UNIVERSIDADE FEDERAL DO CEARÁ
Campus de Quixadá
Prof. Paulo Henrique Macêdo
QXD0181- Pesquisa Operacional

TP01
2019.2

O Problema de Conjunto Dominante Mínimo Conexo

Dado um grafo conexo não direcionado $G = (V, E)$, onde V é o conjunto de vértices e E é o conjunto de arestas, um conjunto dominante de G é um conjunto $D \subseteq V$ tal que $\Gamma(D) = V$, onde $\Gamma(D) = D \cup \{j \in V \mid (i, j) \in E, i \in D\}$. Um conjunto dominante conexo é um conjunto dominante D tal que o subgrafo $G(D) = (D, E(D))$ seja conexo, onde $E(D) = \{(i, j) \in E \mid i \in D, j \in D\}$. O problema do conjunto dominante mínimo conexo, ou *Minimum Connected Dominating Set Problem* (*MCDSP*), consiste em determinar um conjunto dominante conexo de cardinalidade mínima. Em [3], prova-se que *MCDSP* é *NP-Difícil*.

O problema *MCDSP* possui diversas aplicações, tais como: desenho de redes ad-hoc de sensores sem fio, onde topologias de rede podem mudar dinamicamente [1], estratégias de defesa contra o ataque de worms em redes peer-to peer [6] e estudo das interações proteína-proteína [5]. Ele está diretamente relacionado ao problema de *Árvore Geradora com o Máximo Número de Folhas* (ver [4] e [2] para mais detalhes).

QUESTÕES

1. Considere o modelo de programação linear intera para o MCDSP abaixo:

$$
\begin{array}{llll}
(M1) & \min & \sum_{i \in V} y_i & (1)\\
& \text{s.a.} & \sum_{j \in \Gamma(\{i\})} y_j - \sum_{e \in E(\Gamma(\{i\}))} x_e \geq 1, \quad i \in V & (2)\\
& & \sum_{e \in E} x_e = \sum_{i \in V} y_i - 1 & (3)\\
& & y_i \leq \sum_{e \in E_i} x_e, \quad i \in V & (4)\\
& & x_e \leq y_i, x_e \leq y_j, \quad e = (i, j) \in E & (5)\\
& & \sum_{e \in E(S)} x_e \leq |S| - 1, \quad S \subseteq V, |S| > 2 & (6)\\
& & y_i \in \{0, 1\}, i \in V \qquad x_e \in \{0, 1\}, e = (i, j) \in E, &
\end{array}
$$

onde $E_i = \{e = (i, j) \mid j \in V, (i, j) \in E\}$. Nesse modelo, as variáveis binárias y se referem aos vértices e x às arestas de G. Essas variáveis indicam para cada vértice i (cada aresta e) se pertencem ao conjunto dominante, $y_i = 1$ ($x_e = 1$), ou não, $y_i = 0$ ($x_e = 0$). A função objetivo (1) minimiza a cardinalidade do conjunto dominante conexo. A restrição (2) garante, para todo vértice, que ele está ou possui algum vértice adjacente no conjunto dominante mínimo. A restrição (3) induz um conjunto dominante com quantidade de arestas igual à quantidade de vértices menos uma unidade. Se um vértice está no conjunto dominante em uma solução ótima, então existe alguma aresta conectada a ele (restrição (4)). E se uma aresta está no conjunto dominante em uma solução ótima, então seus dois extremos também estão (restrição (5)). A restrição (6) (também chamada de *SEC*) é utilizada para que o conjunto dominante ótimo seja acíclico. Essa restrição junto à restrição (3) garantem a conectividade do conjunto dominante ótimo pois forma uma árvore geradora do subgrafo induzido por y.

Implemente o modelo acima ignorando as restrições SEC (6) e as de integralidade das variáveis. Teste para todas as instâncias DIMACS usando um pacote de otimização.

2. Adicione a restrição de integralidade para todas as variáveis, isto é, exija que o valor das variáveis seja binário, e execute o pacote de otimização novamente para resolver o modelo inteiro para as mesmas instâncias. **Compare os resultados obtidos com os resultados da questão anterior e verifique em quais instâncias a solução inteira ótima retornada tem um conjunto dominante que é conexo.**

Referências

[1] Balabhaskar Balasundaram and Sergiy Butenko. *Handbook of Optimization in Telecommunications*, chapter Graph Domination, Coloring and Cliques in Telecommunications, pages 865–890. Springer US, Boston, MA, 2006.

[2] Tetsuya Fujie. An exact algorithm for the maximum leaf spanning tree problem. *Computers & OR*, 30(13):1931–1944, 2003.

[3] Michael R. Garey and David .S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[4] Abilio Lucena, Nelson Maculan, and Luidi Simonetti. Reformulations and solution algorithms for the maximum leaf spanning tree problem. *Computational Management Science*, 7(3):289–311, 2010.

[5] Tijana Milenkovic, Vesna Memisevic, Anthony Bonato, and Natasa Przulj. Dominating biological networks. *PLoS ONE*, 6(8):1–12, 08 2011.

[6] Liang Xie and Sencun Zhu. A feasibility study on defending against ultra-fast topological worms. *Peer-to-Peer Computing, 2007. P2P 2007. Seventh IEEE International Conference on*, pages 61–70, 2007.