ABSTRACT

KEYWORDS: Domination, Disjunctive Domination, Interval Graphs, Proper Interval Graph, Clique Elimination Ordering.

We consider the recent variation of domination called Disjunctive Domination in Special classes of Graphs called Interval Graphs. Approaches to this problem like Greedy are tried and counter example where the approach didn't work are given. We have proposed a new Elimination ordering for Interval Graphs called Clique Elimination Ordering. Disjunctive Domination problem being solved on sub-class of Interval Graphs i.e Proper Interval Graphs, the solution is extended to Interval Graph using the above proposed ordering whose correctness has to be proved.

Contents

List of Figures

ABBREVIATIONS

LVO	Left endpoint Vertex Ordering
RVO	Right endpoint Vertex Ordering
BCO	Bi-Compatible elimination Ordering
IG	Interval Graph
PIG	Proper Interval Graph
PEO	Perfect Elimination Ordering
CEO	Clique Elimination Ordering

Chapter 1

INTRODUCTION

We consider a city as a graph viz. places as nodes and roads connecting them as edges. If a situation of emergency arises in a place, then the response time for emergency system should be minimal; However we cannot afford to put emergency systems at every place. So we want to find minimum number of emergency systems that can "cover" the whole city. This type of situations are common in communication systems, locating facilities, vigilance systems etc., and are modeled by domination problem in graph theory. It is a variant of the famous NP-Complete Set Covering Problem. The number of facilities can be decreased further by increasing "power" of facilities. This leads to varieties of domination problems. There are different variants of dominating problems like R(x)-Domination, Exponential Domination, Secondary Domination, Distance Domination etc. **?** recently introduced a variation in domination problem inspired from Exponential Domination and similar ideas.

1. Disjunctive Domination

? initially defined *b-disjunctive dominating* set of a Graph G(V,E) as a set S if every vertex, $v \in V \setminus S$ is either adjacent to a vertex in S (or) there exist atleast b vertices in S at a distance of two from v. If $b = 2$ in above case then it is 2-Disjunctive Domination and for simplicity called as Disjunctive Domination.

Definition 1.1. A set S of vertices is said to Disjunctively Dominate a Graph,G(V,E) if every vertex, $v \in V \setminus S$ is either adjacent to a vertex in S (or) there exist atleast two vertices in S at a distance of two from v.

This can also be viewed as every vertex has a dominating power of one within distance one and half at a distance of two and zero thereafter. Given a graph finding minimum size of Disjunctive Dominating set is called **Disjunctive Domination Problem**.

This problem is shown to be NP-Complete for Planar and Bi-partite graphs. It has a polynomial time algorithm for Trees and Proper Interval Graphs.

2. Interval Graph

Intersection graph is a graph that shows the intersection patterns of the family of sets,F. Each node of a graph represents a set,S which belongs to the family,F. An edge exists between two vertices iff their corresponding sets have a non-empty intersection. Infact every graph is an intersection graph. Imposing constraints on the family of sets gives us special classes of graphs.

Definition 1.2. A graph is said to be Interval Graph,(IG) if it can be represented as intersection graph of family of intervals on the real line. Each vertex is mapped to an interval on real line and edge between two vertices exists if and only if the intervals intersect.3.1

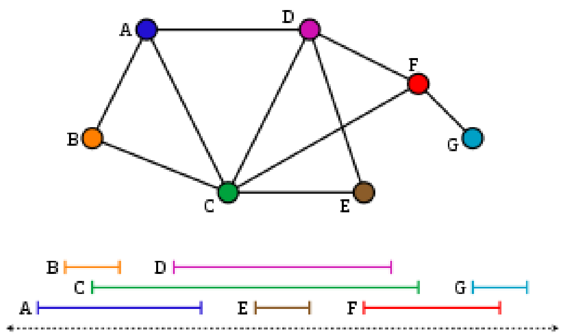

Figure 1.1: Intervals on Real line and Corresponding Interval Graph
Image Source:Wiki

Every Interval Graph is characterized by having a maximal clique ordering and every vertex in interval graph belongs to a continuous set of cliques in the clique ordering of the Interval Graph.

