Dominação Vetorial na Família dos Grafos *Split*-Indiferença

Rodrigo Lamblet Mafort[*1], **Fábio Protti**[*1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brasil

{rmafort,fabio}@ic.uff.br

Abstract. *This work presents a polynomial time algorithm capable of solving the Vector Domination Problem for Split-Indifference graphs. The proposed method is based on two characteristics of this class of graphs: the restriction about the number of simplicial vertices and the division in at most three maximal cliques.*

Resumo. *Este trabalho apresenta um algoritmo polinomial capaz de solucionar o Problema da Dominação Vetorial para grafos Split-Indiferença. O método proposto decorre de duas características inerentes a esta classe de grafos: a limitação do número de vértices simpliciais e a divisão em no máximo três cliques maximais.*

1. Introdução

O Problema da Dominação Vetorial (do inglês *Vector Domination Problem*) busca, dados um grafo $G = (V, E)$, um vetor de inteiros $R = (R[v] \in \{0, 1, \ldots, d(v)\} : v \in V\}$ determinar um conjunto $S \subseteq V$ mínimo tal que todo vértice $v \in V$ ou está contido em S ou possui $R[v]$ vizinhos neste conjunto. Um conjunto S capaz de dominar todos os vértices de G é denominado um conjunto R-dominante.

Existem outras variações deste problema, formadas pela aplicação de restrições adicionais. A variação mais conhecida é o Conjunto Dominante, onde todos os vértices possuem requisitos unitários ($R[v] = 1 \ \ \forall v \in V$).

A demonstração da complexidade computacional do problema da Dominação Vetorial decorre justamente desta variação. Considerando que o problema do Conjunto Dominante é NP-Completo para grafos em geral [Garey and Johnson 1990] e constitui uma restrição ao problema da Dominação Vetorial conclui-se que o Dominação Vetorial também é NP-Completo para tais grafos.

O estudo da complexidade computacional do problema da dominação vetorial na classe dos grafos *split*-indiferença foi motivado pela intratabilidade deste mesmo problema quando restrito aos grafos *split*, conforme demonstrado por [Bertossi 1984]. Outro ponto que instigou esta pesquisa foi a indeterminação da complexidade computacional deste problema no escopo dos grafos de intervalo próprio. Este trabalho é parte de uma análise mais aprofundada que busca identificar os limiares da intratabilidade do problema da dominação vetorial em algumas classes de grafos.

[*]Os autores agradecem a CAPES e a FAPERJ o apoio recebido para execução deste trabalho.

2. Grafos *Split*-Indiferença

Os grafos Split-Indiferença são constituídos pela interseção entre a classe dos Grafos *Split* e os Grafos de Intervalo Próprio, isto é, atendem em simultâneo as restrições inerentes a ambas as classes. A primeira definição formal desta classe foi apresentada por [Ortiz et al. 1998] no seguinte teorema:

Teorema 1. *[Ortiz et al. 1998] Seja G um grafo conexo, $G = (V, E)$ é um grafo split-indiferença se e somente se*

1. *G é um grafo completo, ou*
2. *G possui duas cliques maximais C_1 e C_2, tais que $|C_1 \setminus C_2| = 1$ ou*
3. *G possui três cliques maximais C_1, C_2 e C_3, tais que*
   - $|C_1 \setminus C_2| = |C_3 \setminus C_2| = 1$
   - $C_1 \cap C_3 = \emptyset$ *ou* $C_1 \cup C_3 = V$

O Teorema 1 prova a existência de quatro casos possíveis de grafos split-indiferença. Sua definição é baseada no número de cliques e em suas interseções.

3. Algoritmo proposto

O algoritmo desenvolvido para o problema da dominação vetorial em grafos split-indiferença é baseado no Teorema 1, uma vez que para cada caso apresentado no teorema, um método distinto é aplicado. Sejam o grafo $G = (V, E)$ e o vetor de requisitos R as entradas do algoritmo e seja S o conjunto R-dominante em construção.

O primeiro caso do Teorema 1 consiste de apenas uma única clique e possui solução trivial: inicialmente, os vértices são ordenados de forma decrescente por seus requisitos. Em seguida, essa ordenação é percorrida e a cada iteração um vértice $x \in V$ é analisado. Se $R[x] > |S|$, o algoritmo adicionará x ao conjunto S. Do contrário, x já foi dominado e o algoritmo pode ser encerrado (os demais vértices estão dominados por S).

