

1 - Heurísticas Implementadas

1. **Nearest Neighbor:** A heurística do vizinho mais próximo começa em uma cidade e se conecta com a cidade não visitada mais próxima. Isso se repete até que todas as cidades tenham sido visitadas. Por fim, o algoritmo retorna à cidade inicial.
2. **Cheapest Insertion:** A inserção mais barata busca construir uma rota passo a passo, partindo de uma rota inicial envolvendo 2 cidades e adicionar a cada passo, a cidade k (ainda não visitada) entre a ligação (i, j) de cidades já visitadas, cujo custo de inserção $(d_{ik} + d_{kj} - d_{ij})$ seja o mais barato.
3. **Nearest Insertion:** A implementação do Nearest Insertion realizada começa com duas cidades. Em seguida, ele encontra repetidamente a cidade que ainda não está na rota que está mais próxima de qualquer cidade na rota e a coloca entre as duas cidades que permitem que a rota resultante seja a mais curta possível. O algoritmo para quando todas as cidades foram inseridas.
4. **Farthest Insertion:** Ao contrário do algoritmo anterior, a inserção mais distante começa com uma cidade e a conecta com a cidade mais distante dela. Em seguida, ele encontra repetidamente a cidade que ainda não está na rota e que está mais distante de qualquer cidade já visitada e a coloca entre as duas cidades que permitem que a rota resultante seja a mais curta possível.

2 - Resultados: Os algoritmos foram implementado em Python (3.9). E os resultados estão expressos na forma “*media +/- desvio padrão*”. O número de testes foi fixado em 50 ou um valor igual ao número de cidades (para arquivos com menos de 50 cidades).

Table 1: Custo obtido por cada algoritmo em cada arquivo de teste.

Arquivo	Nearest Neighbor	Cheapest Insertion	Nearest Insertion	Farthest Insertion
att48.tsp	13030.40 +/- 468.47	12017.31 +/- 431.54	11975.48 +/- 179.05	11624.62 +/- 100.59
kroA150.tsp	33647.29 +/- 989.52	31735.85 +/- 553.46	31225.94 +/- 357.32	30666.78 +/- 352.37
kroB100.tsp	27993.41 +/- 954.37	26687.01 +/- 712.54	27140.42 +/- 289.17	24641.29 +/- 281.92
pr107.tsp	50488.58 +/- 2828.90	53042.63 +/- 1120.43	52825.06 +/- 978.54	46878.53 +/- 1448.97
kroC100.tsp	26710.14 +/- 1489.50	26386.80 +/- 435.21	25968.68 +/- 232.88	21868.71 +/- 356.45
rat99.tsp	1475.26 +/- 56.59	1552.84 +/- 45.10	1459.65 +/- 11.41	1298.17 +/- 16.36
st70.tsp	828.44 +/- 35.17	813.45 +/- 20.70	792.46 +/- 14.36	735.59 +/- 18.58
kroB150.tsp	34520.35 +/- 1518.87	32418.06 +/- 679.67	31884.90 +/- 231.40	29255.11 +/- 217.13
kroB200.tsp	36781.66 +/- 845.64	37074.85 +/- 805.22	36388.23 +/- 162.74	34132.00 +/- 241.54
pr136.tsp	121827.91 +/- 3952.13	113553.97 +/- 2154.24	107226.16 +/- 569.19	105807.17 +/- 2495.32
pr144.tsp	65409.53 +/- 3997.38	75514.97 +/- 4624.39	72986.62 +/- 1596.30	60899.52 +/- 496.41
pr124.tsp	71742.08 +/- 1965.26	72434.41 +/- 2878.08	69651.60 +/- 1630.48	63897.54 +/- 454.10
pr76.tsp	147632.54 +/- 7558.51	131368.85 +/- 3425.34	130136.15 +/- 1015.13	117269.88 +/- 3254.07
kroD100.tsp	27428.20 +/- 936.79	26875.45 +/- 725.89	25304.96 +/- 401.72	22926.58 +/- 421.73
kroA200.tsp	37265.10 +/- 1454.23	36707.18 +/- 703.77	36302.81 +/- 571.24	33391.73 +/- 758.89
kroE100.tsp	27764.79 +/- 1332.63	27086.24 +/- 955.38	27081.34 +/- 392.77	24600.24 +/- 326.30
lin105.tsp	18818.14 +/- 1029.06	18268.97 +/- 431.35	18478.54 +/- 263.32	15636.67 +/- 239.96
rat195.tsp	2774.36 +/- 75.37	2825.42 +/- 71.61	2921.57 +/- 36.21	2797.09 +/- 33.46
berlin52.tsp	9315.93 +/- 433.02	9381.53 +/- 342.91	9071.50 +/- 143.53	8051.28 +/- 248.90
kroA100.tsp	26911.90 +/- 869.76	27055.13 +/- 553.64	25799.03 +/- 320.98	24343.88 +/- 589.10
att48.tsp	12925.62 +/- 392.91	11982.79 +/- 443.30	11961.62 +/- 215.91	11560.96 +/- 160.72
pr152.tsp	87030.31 +/- 3208.47	91697.86 +/- 699.42	88971.96 +/- 1496.21	78513.31 +/- 630.09