3. Organization of Thesis

Initially we show some equivalent characterizations to Bi-Compatible Elimination Ordering on Proper Interval Graph,PIG. We then attempt to solve the Disjunctive Domination Problem on Interval Graphs using labeling method. It is similar to method used on Trees which uses famous Min Cost-Max Flow problem. We show a Counter Example where the method fails in our case. As the problem is solved on Proper Interval Graphs, We extend this algorithm to its super class, Interval Graph. We introduce a new characterization for Interval Graphs similar to Bi-Compatible

Elimination Ordering(in case of PIG) called Clique Elimination Ordering. However the correctness of the algorithm have to be proved.

Chapter 2

Proper Interval Graph

Definition 2.1. An interval graph in which no interval is contained in another interval is called Proper Interval Graph(PIG).

Definition 2.2. The complete Bi-partite graph $K_{1,3}$ is called Claw Graph,fig:2.1. It is isomorphic to the Star Graph,S_4.

PIG can also be characterized by forbidden graph characterization.

Theorem 2.3. An interval graph whose induced sub-graph is not a Claw,fig:2.1 is called Proper Interval Graph.**?**

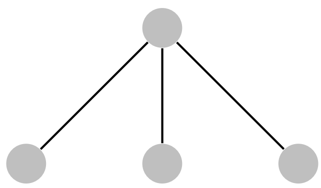

Figure 2.1: Claw Graph

Definition 2.4. A vertex in a graph G(V,E) is said to be **Simplicial vertex** if all it's adjacent vertices form a clique.

Definition 2.5. Perfect Elimination Ordering,$(v_1, v_2, \cdots, v_n)$ of a Graph,G(V,E) is the ordering of vertices of G, such that each vertex,v_i is simplicial vertex of the induced subgraph, $G[v_i, v_{i+1}, \cdots, v_n]$.

Theorem 2.6. A Graph,G(V,E) have a Perfect Elimination Ordering(PEO) iff it is Chordal Graph.**?**

Definition 2.7. A PEO of Chordal Graph is Bi-Compatible Elimination Ordering,$(v_1, v_2, \cdots, v_n)$ iff the reverse ordering i.e $(v_n, v_{n-1}, \cdots, v_1)$ is also a PEO of the Graph,G(V,E).

Theorem 2.8. A Graph,G(V,E) have a Bi-Compatible elimination Ordering(BCO) iff it is Proper Interval Graph.**?**

Let's say $[l_i, r_i]$ is the interval corresponding to vertex, v_i in Proper Interval Graph,G(V,E). We can order the vertices according to the endpoints.

Definition 2.9. Left endpoint Vertex Ordering(LVO) is the ordering of vertices in the ascending order of their left endpoints,l_i.

Definition 2.10. Right endpoint Vertex Ordering(RVO) is the ordering of vertices in the ascending order of their right endpoints,r_i.

Now we will prove that in the case of PIG all these orderings are same.

Theorem 2.11. In case of PIG, Left Vertex Ordering(LVO) of vertices is same as Right Vertex Ordering(RVO) is same as Bi-compatible Elimination Ordering(BCO).
$LVO \equiv RVO \equiv BCO$

Proof:

$LVO \equiv RVO$

Let us assume the contrary.

Then $\exists$ two vertices v_i, v_j such that v_i comes before v_j in LVO

$$\implies l_i < l_j$$

and v_j comes before v_i in RVO

$$\implies r_j < r_i$$

where l_i, r_i are the left and right endpoints of the interval represented by the vertex v_i.

$$\implies l_i < l_j < r_j < r_i$$

We can see that interval j is subset of the interval i which leads to contradiction that the graph is Proper Interval Graph.

$$\text{Hence, } LVO \equiv RVO \tag{1}$$

$LVO \equiv BCO$

To prove that v_i is simplicial vertex $\forall v_i \in V$ in the ordering LVO. Let $v_{Max(i)}$ is the vertex that is maximum among the vertices adjacent to v_i. Since it is left endpoint ordering every vertex between v_i and $v_{Max(i)}$ is adjacent to v_i.