No segundo caso, o grafo é constituído de duas cliques maximais, sendo que uma delas possui exatamente um vértice simplicial v. Considerando que v pode ou não estar contido na solução, os dois casos são analisados. O primeiro teste consiste de incluir v em S, atualizar os requisitos dos vizinhos de v para sinalizar que já possuem um vizinho em S e resolver a outra clique, que, por definição, contém todos os vértices da primeira, exceto v. Finalmente, o conjunto obtido para esta clique é incorporado à S. O segundo teste considera que v não será incluído na solução, implicando que $R[v]$ vizinhos de v devem estar contidos em S. Os requisitos dos vizinhos dos vértices adicionados devem ser debitados em $|S|$ unidades para considerar que já possuem $|S|$ vizinhos em S e, em seguida, a clique induzida por $(V \setminus (S \cup \{v\}))$ é resolvida pelo Caso 1 e conjunto obtido é incorporado à S. A solução para este caso consiste do conjunto de menor cardinalidade dentre os dois obtidos.

Já no terceiro caso descrito existem três cliques C_1, C_2 e C_3 tais que $|C_1 \setminus C_2| = |C_3 \setminus C_2| = 1$ e $C_1 \cap C_3 = \emptyset$. Tendo em vista que C_1 e C_3 não possuem vértices em comum, é possível utilizar o mesmo procedimento aplicado ao caso anterior, considerando não apenas um vértice simplicial, mas dois (v e w). Desta forma, existem quatro casos que devem ser analisados isoladamente: (a) $v \notin S$ e $w \notin S$; (b) $v \in S$ e $w \notin S$; (c) $v \notin S$ e $w \in S$; (d) $v \in S$ e $w \in S$. A solução para este caso também consiste do menor conjunto dentre os obtidos.

O quarto e último caso apresentado pelo Teorema 1 também é constituído por três cliques C_1, C_2 e C_3, tais que $|C_1 \setminus C_2| = |C_3 \setminus C_2| = 1$. Contudo, difere do caso anterior pela existência de vértices comuns as cliques C_1 e C_3, o que inviabiliza o estudo isolado do caso onde v e w não estão presentes em S, uma vez que ao adicionar vizinhos de v neste conjunto, o requisito de w também é impactado. Para este caso uma nova abordagem foi aplicada. Os demais casos podem ser resolvidos através do método apresentado para o terceiro caso do teorema, pois v e w não são adjacentes.

Inicialmente assume-se que v e w não participarão do conjunto S, e, portanto, devem ser dominados por seus vizinhos. Na abordagem desenvolvida para este caso, são construídas soluções S_1 e S_3 para as cliques C_1 e C_3. Considerando que $C_1 \cup C_3 = V(G)$, sabe-se que $S = S_1 \cup S_3$ é capaz de dominar G, contudo, não se pode afirmar ainda que S é mínimo. Desta forma, o algoritmo inicia uma fase que busca reduzir S através de sucessivas remoções e trocas.

Na fase de remoções, os vértices contidos em S são estudados, em ordem crescente de requisitos, para avaliar se sua exclusão transformaria um vértice já dominado em não dominado. Caso essa exclusão seja possível, esse vértice é imediatamente removido de S. Quando nenhum vértice puder ser removido, o algoritmo passará a procurar formas de diminuir S utilizando trocas de vértices.

As trocas são movimentos que removem dois vértices de S e acrescentam apenas um. Para que isso seja possível, três condições devem ser satisfeitas: (a) São necessários dois vértices x_1 e x_3 tal que $x_1 \in ((C_1 \setminus C_3) \cap S)$ e $x_3 \in ((C_3 \setminus C_1) \cap S)$; (b) Todos os vértices de $(C_2 \setminus (S \setminus \{x_1, x_3\}))$ devem possuir requisitos estritamente menores que o tamanho de S; (c) Existência de um vértice z tal que z não esteja contido em S e seja adjacente à v e à w. Se essas três condições forem satisfeitas, a troca é executada: x_1 e x_3 são removidos e z é adicionado ao conjunto S. Dentre todos os vértice que atendam as restrições, x_1 e x_3 correspondem aos de menores requisitos. Já z corresponde ao vértice de maior requisito que atenda as restrições.

Uma vez que todas as trocas possíveis forem executadas, os vértices de S são analisados novamente em busca de novas remoções. Quando nenhuma remoção for possível, S é um conjunto R-dominante para o grafo G.

3.1. Corretude do algoritmo

Teorema 2. *O algoritmo proposto encontra um Conjunto R-Dominante S para um grafo split-indiferença G e um vetor de requisitos R.*

Demonstração. Caso 1: Considerando que nenhum conjunto de cardinalidade inferior à S é suficiente para converter o último vértice adicionado e que todos os anteriores a esse possuem requisitos superiores, conclui-se que S é mínimo.

