Atividade 3 - Técnica de Relaxação Lagrangiana para o kSTSP

Lucas Guesser Targino da Silva - RA: 203534
Renan Fernando Franco da Silva - RA: 223989

29 de abril de 2022

1 Enunciado do Problema

Sejam:

1. $G = \langle V, E \rangle$: um grafo não-orientado completo:
   (a) V: conjunto de vértices;
   (b) E: conjunto das arestas;
2. $c^k : E \to \mathbb{R}_+, \ \forall k \in \{1, 2\}$, duas funções custo nos vértices;
   (a) dada uma aresta e, escrevemos $c^k(e) = c_e^k$;
3. σ: parâmetro de similaridade de ciclos;

Objetivo: encontrar dois ciclos Hamiltonianos com custo total mínimo, tal que pelo menos σ arestas do grafo sejam visitadas por ambos os ciclos.

2 Modelo Matemático

Nessa seção, será apresentada a formulação do problema utilizando Programação Linear Inteira. Esse não foi resolvido diretamente (j[a que não é o intuito da atividade), mas foi utilizado para derivar uma Relaxação Lagrangiana (Seção 3), que foi modelo de fato implementado e resolvido.

2.1 Variáveis de Decisão

- x_e^k: presença da aresta e no ciclo k;
- z_e: presença de duplicação da aresta e;

Todas as variáveis de "presença" são decisões binárias com a seguinte interpretação de valores:

0 : ausente

1 : presente

2.2 Problema de Otimização

Minimizar:

$$\sum_{k \in \{1,2\}} \sum_{e \in E} c_e^k \, x_e^k \tag{1}$$

Sujeito a:

$$\sum_{e \in \delta(v)} x_e^k = 2 \qquad \forall v \in V, \forall k \in \{1,2\} \tag{2}$$

$$\sum_{e \in E(S)} x_e^k \leqslant |S| - 1 \qquad \forall S \subseteq V, S \neq V, S \neq \emptyset, \forall k \in \{1,2\} \tag{3}$$

$$x_e^k \geqslant z_e \qquad \forall e \in E, \forall k \in \{1,2\} \tag{4}$$

$$\sum_{e \in E} z_e \geqslant \sigma \tag{5}$$

$$x_e^k, z_e \in \{0,1\} \qquad \forall e \in E, \forall k \in \{1,2\} \tag{6}$$

2.3 Explicação das Restrições

- A função objetivo (1) é soma do custo de todas as arestas selecionadas em todos os ciclos.
- A restrição (2) garante que a quantidade de arestas incidentes em todos os vértices seja 2. Essa condição faz com que todos os vértices tenham que ser visitados (duas arestas pois uma é a de "entrada" e a outra a de "saída").
- A restrição (3) garante que não existam subciclos nos ciclos. Nessa restrição, S é um subconjunto próprio e não-vazio dos vértices do problema. A expressão $E(S)$ é o conjunto das arestas cujos vértices (ambos) estão em S.
- A restrição (4) garante que, se uma aresta foi escolhida para ser duplicada, então essa aresta aparecerá nos dois ciclos.
- A restrição (5) garante que pelo menos σ arestas serão escolhidas para serem duplicadas.
- A restrição (6) garante que todas as variáveis são decisões binárias, ou seja, assumem apenas um de dois possíveis valores: 0 e 1.

2.4 Tamanho das Restrições

- Restrição (2): uma para cada vértice e para cada ciclo. Total: $2 \cdot |V|$;
- Restrição (3): uma para $S \in \mathcal{P}(V), S \neq V, S \neq \emptyset$ e para cada ciclo. Total: $2 \cdot \left(2^{|V|} - 2\right)$;
- Restrições (4): uma para cada aresta e para cada ciclo. Total: $2 \cdot |E| = 2 \cdot \dfrac{|V|^2 - |V|}{2}$ (já que o grafo é completo);
- Restrições 5: apenas uma. Total: 1;

Assim, o número total de restrições é:

$$T_r = 2 \cdot |V| + 2 \cdot \left(2^{|V|} - 2\right) + 2 \cdot \frac{|V|^2 - |V|}{2} + 1 \tag{7}$$

Assim:

$$T_r \in \mathcal{O}\left(2^{|V|}\right) \tag{8}$$

Note que há um número exponencial de restrições.

3 Relaxação Lagrangiana

3.1 Escolha da Restrição a ser Relaxada

A técnica de Relaxação Lagrangiana consiste em escolher uma ou mais restrições para relaxar. Tal escolha visa remover as restrições que fazem do problem "difícil de ser resolvido" [1].

3.1.1 Análise das Restrições

- Restrição 2: não parece ser uma restrição difícil;
- Restrição 3: essa restrição é de fato difícil. Entretanto, há um número exponencial delas, de forma que é inviável relaxá-la. Além disso, há técnicas que lidam bem com ela[1];
- Restrição 4: no trabalho anterior, notou-se que o fator de similaridade causa bastante impacto no tempo computacional Figura 1. Isso significa que ele deixa o problema difícil, sendo então um bom candidato à relaxação;
- Restrição 5: essa restrição tem efeitos bem parecidos com a Restrição 4. Ela representa, entretanto, apenas uma equação de restrição, não sendo assim interessante para a relaxação;
- Restrição 6: já é considerada a relaxação dessa restrição pelo método;

Portanto, a restrição escolhida para ser relaxada é a Restrição 4.

