

EACH - USP

Laboratório de Otimização Combinatória  
Relatório do EP 2

Nome: Matheus Taipina Benini

Nome: Larissa Marques

# Enunciado do Problema

O problema a ser modelado é conhecido como “Problema do caixeiro viajante”. Pode ser enunciado da seguinte forma: “Dado uma lista de cidades e a distância entre cada par de cidades, qual é a menor rota possível que visita cada cidade e retorna para a cidade original?”

## Modelo

Estaremos implementando um modelo para o problema do caixeiro viajante simétrico, ou seja, o custo de uma cidade  $x1$  para  $x2$  é o mesmo da cidade  $x2$  para  $x1$ .

O problema será modelado como um grafo não-direcionado, onde seus vértices simbolizam as cidades e suas arestas o trajeto entre elas. Para cada aresta, um valor inteiro é atribuído, definindo o custo deste trajeto.

## Constantes

$C_{ij}$  = Custo do trajeto entre a cidade  $i$  e  $j$ .  $i = (1, 2, \dots, nV)$  e  $j = (1, 2, \dots, nV)$

$V_e$  = Vértices do grafo, indexado por  $e = (1, 2, 3, \dots, nV)$

$nV$  = número de vértices no grafo,  $nV \in \mathbb{N}^*$

## Variáveis

$X_{ij}$  = Informa se a aresta entre a cidade  $i$  e  $j$  foi utilizada na solução (1 = utilizado, 0 = não utilizado).

## Restrições

1.  $\sum_j (X_{ij}) = 2, \forall i \in (1, 2, \dots, nV)$
2.  $\sum_e (X_e) \geq 2, \forall e \in A: e = (u, v), u \in S, v \notin S, \forall S \subset V$

A restrição 2 foi implementada utilizando uma *lazy constraint callback*, que utiliza o algoritmo de “fluxo-máximo e corte mínimo” para determinar o conjunto  $S$  a ser utilizado.

## Função objetivo

$$\min \sum_i (\sum_j (X_{ij} * C_{ij}))$$

## Conjunto de testes

Os conjuntos de teste foram gerados aleatoriamente, com os seguintes parâmetros:

- Grafos contendo 50 vértices
- Custo entre os vértices variando entre 1 e 50
- Foram gerados 100 grafos.

Os conjuntos de testes estão presentes na pasta “samples”, em formato .txt, enviados junto ao relatório. Cada instância é representada como uma matriz que informa o custo entre o vértice  $i$  (linha) e  $j$  (coluna).

## Resultados

Os resultados estão presentes no arquivo results-samples-50.txt.

Cada instância é iniciada pela linha “Executando Sample-50-x.txt”, sendo  $x$  o número da instância em questão.

Após isso, é apresentado o log do CPLEX durante a solução do problema. Por último, é informado o “menor custo para o trajeto” e a sequência de vértices da solução encontrada, no formato “ $x \rightarrow y$ ”, onde  $x$  e  $y$  são os números dos vértices correspondentes, iniciando com zero. A sequência de vértices sempre começa e termina no vértice zero (0).

## Comentários

A resolução de instâncias ocorreu de forma bem rápida, apesar da quantidade de vértices presentes no problema. Nota-se que a maioria dos custos encontrados para as instâncias é bem baixo, se comparado com a quantidade de vértices (50), e o range de custos possíveis (1-50). Ao avaliar algumas soluções individualmente, foi percebido que, por razões triviais, o *solver* estava se utilizando de algumas arestas que acabavam tendo seus custos gerados bem baixos, em torno de 1 ou 2.