Caso 2: Sejam C_1 e C_2 as duas cliques maximais de G e seja v o vértice simplicial de C_1. Ainda, seja S_{C_2} o conjunto de todos os conjuntos R-dominantes para C_2 e seja A o conjunto dos $R[v]$ vizinhos de v de maiores requisitos. Supondo que existe um conjunto $S \in S_{C_2}$ capaz de converter v, sabe-se que um destes conjuntos contém A. Desta forma, seja S um conjunto de S_{C_2} que contenha A. Além disso, sejam G' o subgrafo induzido por $(V \setminus (A \cup \{v\}))$ e $R'[u] = R[u] - |A|\ \forall u \in V(G')$ o vetor de requisitos atualizado. Aplicando o Caso 1 do algoritmo é possível obter um conjunto R'-dominante para G'.

Seja B esse conjunto. Sabe-se que $A \cup B$ é capaz de dominar todos os vértices de G (A domina v e B, os demais), contudo, resta demonstrar que este conjunto é mínimo. Tendo em vista como A e B foram construídos, pode-se concluir que $S \subseteq (A \cup B)$. Desta forma, supondo, por absurdo, que $A \cup B$ não seja mínimo, seja u o vértice que pode ser removido de $A \cup B$. Considerando que A possui os vértices de maiores requisitos dentre os vizinhos de v e que nenhum subconjunto de A domina v, conclui-se que $u \in B$. Como B contém os vértices de maiores requisitos de G' pode-se deduzir que $R'[u] \geq |B|$. Em contrapartida, considerando que u pode ser removido de $A \cup B$, conclui-se que $R[u] < |A| + |B|$, o que é absurdo, pois $R'[u] = R[u] - |A|$. Sendo assim, fica demonstrado que $A \cup B$ é mínimo.

Resta demonstrar o caso em que nenhum conjunto $S \in S_{C_2}$ é capaz de dominar v, isto é, para dominar v é necessário pelo menos mais um vértice. Desta forma, o próprio v será adicionado ao conjunto R-dominante. Seja R' o vetor de requisitos atualizado para considerar que v já contribui na dominação de seus vizinhos ($R'[u] = R[u] - 1 \ \forall u \in adj(v)$) e seja B um conjunto R'-dominante para C_2. Considerando que B é mínimo, domina C_2, mas não domina v ($B \subseteq S$), conclui-se que $B \cup \{v\}$ é um conjunto R-dominante para G.

Caso 3: Sejam C_1, C_2 e C_3 as três cliques maximais de G, tais que $C_1 \cap C_3 = \emptyset$ e sejam v e w os vértices simpliciais de C_1 e C_3 respectivamente. Considerando que $adj(v) \cap adj(w) = \emptyset$, pode-se aplicar a mesma lógica do caso anterior para os quatro casos possíveis.

Caso 4: Sejam C_1, C_2 e C_3 as três cliques maximais de G, tais que $C_1 \cup C_3 = V$ e sejam v e w os vértices simpliciais de C_1 e C_3 respectivamente. Nos casos em que v ou w estão contidos no conjunto R-dominante, a demonstração segue o caso anterior. Entretanto, o caso em que v e w não estão contidos no conjunto R-dominante deve ser demonstrado. Considerando que nos demais casos o algoritmo encontra um conjunto R-dominante (demonstrado no Caso 3), supõe-se que não existe um conjunto R-dominante que contenha v ou w, pois do contrário este seria localizado pelos casos anteriores. Contudo, tendo em vista as limitações relativas ao tamanho deste resumo, esta parte da prova foi omitida.

Tendo em vista que em todos os casos o algoritmo localizou um Conjunto R-Dominante para o grafo G, conclui-se que o método apresentado está correto. □

A complexidade do algoritmo proposto é $O(n^2)$, pois no pior caso, ao selecionar um vértice, todo o vetor de requisitos deve ser percorrido para atualização.

Referências

Bertossi, A. A. (1984). Dominating sets for split and bipartite graphs. *Information Processing Letters*, 19(1):37–40.

Booth, K. S. (1980). Dominating Sets in Chordal Graphs. Technical report, University of Waterloo.

Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

Ortiz, C. Z., Maculan, N., and Szwarcfiter, J. L. (1998). Characterizing and edge-colouring split-indifference graphs. *Discrete Applied Mathematics*, 82(1-3):209–217.