TP - 1

José Joaquim de Andrade Neto

November 22, 2024

1 Introdução

Nesse trabalho, resolvi as instâncias fornecidas no moodle utilizando a heurística construtiva *Nearest Neighbor* (NN). De acordo com o Algoritmo 1, seja *R* um *tour* inicialmente composto pelo nó inicial 0. O laço que compreende linhas 4-15 é responsável por encontrar o nó *k* mais próximo do nó corrente ainda não visitado e atualizá-lo para ser o próximo nó corrente. O laço descrito anteriormente é executado uma vez para cada nó. Depois que o mesmo é finalizado, ligamos o último nó visitado ao vértice inicial 0, satisfazendo a definição de um *tour* de TSP.

Implementei a heurística construtiva NN utilizando a linguagem Rust. A Tabela 1 apresenta o resultado para cada instância disponibilizadda no moodle:

```
Algorithm 1: CH
```

```
input: N = \{1, 2, ..., n\}, f: R \mapsto \mathbb{R}, d: N \times N \mapsto \mathbb{R}
   output: R, f(R)
 1 R \leftarrow \langle 0 \rangle
 2 counter ← 0
 3 atual ← 0
 4 while counter < |N| do
         next \leftarrow -1
         menor \leftarrow 1000000007
        foreach j \in N do
              if d(atual, j) < \text{menor} \land j \notin R then
                   next \leftarrow j
                   menor \leftarrow f(atual, j)
10
11
              end
         end
12
         R \leftarrow R \cup \text{next}
13
         counter \leftarrow counter + 1
14
15 end
16 return R, f(R)
```

Table 1: Instância e tamanho da rota encontrada

Instância	Distância da rota
att48.tsp	39964.12
kroA100.tsp	25345.75
kroA150.tsp	32593.13
kroA200.tsp	34483.38
kroB100.tsp	25990.58
kroB150.tsp	32446.42
kroB200.tsp	34748.27
kroC100.tsp	24010.08
kroD100.tsp	23979.38
kroE100.tsp	26534.79
lin105.tsp	17173.36
pr107.tsp	38628.15
pr124.tsp	67236.48
pr136.tsp	112986.69
pr144.tsp	59403.92
pr152.tsp	69964.63
pr76.tsp	137192.63
rat195.tsp	2637.99
rat99.tsp	1467.93
st70.tsp	717.53