**Trabalho Prático 2: Problema do Caixeiro Viajante Assimétrico**

**Leandro Lázaro (3513), Mateus Pinto (3489), Vinícius Júlio (3495)**

Ciência da Computação – Universidade Federal de Viçosa – Campus Florestal (UFV- Caf) – Florestal – MG – Brasil

{leandro.lazaro, mateus.p.silva, vinicius.julio}@ufv.br

***Resumo.*** *Implementamos o Problema do Caixeiro Viajante Assimétrico com o intuito de testá-lo com diferentes tamanhos de entrada para avaliar o seu desempenho baseado no seu tempo de execução.*

**1. Introdução**

Neste trabalho implementaremos o Problema do Caixeiro Viajante Assimétrico, que envolve o cálculo do caminho de menor distância entre N cidades à serem visitadas pelo Caixeiro por meio de permutações, com o intuito de analisar o desempenho do algoritmo com diferentes números de cidades a serem visitadas baseado no seu tempo de execução, que será medido usando comandos do sistema operacional. Serão disponíveis formas de entrada manual e automática (por leitura de arquivo) para inserir as cidades e os custos de seus caminhos, além da cidade inicial.

**2. Algoritmo de Permutação**

O algoritmo de permutação foi criado pelo usuário do GitHub *marcoscastro* (Link do projeto <https://gist.github.com/marcoscastro/60f8f82298212e267021>) calcula todas as possibilidades de caminhos e seus custos usando a função *setCustomCaminho* e compara as distâncias armazenando a menor na variável *melhorCaminho*, ou seja, também encontra o menor caminho.

**3. Funções**

**3.1 Função troca**

A função *troca* tem como objetivo ser auxiliar do algoritmo de permutação. É utilizada quando é necessária a inversão de duas posições de um vetor.

**3.2. Função setCidadeinicial**

A função *setCidadeinicial* define qual a cidade será o ponto de partida do Caixeiro Viajante baseado no resto da soma dos números de matrícula dos integrantes do grupo dividido pelo número de cidades.

**3.3. Função setCustoCaminho**

A função *setCustoCaminho* calcula o custo do caminho a ser percorrido que é formado por uma sequência de cidades.

**3.4. Função getTracejado**

Função *getTracejado* imprime um tracejado para melhor organização da tela do programa.

**4. Programa Principal (main)**

O Programa Principal é a implementação do algoritmo de permutação do Problema do Caixeiro Viajante Assimétrico. O algoritmo pode ser utilizado em dois modos: interativo, onde o usuário informa as manualmente os dados, e o modo por arquivo, onde os parâmetros serão passados por meio da leitura de um arquivo contendo as cidades, suas distâncias e os números de matrícula. Ambos os modos realizam as mesmas operações. O tempo é calculado pela função *clock.* Assim, as informações são inseridas, a cidade inicial definida, os caminhos e suas distâncias são calculadas, e por fim, após as comparações, é definido qual o melhor caminho a ser percorrido, e é mostrado na tela cada um desses dados.

**5. Resultados do Tempo de Execução**

\*/tabela e grafíco/\*

**6. Considerações Finais**

Caso fóssemos contratados por uma transportadora para desenvolver um sistema que encontrasse a menor rota a ser percorrida por seus caminhões, saindo e chegando do galpão de estoque e percorrendo uma única vez um conjunto de N localidades, não utilizaríamos a solução desenvolvida neste trabalho, uma vez que sua complexidade é exponencial e existem algoritmos mais eficazes para tal tarefa.

A implementação do problema do Caixeiro Viajante Assimétrico nos permitiu estudar sobre o desempenho de um algoritmo de acordo com o seu tempo de execução. Ao analisar o tempo gasto pelo algoritmo de permuta na sua execução, foi percebido um aumento exponencial do mesmo enquanto o número de cidades testadas aumentava linearmente, o que configura o problema como intratável.