas para uma linha de ônibus, minimizando a maior distância entre dois pontos consecutivos no percurso. Para isso, o grafo foi modelado como sendo completo, uma estrutura não direcionada, ponderada e conexa.

O grafo $G=(V,E)$ é definido da seguinte forma:

- **Vértices (V):** Cada vértice representa uma cidade ou ponto no espaço definido por suas coordenadas geográficas.
- **Arestas (E):** Cada aresta conecta dois vértices e é ponderada com a distância entre eles.
- **Peso das arestas:** As distâncias entre os vértices são calculadas com base em métricas geométricas ou geográficas, dependendo do tipo de entrada da instância. (distância euclidiana ou distância haversine)

Um exemplo prático dessa questão que nos auxiliou na visualização da modelagem é uma linha de ônibus circular no centro da cidade de Belo Horizonte (Linha 101). Os principais elementos foram representados da seguinte forma:

Vértices: Os 5 pontos principais do trajeto, a saber:

1. Terminal Rodoviário de Belo Horizonte (TRBH)
2. Praça Sete de Setembro
3. Mercado Central
4. Praça Raul Soares
5. Parque Municipal Américo Renné Giannetti

Arestas: O caminho entre os pontos de parada.

Pesos: As distâncias em metros entre os pontos.

Neste modelo, é possível visitar todos os pontos exatamente uma vez e retornar ao ponto inicial. Além disso, o grafo é completo, pois há conexões diretas entre todos os pares de vértices.

Ida / Para	1- (TRBH)	2- Praça Sete	3- Mercado Central	4- Praça Raul Soares	5- Parque Municipal

1- (TRBH)	0	500	1200	1500	700
2- Praça Sete	500	0	800	1000	600
3- Mercado Central	1200	800	0	600	1000
4- Praça Raul Soares	1500	1000	600	0	700
5- Parque Municipal	700	600	1000	700	0

Descrição da Solução

A solução foi estruturada em:

1. Representação do grafo pela estrutura de nós
2. Matriz de adjacência gerada a partir dos nós
3. As distâncias foram calculadas levando em consideração ao tipo fornecido nas instâncias (geométricas ou geográficas)
4. Como solução inicial foi utilizado o algoritmo Nearest Neighbor (Vizinho Mais Próximo)
5. Para otimização foi usada a Busca Local (2-opt) pensando em eliminar cruzamentos que aumentam o custo do percurso

Geração da Solução Inicial

A solução inicial foi gerada utilizando o **algoritmo do vizinho mais próximo (Nearest Neighbor)**. Esse método inicia em um nó arbitrário (nó 0) e constrói a rota escolhendo, a cada passo, o nó mais próximo que ainda não foi visitado. Esse processo é repetido até que todos os nós tenham sido incluídos na rota e o percurso seja fechado retornando ao nó inicial.

A vantagem desse método é ser rápido e construir uma solução de forma eficiente, mas como a decisão local de escolha do próximo nó pode levar a escolhas ruins em longo prazo, ele pode acabar resultando em uma solução subótima.

Algoritmo de Melhoria

Para melhorar a solução inicial, utilizamos uma **otimização baseada na heurística 2-opt**. Esse método tenta reduzir o maior custo de aresta ao inverter segmentos da rota sempre que encontrar uma melhora na distância máxima entre dois pontos consecutivos. Esse processo é repetido até que não seja possível mais nenhuma melhoria.

O critério de parada do algoritmo de melhoria acontece quando nenhuma troca adicional pode reduzir a maior distância entre dois pontos consecutivos.

Resultados e Análise

Experimentos Realizados

Foram realizados experimentos computacionais utilizando 10 instâncias fornecidas na descrição do trabalho. Os seguintes critérios foram levados em consideração:

Solução inicial (SI): Valor da solução obtida pelo algoritmo do vizinho mais próximo.

Solução final (SF): Valor da solução após aplicação da heurística de melhoria 2-opt.

Desvio percentual da SF em relação à SI: Medida do ganho de qualidade obtido pela heurística de melhoria.

Desvio percentual da SF em relação à solução ótima: Avaliação da qualidade da solução final comparada ao melhor valor conhecido.

Tempo de execução: Tempo total gasto pelo algoritmo em milissegundos.

Configuração do computador utilizado: Informação detalhada sobre o sistema em que os testes foram realizados.

Os resultados experimentais foram armazenados no arquivo “results.table.txt” e o que se pode observar foi que o algoritmo de melhoria foi eficaz na redução da maior distância entre dois pontos consecutivos no percurso. Em média, observou-se um ganho de aproximadamente 67.5% na qualidade da solução final em relação à inicial. No entanto, ainda há uma diferença de ~7.7% em relação à solução ótima conhecida para cada instância.

<u>Instância</u>	SI	SF	<u>Desvio (SF vs SI)</u>	<u>Desvio (SF vs Ótimo)</u>	Tempo de <u>Execução (ms)</u>
06	1381.27	431.068	68.79%	0.0158%	965 ms
03	2874.76	1476	48.66%	0.000048%	1315 ms
01	18854.5	5099	72.96%	27.92%	139 ms
01	18854.5	5099	72.96%	27.92%	146 ms
04	2010.92	1133.29	43.64%	0.0258%	1587 ms
05	1736.09	651.494	62.47%	19.32%	1046 ms
07	1920.49	348.21	81.87%	59.00%	858 ms
08	1191.15	263.017	77.92%	-1.12%	494 ms
09	347.109	27.2029	92.16%	-47.69%	252 ms

Conclusão

Os resultados mostram que a abordagem utilizada fornece soluções de boa qualidade em tempo computacional reduzido.