Its enough to prove $G[v_i, v_{i+1}, \cdots v_{Max(i)}]$,induced graph is a clique.

$$\begin{array}{cc} \because LVO & l_i < l_{i+1} < \cdots < l_{Max(i)} \\ \text{From (1)} & r_i < r_{i+1} < \cdots < r_{Max(i)} \end{array}$$

$$\because l_i \text{ and } l_{Max(i)} \text{ has edge } \implies l_{Max(i)} < r_i$$

$$\therefore l_i < l_{i+1} < \cdots < l_{Max(i)} < r_i < r_{i+1} < \cdots < r_{Max(i)}$$

$$\forall i \leq x < y \leq Max(i)$$

$$l_x < l_y < r_x < r_y \implies v_x \text{ and } v_y \text{ has edge}$$

$$G[v_i, v_{i+1}, \cdots v_{Max(i)}] \text{ forms a clique.}$$

$$\text{Hence, } LVO \equiv BCO \tag{2}$$

Similarly we can prove that $G[v_{low(i)}, \cdots v_{i-1}, v_i]$ is a clique; where $v_{low(i)}$ is the smallest vertex adjacent to v_i.

This shows that in the reverse ordering,$(v_n, \cdots, v_1)$ also each vertex v_i is simplicial in induced graph $G[v_i, v_{i-1}, \cdots, v_1]$. making it a Perfect Elimination Ordering. Hence it is Bi-Compatible elimination Ordering.

From (1) and (2) $LVO \equiv RVO \equiv BCO$ $\square$

The complexity of getting BCO from Proper Interval Graph is $O(n+m)$ where n is number of vertices and m is number of edges. However it is $O(n \log n)$. Best case order of m is having k cliques with $\frac{n}{k}$.

Total number of edges = $k * \frac{n}{k} * \frac{n}{k} = \frac{n^2}{k}$

If number of cliques(k) are less when compared to n, then m is large in which case $O(n \log n)$ is better than $O(n+m)$. And so computing LVO is faster than BCO.

Chapter 3

Labeling method on Interval Graphs

1. Labeling method

A polynomial time Algorithm for Disjunctive Domination Problem on Trees is given using Labeling approach by **?**. They have used labeling method where we label the vertices with different labels for denoting the domination status of vertices. In case of trees it can be observed that the Dominating Set of a Tree need not contain any leaves. So, all the un-dominated leaves have to be dominated by Internal vertices of the tree. Model the domination problem as Min Cost-Max Flow Problem. Find a Partial dominating set which dominates all the leaves. Remove the vertices which are dominated by the updated dominating set, from the tree, which results in a smaller tree. Repeat the process until all the vertices are dominated.

A detailed explanation of construction of flow graph is given in **?**.

2. Application to Interval Graph

Every Interval graph has a clique ordering and so there exist some vertices (at least 2 from both ends) which belongs to one clique. It can be proved that Disjunctive Dominating set only contain vertices belonging to more than one clique.

Definition 3.1. The vertices present in the intersection of two cliques separate the other vertices in both cliques. So, the intersection is called separator set for the cliques.

Theorem 3.2. For any Interval Graph G, there is a Disjunctive Dominating set D such that D does not contain any vertices belonging to single clique.

Proof: Let us assume that there is a vertex v_p which is only in one clique and also present in the Disjunctive dominating set,D. We take a vertex v_r which is in the intersection of this clique with other cliques. Since v_r is also belonging to the clique so all the vertices adjacent to v_p are also adjacent to v_r.

$$\text{Hence, } Adj(v_p) \subset Adj(v_r)$$

Similarly any vertex which is at a distance of two from v_p is also at a maximum distance of two from v_r.

$$\text{Hence, } Adj^2(v_p) \subseteq Adj^2(v_r) \cup Adj(v_r)$$

$\therefore D \setminus \{v_p\} \cup \{v_r\}$ is also Disjunctive Dominating set of the Graph. □

Hence we can say that vertices belong to single clique need not be there in dominating set. So all vertices belonging to one clique have to be dominated by those in the separators. This observation is similar to the one in trees(leaves vs internal vertices).