3.2 Modelo Matemático da Relaxação Lagrangiana

Minimizar:

$$\sum_{k \in \{1,2\}} \sum_{e \in E} c_e^k \, x_e^k + \lambda_e^k \left(z_e - x_e^k\right) \tag{9}$$

Sujeito às restrições 2, 3, e 5 do problema original (Subseção 2.2), e a restrições de domínio:

$$0 \leqslant x_e^k, z_e \leqslant 1 \qquad \forall e \in E, \forall k \in \{1,2\} \tag{10}$$

$$\lambda_e^k \geqslant 0 \qquad \forall e \in E, \forall k \in \{1,2\} \tag{11}$$

[1] *Lazy constraints* por exemplo, abordado no trabalho anterior.

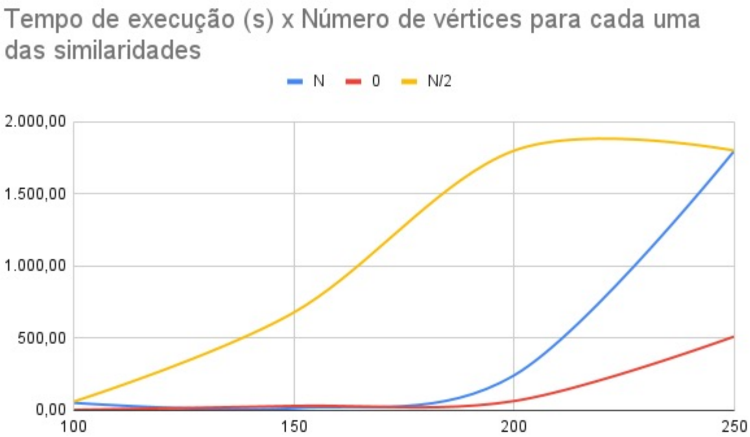

Figura 1: tempo de execução dos experimentos do trabalho anterior.

3.3 Método do Subgradiente

TO DO

Descrever como fazemos para atualizar λ_e^k.

3.4 Heurística Lagrangiana

TO DO

Descrever como transformamos uma solução infactível numa solução factível.

4 Algoritmo de Solução do Problema

TO DO

Descrever o algoritmo completo, em pseudocódigo.

5 Experimento Computacional

5.1 Configuração da Máquina

O problema foi executado num ideapad S145 81S90005BR: Lenovo IdeaPad S145 Notebook Intel Core i5-8265U (6MB Cache, 1.6GHz, 8 cores), 8GB DDR4-SDRAM, 460 GB SSD, Intel UHD Graphics 620.

O sistema operacional foi o Fedora 35 executando gcc (GCC) 11.2.1 20220127 (Red Hat 11.2.1-9), Concorde [2] e o solver QSOpt [3].

Como linguagem de programação, utilizamos C++ [4].

5.2 Dados do Problema

Os dados do problema foram fornecidos em um arquivo contendo 4 colunas e 250 linhas. A interpretação dos dados é a seguinte: cada linha representa um vértice e cada par de coluna as coordenadas desse vértice. A razão para um vértice ter duas posições diferentes é simplesmente para que as distâncias entre eles tenham valores diferentes no primeiro e no segundo ciclo.

O modelo na verdade precisa apenas de pesos. Construímos a primeira função de custo como a distância euclidiana entre os pontos das colunas 1 e 2. Da mesma forma, utilizamos distância euclidiana entre os pontos das colunas 3 e 4 para construir a segunda função de custo.

Note que, com essa interpretação, parece que temos dois conjuntos de vértices, um definido pelas colunas 1 e 2, e outro definido pelas colunas 3 e 4. Conforme explicitado no primeiro parágrafo da seção, esse não é o caso. Cada linha é um vértice e os valores fornecidos servem apenas para calcular a distância euclidiana e usá-la como peso para as arestas. Dessa forma, a aresta que liga os vértices representados pelas linhas 12 e 84, por exemplo, possui dois pesos diferentes, um para ser utilizado no primeiro ciclo e outro para ser utilizado no segundo.

5.3 Geração das Instâncias

Para gerar instâncias de um dado tamanho N, utilizamos as primeiras N linhas dos dados fornecidos.

5.4 Resultados

TO DO

5.5 Análise dos Resultados

TO DO

Referências

[1] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*, vol. 6. Athena Scientific Belmont, MA, 1997.

[2] W. Cook, "Concorde tsp solver," 2022.

[3] D. Applegate, W. Cook, S. Dash, and M. Mevenkamp, "Qsopt linear programming solver," 2022.

[4] B. Stroustrup, *The C++ programming language.* Pearson Education, 2013.