

**Relatório**  
**Trabalho Prático**  
**Daniel Assis Gonçalves - 202020115**  
**Franciele de Souza Fernandes - 202020111**

## 1. Introdução

O problema em questão é uma variação do Problema do Caixeiro Viajante (TSP), onde o objetivo não é minimizar o custo total do percurso, mas sim minimizar a distância máxima entre dois pontos quaisquer no percurso de uma linha de ônibus. A solução consiste em otimizar a rota de forma que a distância máxima percorrida entre dois pontos consecutivos seja a menor possível.

## 2. Formulação

O problema pode ser representado como um grafo completo  $G = (V, E)$ , onde:

- $V$  é o conjunto de nós, representando os pontos de parada dos ônibus.
- $E$  é o conjunto de arestas, representando as conexões entre os pontos.
- Cada aresta possui um peso que corresponde à distância euclidiana entre dois pontos.

O objetivo é encontrar um ciclo Hamiltoniano  $C = v_0 v_1 \dots v_{n-1}$  que minimize a maior distância entre dois pontos consecutivos no percurso.

## 3. Descrição da Solução

### 3. 1. Representação do Problema

O grafo é representado por uma matriz de distâncias, onde cada célula contém a distância euclidiana entre dois pontos. A distância entre dois pontos  $d(i, j)$  é calculada pela fórmula de distância euclidiana:

$$d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

### 3. 2. Estrutura de Dados

A estrutura de dados utilizada para representar o grafo é uma matriz bidimensional, onde `distance_matrix[i][j]` armazena a distância entre o ponto  $i$  e o ponto  $j$ . Esta estrutura permite acesso rápido às distâncias entre os pontos, tornando o algoritmo eficiente para o problema. Além disso, para o algoritmo é utilizado um conjunto para armazenar os pontos não visitados, pois a operação de

remoção que será muito utilizada é mais eficiente utilizando esta estrutura do que em uma lista tradicional.

### 3.3. Algoritmo

O algoritmo utilizado é uma heurística do vizinho mais próximo com um fator de aleatoriedade ( $\alpha$ ). O processo de construção da solução ocorre da seguinte forma:

1. Inicia-se do ponto de origem (ponto 0).
2. Em cada iteração, escolhe-se o próximo ponto como o vizinho mais próximo do último ponto inserido no tour e mais próximo do vértice inicial, ponderando estes critérios usando um parâmetro  $\alpha \in [0, 0.5]$ . A ideia de considerar a distância para o vértice inicial deve-se ao fato de que muitas vezes na heurística tradicional o último vértice escolhido é muito distante do ponto inicial.
3. A busca continua até que todos os pontos sejam visitados.
4. O ciclo termina voltando ao ponto inicial.

Após gerar a solução, calcula-se a distância máxima entre dois pontos consecutivos do percurso, o que define a qualidade da solução.

### 3.4. Melhorias

A melhoria do algoritmo é realizada por meio da repetição da heurística, realizando várias execuções (cinco, neste caso) e selecionando a melhor solução. O valor máximo da distância entre os pontos é calculado a cada iteração, e a melhor solução é escolhida com base na menor distância máxima.

## 4. Resultados Obtidos com Análise

Durante os experimentos, foram realizadas 5 execuções para cada instância do problema, coletando os dados de melhor, pior e solução média. Além disso, é capturado o tempo de execução média. A Tabela abaixo resume os resultados obtidos:

Instância	Solução Inicial (SI)	Solução Final (SF)	Solução Ótima (SO)	$((SI-SF)/SI)*100$	$((SF-SO)/SF)*100$	Tempo Médio (s)
01	334,96	110,96	3986	66,87%	-3492,28%	0,0259
02	3367,87	1841,89	1289	45,30%	30,01%	0,1610
03	3239,20	1958,44	1476	39,53%	24,63%	0,2789

04	4136,61	1550,23	1133	62,52%	26,91%	0,4687
05	2703,45	2117,39	546	21,67%	74,21%	0,1996
06	1930,61	1560,06	431	19,19%	72,37%	0,2528
07	2493,79	1443,61	219	42,11%	84,82%	0,2016
08	2800,64	1625,46	266	41,96%	83,63%	0,1514
09	541,33	287,94	52	46,80%	81,94%	0,0760
10	2886,33	1593,37	237	44,79%	85,12%	0,3330

Configuração do computador utilizado:

Processador: 11th Gen Intel® Core™ i7-1165G7 @ 2.80GHz (Base) | 1.69 GHz (Clock Atual)

Memória RAM instalada: 32,0 GB (utilizável: 31,7 GB)

## 5. Conclusão

A heurística de vizinho mais próximo com um fator de aleatoriedade mostrou-se eficaz para este tipo de problema, apesar de não garantir a solução ótima. A repetição do processo e a escolha da melhor solução encontrada entre várias execuções permitiram resultados satisfatórios com um custo computacional razoável. Uma possível melhoria seria a utilização de técnicas de busca mais sofisticadas, como o algoritmo de busca local (local search) ou algoritmos evolutivos.

## 6. Bibliografia

CHUGANI, Vinod. *Entendendo a distância euclidiana: Da teoria à prática*. 2024. Disponível em: <https://www.datacamp.com/pt/tutorial/euclidean-distance>. Acesso em: 30 jan. 2025.

DIRENE, Alexandre I. Algoritmos de Busca Heurística (Parte 1). Universidade Federal do Paraná, Departamento de Informática. Disponível em: <https://inf.ufpr.br/alex/abh/abh1.pdf>. Acesso em: 30 jan. 2025.