Table 2: Tempo de execução (*ms*) de cada algoritmo em cada arquivo de teste. Processador i7 7th Gen., 8 Gb de RAM, sistema Linux.

Arquivo	Nearest Neighbor	Cheapest Insertion	Nearest Insertion	Farthest Insertion
att48.tsp	0.3 +/- 0.1	13.7 +/- 0.8	4.9 +/- 0.6	5.0 +/- 0.7
kroA150.tsp	0.9 +/- 0.2	411.6 +/- 42.1	53.5 +/- 5.5	52.7 +/- 5.6
kroB100.tsp	0.6 +/- 0.1	135.0 +/- 18.1	24.3 +/- 3.5	24.8 +/- 3.5
pr107.tsp	0.5 +/- 0.1	154.0 +/- 20.3	25.1 +/- 3.5	25.4 +/- 3.7
kroC100.tsp	0.6 +/- 0.1	135.4 +/- 17.4	24.9 +/- 3.4	24.8 +/- 3.4
rat99.tsp	0.5 +/- 0.1	129.0 +/- 17.6	22.6 +/- 3.6	22.4 +/- 3.3
st70.tsp	0.4 +/- 0.1	39.7 +/- 1.0	10.1 +/- 0.8	10.1 +/- 0.8
kroB150.tsp	1.0 +/- 0.2	457.7 +/- 57.7	58.8 +/- 8.1	58.5 +/- 8.2
kroB200.tsp	1.3 +/- 0.4	1085.6 +/- 123.5	108.3 +/- 13.9	108.9 +/- 15.8
pr136.tsp	0.8 +/- 0.2	343.5 +/- 42.2	44.8 +/- 5.9	45.6 +/- 6.0
pr144.tsp	0.8 +/- 0.1	425.8 +/- 53.0	51.5 +/- 7.1	51.0 +/- 6.4
pr124.tsp	0.7 +/- 0.2	271.0 +/- 31.0	37.2 +/- 4.7	38.2 +/- 5.1
pr76.tsp	0.4 +/- 0.1	57.3 +/- 7.4	13.1 +/- 1.9	13.0 +/- 1.8
kroD100.tsp	0.6 +/- 0.2	142.0 +/- 16.8	26.3 +/- 3.2	26.4 +/- 3.4
kroA200.tsp	1.4 +/- 0.3	1181.2 +/- 133.8	123.8 +/- 23.0	121.0 +/- 24.5
kroE100.tsp	0.6 +/- 0.2	144.1 +/- 14.8	26.3 +/- 3.0	26.8 +/- 3.2
lin105.tsp	0.6 +/- 0.2	163.3 +/- 18.4	27.2 +/- 3.2	26.7 +/- 3.2
rat195.tsp	1.3 +/- 0.3	1061.5 +/- 101.8	95.6 +/- 10.5	96.5 +/- 10.6
berlin52.tsp	0.2 +/- 0.1	16.4 +/- 1.2	5.5 +/- 0.7	5.6 +/- 0.8
kroA100.tsp	0.7 +/- 0.2	150.7 +/- 4.0	27.9 +/- 1.5	27.6 +/- 1.6
att48.tsp	0.2 +/- 0.0	13.3 +/- 1.2	5.0 +/- 0.9	5.0 +/- 0.7
pr152.tsp	0.9 +/- 0.2	505.2 +/- 52.3	58.5 +/- 6.6	58.4 +/- 7.1