The algorithm is similar to the one in paper **?** where only the definitions of the sets X and Y change.

Input: Interval Graph,G(V,E)

Output: Disjunctive Dominating Set

Initialization: $D = \phi$, G'(V,E) is same as G(V,E) with all vertices labeled as 2

; **while** *(all vertices are not labeled zero)* **do**

$Y = \{y|y \in$Single Clique in the graph $G'\}$

$X = \{x|x$ is at a distance ≤ 2 from y$\in Y\}$

Construct a flow graph as given in **?**

Find a partial dominating set D_{part} using Min Cost-Max Flow

$D = D \cup D_{part}$

Update the labels of the vertices as per the new dominating set D

Delete all the vertices from graph,G' whose labels are zero i.e completely dominated

end

Algorithm 1: Labeling method on Interval Graph

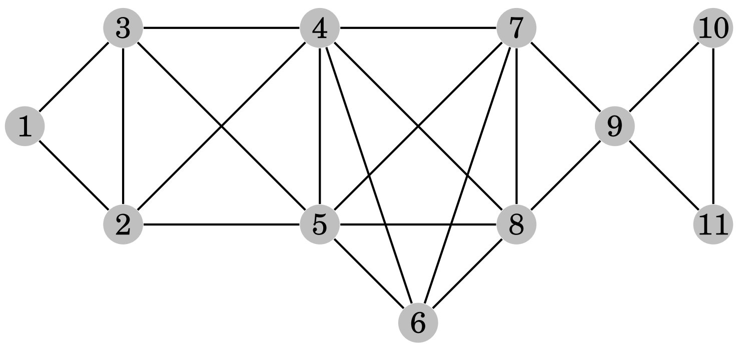

Figure 3.1: Interval Graph

Applying Min Cost Max Flow on figure 3.2 gives a dominating set of $\{2,9\}$ which dominates the whole graph 3.1.

