MO824A/MC859A – Tópicos em Otimização Combinatória
Segundo semestre de 2020

Atividade 2
Entrega: 9 de outubro 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 dois caixeiros viajantes (2TSP). Para isso, você pode utilizar como base o código de exemplo do Gurobi para a solução do TSP.

3.1 Formulação do problema

O objetivo do 2TSP consiste em encontrar dois ciclos Hamiltonianos com custo total mínimo disjuntos nas arestas. Faça a formulação em programação linear inteira do 2TSP.

3.2 Geração de instâncias

Gere 5 instâncias aleatórias variando a quantidade de vértices $|V| = \{20, 40, 60, 80, 100\}$. Para calcular os custos de cada aresta do grafo completo, posicione cada vértice em coordenadas aleatórias no plano, com distribuição uniforme, no intervalo $[0, 1]$, e depois calcule a distância Euclidiana de cada par de vértices. O custo de uma aresta será igual à distância Euclidiana de seus vértices terminais.

3.3 Execução de experimentos

Resolva as 5 instâncias para os problemas TSP e 2TSP no solver Gurobi, anotando o tempo de execução e o custo da solução. Você deve limitar o tempo de execução em 30 minutos.

3.4 Entrega

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

- Modelo matemático: apresente o modelo para o 2TSP e descreva o significado das variáveis de decisão e das restrições.

- Resultados: tabela de resultados contendo, para cada instância: o custo da solução do TSP (multiplique o custo por dois), tempo de execução do TSP, o custo da solução do 2TSP e tempo de execução do 2TSP.

- Análise: avalie os resultados quanto aos custos obtidos e os tempos computacionais.

4 Referências

1. Exemplo de código Gurobi para resolver o TSP:

   https://www.gurobi.com/documentation/9.0/examples/tsp_java.html#subsubsection:Tsp.java