MO824A/MC859A – Tópicos em Otimização Combinatória
Primeiro semestre de 2022

Atividade 2
Entrega: 15 de abril até 23:59

Prof. Fábio Luiz Usberti (fusberti@ic.unicamp.br)
Prof. Celso Cavellucci (celsocv@ic.unicamp.br)

---

1 Objetivo

O objetivo desta atividade consiste na modelagem de um problema de programação linear inteira utilizando *lazy constraints* e a solução desse modelo utilizando o software Gurobi.

A atividade deve ser realizada em equipes de 2 a 3 alunos. Os docentes vão sortear as equipes aleatoriamente. As equipes com 2 alunos ganharão um bônus na nota em virtude do número menor de alunos.

2 Descrição do Problema

O problema do caixeiro viajante (*travelling salesman problem* – TSP) pode ser descrito da seguinte forma. Seja um grafo não-orientado completo $G(V, E)$, onde V é o conjunto dos vértices e E é o conjunto das arestas. Em cada aresta $e \in E$ há um custo $c_e : E \to \mathbb{R}^+$. O objetivo do problema consiste em encontrar um ciclo Hamiltoniano (que visita todos os vértices do grafo) de custo mínimo.

Um modelo de programação linear inteira para o TSP é fornecido a seguir:

$$MIN \quad \sum_{e \in E} c_e x_e \tag{1}$$

$$s.t.$$

$$\sum_{e \in \delta(i)} x_e = 2 \qquad \forall i \in V \tag{2}$$

$$\sum_{e \in E(S)} x_e \leqslant |S| - 1 \qquad \forall S \subset V \tag{3}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E \tag{4}$$

Onde:

- x_e é uma variável de decisão binária associada à presença ($x_e = 1$) ou não ($x_e = 0$) da aresta e na solução;
- $\delta(i)$ é o conjunto de arestas que incidem no vértice i;
- $S \subset V$ é um subconjunto próprio de vértices;
- $E(S)$ é o conjunto das arestas cujos dois vértices terminais estão em S.

A função objetivo (1) minimiza o custo da solução, composto pela soma dos custos de todas as arestas presentes na solução. O conjunto de restrições (2) diz que cada vértice deve possuir duas arestas incidentes na solução. Finalmente, o conjunto de restrições (3) visa a eliminação de subciclos ilegais, ou seja, a solução deve possuir um único ciclo que visita todos os vértices. O modo como as restrições (3) eliminam subciclos ilegais está em impedir que em qualquer subconjunto próprio de vértices existam mais do que $|S| - 1$ arestas, o que seria suficiente para formar um ciclo.

Cabe notar que a quantidade de restrições (3) é exponencial em função da quantidade de vértices do grafo. Sendo assim, torna-se impraticável enumerar todas as restrições (3) para instâncias grandes. Uma forma de contornar essa questão está em utilizar o conceito de *lazy constraints*, onde as restrições são consideradas apenas sob demanda. Dessa forma, primeiro é passado ao resolvedor somente o conjunto de restrições (2). O resolvedor trata o modelo incompleto e ao encontrar uma solução inteira, essa solução é passada ao usuário por meio de uma função de *callback*. O usuário verifica se a solução é composta por um ou mais ciclos. Se possuir somente um ciclo, a solução é válida e o algoritmo declara otimalidade. Caso contrário, o usuário insere uma ou mais restrições (3) violadas pela solução e o resolvedor re-otimiza considerando as novas restrições.

O modelo TSP com *lazy constraints* encontra-se já implementado como um exemplo de código do Gurobi (vide referência abaixo).

3 Requisitos da atividade

Nesta atividade você deverá modelar e resolver um problema derivado do TSP, denominado problema dos caixeiros viajantes k-semelhantes (k-similar travelling salesmen problem, kSSTP). Para isso, você pode utilizar como base o código de exemplo do Gurobi para a solução do TSP.

3.1 Descrição do problema

O kSTSP pode ser descrito da seguinte forma. Seja um grafo não-orientado completo $G(V, E)$, onde V é o conjunto dos vértices e E é o conjunto das arestas. Em cada aresta $e \in E$ há dois custos $c_e^1, c_e^2 : E \to \mathbb{R}^+$. O objetivo do problema consiste em encontrar dois ciclos Hamiltonianos com custo total mínimo, tal que pelo menos k arestas do grafo sejam visitadas por ambos os ciclos. O parâmetro k define a similaridade entre os ciclos, de modo que $k = |V|$ implica que uma solução factível corresponda a dois ciclos Hamiltonianos disjuntos nas arestas, enquanto $k = 0$ resulta em uma solução factível ser qualquer par de ciclos Hamiltonianos.

Faça a formulação em programação linear inteira do STSP.

3.2 Geração de instâncias

Você deve gerar 12 instâncias, combinando quatro conjuntos de vértices ($|V| = \{100, 150, 200, 250\}$) e três valores de similaridade ($k = \{0, \frac{|V|}{2}, |V|\}$).

Será fornecido um arquivo texto com coordenadas inteiras (x_i^1, y_i^1) e (x_i^2, y_i^2) para $i = \{1, \ldots, 250\}$. Os dados no arquivo estarão organizados conforme abaixo:

```
<coordenada x^1_1> <coordenada y^1_1> <coordenada x^2_1> <coordenada y^2_1>
<coordenada x^1_2> <coordenada y^1_2> <coordenada x^2_2> <coordenada y^2_2>
...
<coordenada x^1_250> <coordenada y^1_250> <coordenada x^2_250> <coordenada y^2_250>
```

A partir dessas coordenadas você deverá gerar os custos c_e^1 e c_e^2 de uma aresta $e = (i, j)$ calculando o teto da distância Euclidiana dos pares de coordenadas, conforme as equações abaixo:

$$c_e^1 = \left\lceil \sqrt{(x_i^1 - x_j^1)^2 + (y_i^1 - y_j^1)^2} \right\rceil$$
$$c_e^2 = \left\lceil \sqrt{(x_i^2 - x_j^2)^2 + (y_i^2 - y_j^2)^2} \right\rceil$$

Para cada instância com conjunto de vértices V você deverá utilizar as primeiras $|V|$ linhas do arquivo de coordenadas para gerar os custos c_e^1 e c_e^2.

3.3 Execução de experimentos

Resolva as 12 instâncias do STSP, anotando o tempo de execução, o custo da solução e o *gap* de otimalidade. Você deve limitar o tempo de execução em 30 minutos por instância.

3.4 Entrega

A atividade exige a entrega do código-fonte e de um relatório (aproximadamente 3 a 5 páginas) contendo:

- Modelo matemático: apresente o modelo para o STSP e descreva o significado das variáveis de decisão e das restrições.
- Resultados: tabela de resultados contendo, para cada instância: o custo da solução, *gap* de otimalidade e tempo de execução.
- Análise: avalie os resultados quanto aos custos obtidos e os tempos computacionais.

3.5 Critério de avaliação

A correção do relatório será pautada pela qualidade dos seguintes quesitos:

- Texto: qualidade da redação, clareza, síntese, estrutura, organização.
- Modelo: descrição do modelo, variáveis, parâmetros, restrições, domínios.
- Experimentos: descrição dos experimentos, configuração da máquina, geração de instâncias, linguagem de programação.
- Análise: análise dos resultados quanto ao valor das soluções e tempos computacionais.

4 Referência

1. Exemplo de código Gurobi para resolver o TSP:
   https://www.gurobi.com/documentation/9.0/examples/tsp_java.html#subsubsection:Tsp.java