In the case of figure 3.3 constructing the flow graph and applying the Min cost

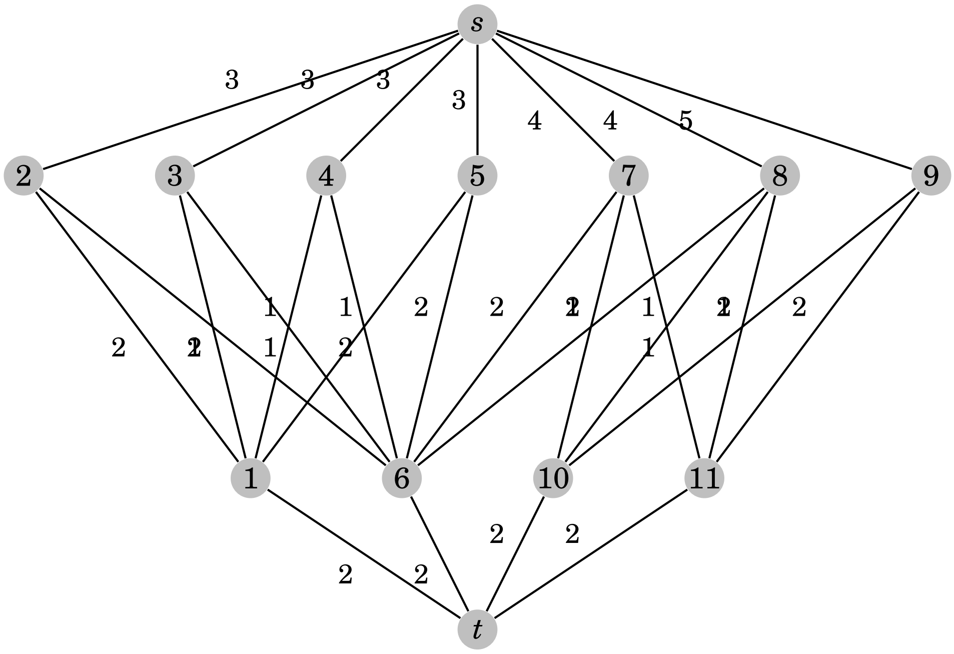

Figure 3.2: Flow graph constructed from graph 3.1 as per the algorithm

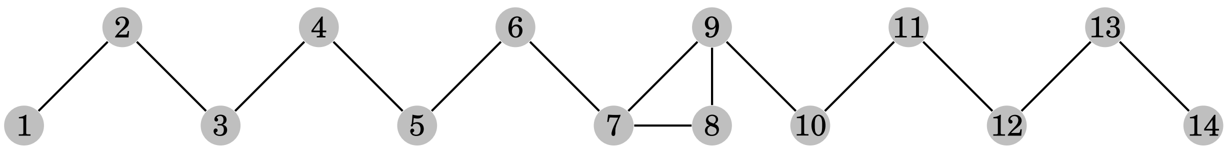

Figure 3.3: Interval Graph

Max flow algorithm gives us three choices with same cost. They are $\{2,7,13\}$, $\{2,9,13\}$, $\{2,6,10,13\}$. Here although we may be tempted to choose that set which is smaller but we should choose the third set because it can dominate more vertices when compared to other two choices. So in case of multiple options, instead of choosing that set which is smaller choose the one that can dominate more vertices.

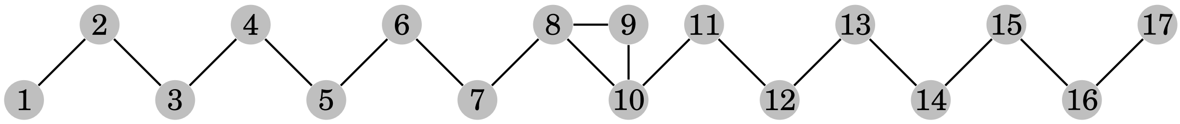

Figure 3.4: Interval Graph

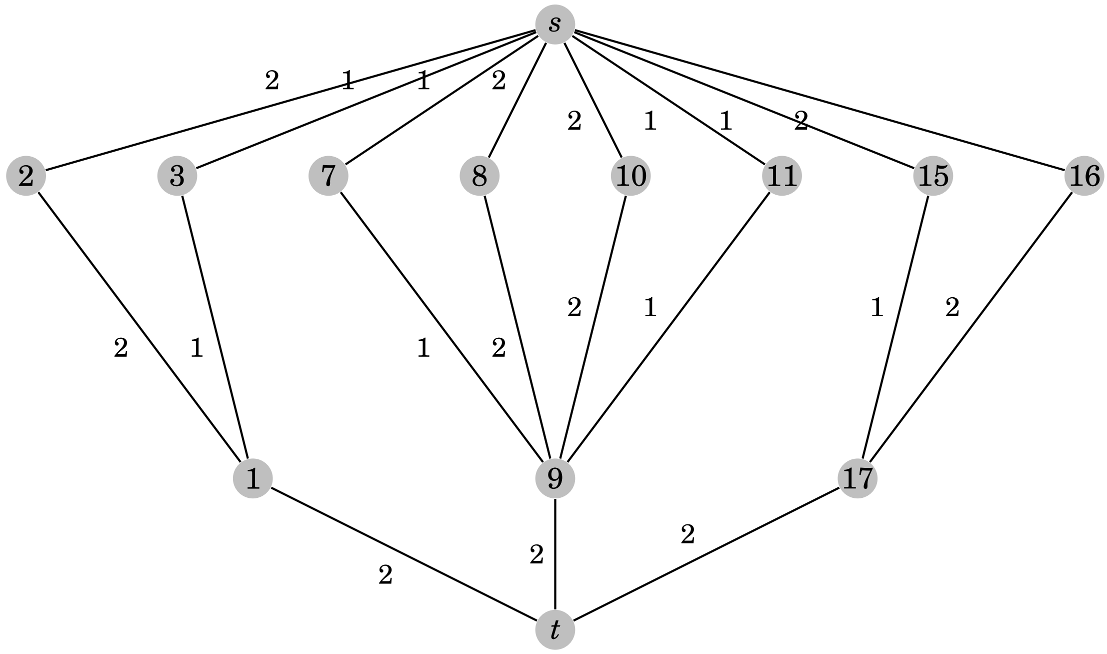

Figure 3.5: Flow graph constructed from graph 3.4 as per the algorithm

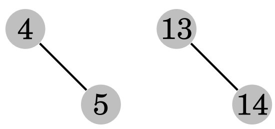

Figure 3.6: Interval Graph reduced and all vertices now have label one.

