

PCS3438 - Inteligência Artificial

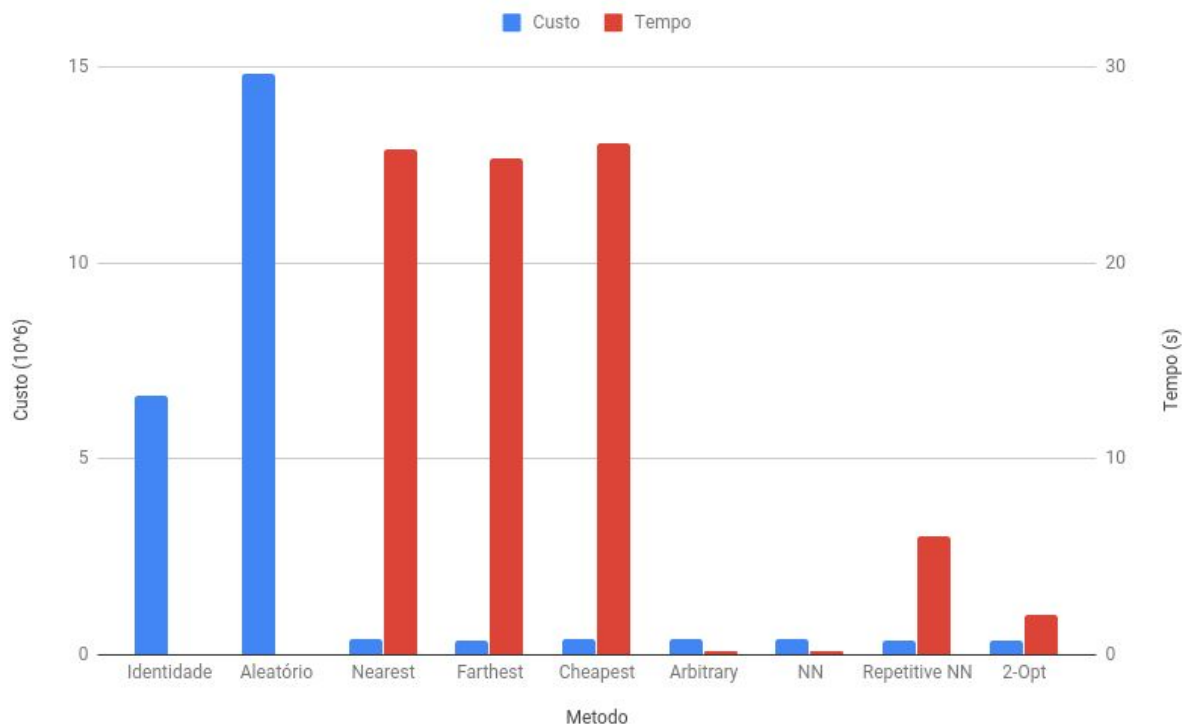
Exercício Programa I

Daniel Nery Silva de Oliveira
9349051

Para a resolução deste Exercício Programa, foi utilizada a biblioteca TSP disponível no CRAN, que pode ser instalado com:

```
install.packages('TSP')
```

O método escolhido para ser utilizado foi o "Arbitrary Insertion" para escolha do caminho inicial e "2-Opt" para sua otimização, esse método foi escolhido entre os disponíveis na biblioteca por ter se mostrado o mais eficiente dentre os disponibilizados pela biblioteca, como pode ser visto no gráfico a seguir, gerado a partir de testes num dataset com 1889 cidades:



Esse algoritmo consiste em iniciar o caminho por uma cidade aleatória e a cada passo escolher uma outra cidade aleatória k que ainda não esteja no caminho e inseri-la entre duas outras cidades de modo que o crescimento no custo $d(i,k) + d(k,j) - d(i,j)$ seja minimizado. O algoritmo para quando todas as cidades estiverem no caminho.

Listagem do Código

A função criada para resolver segue abaixo:

```
library('TSP')

## Função principal
EP1_9349051 <- function(entrada, saida) {
  # Lê o arquivo de entrada
  dados <- read.table(
    entrada,
    header = TRUE,
    sep = ',')

  # Transforma os dados lidos numa matriz de
  # adjacências
  dados <- as.dist(dados)

  # Transforma a matriz num objeto da classe
  # TSP, para ser resolvido
  tsp <- TSP(dados)

  # Computa o momento inicial de processamento
  initial_time <- Sys.time()

  # Resolve o TSP, usando o método da biblioteca
  # arbitrary insertion
  solved_tsp <- solve_TSP(tsp, method="arbitrary_insertion")

  # Extrai as informações necessárias da solução
  # Computa o tempo total de solução, a rota
  # começando pela cidade 1 e custo total
```

```

execution_time <- as.numeric(Sys.time() - initial_time)
tour <- cut_tour(solved_tsp, 1, exclude_cut=FALSE)
total_cost <- tour_length(solved_tsp)

# Imprime dados relevantes na tela
cat(sprintf("Custo: %f\n", total_cost))
cat(sprintf("Tempo de Execução: %fs\n", execution_time))
cat("Rota:")
for (city in tour) {
  cat(sprintf(" %d", city))
}
cat("\n")

# Escreve os dados pedidos no arquivo de saída
results <- matrix(c(total_cost, execution_time, tour), nrow=1)
write.table(
  results,
  file=saida,
  sep=',',
  row.names=FALSE,
  col.names=FALSE)
}

```

Referências

<https://cran.r-project.org/web/packages/TSP/vignettes/TSP.pdf>