Applying Min cost-Max flow on the flow graph 3.5 gives us the partial dominating set $\{2,7,11,16\}$ as it has maximum span. Remaining graph with labels is given in figure 3.6. Repeating the process leaves us with two more vertices each from both components, say $\{4,13\}$. Thus giving us a disjunctive dominating set $\{2,4,7,11,13,16\}$ whose size is 6. However we can find a smaller disjunctive dominating set of size 5 which is $\{2,6,10,14,16\}$.

The labeling method is not producing the optimal solution as the single clique vertices are looking for locally optimal solution oblivious of the remaining part of the graph. Here we have chosen vertex 7,11 to dominate vertex 9 however choosing vertex 10 would have given optimal result. However this decision is possible only if algorithm considers structure of the whole graph. So, a greedy solution doesn't work in this case.

Chapter 4

Interval Graph

1. Clique Elimination Ordering

Every Interval Graph has a maximal clique ordering,$(C_1, C_2, \cdots, C_k)$ in which every vertex belong to a continuous set of cliques.

Definition 4.1. $level(v_i)$ is the maximum index,l in the clique ordering of the Interval Graph such that C_l contains vertex,v_i.

Definition 4.2. $Max(v_i)$ is the largest vertex in the CEO,$(v_1, v_2, \cdots, v_n)$ of the Graph,G(V,E) i.e adjacent to v_i.

Definition 4.3. Clique Elimination Ordering,CEO is the ordering of the vertices of the Interval Graph,G(V,E) in the ascending order of $level(v_i)$ and if two vertices have same $level(v_i)$ then order them by the left endpoint of the intervals represented by them.

Lemma 4.3.1. Every vertex,v_i in CEO is a simplicial vertex of the induced graph,G$[v_i, v_{i+1}, \cdots, v_n]$.

Proof: According to ordering,CEO all vertices in the ordering after v_i, either belong to clique,$C_{level(v_i)}$ or a greater clique. So all vertices adjacent to vertex v_i belongs to clique $C_{level(v_i)}$. Hence vertex v_i is simplicial vertex of the induced Graph,G[$v_i, v_{i+1}, \cdots, v_n$]. □

Theorem 4.4. In CEO,($v_1, v_2, \cdots, v_n$) of Graph G(V,E) all vertices between v_i and $Max(v_i)$ are adjacent to $Max(v_i)$.

Proof: Since v_i is adjacent to $Max(v_i)$

$\Longrightarrow$ Both of them are in clique $C_{level(v_i)}$ (1)

$Max(v_i)$ is in the clique $C_{level(Max(v_i))}$ (2)

Since each vertex belongs to continuous set of Cliques

From (1) and (2) $Max(v_i)$ belongs to all cliques between $C_{level(v_i)}$ and $C_{level(Max(v_i))}$

$\forall$ vertex, $v_x \in (v_i, Max(v_i))$

$$level(v_i) \leq level(v_x) \leq level(Max(v_i))$$
$$\Longrightarrow Max(v_i) \text{ and } v_x \text{ are in the clique } C_{level(v_x)}$$
$$\Longrightarrow v_x \text{ is adjacent to } Max(v_i)$$
□

2. Extended Algorithm

Using Clique Elimination Ordering we extend the algorithm given for Proper Interval Graphs,**?** to Interval Graphs. The algorithm is as follows.

Input: Clique Elimination Ordering,$(v_1, v_2, \cdots, v_n)$ of Interval Graph

Output: Disjunctive Dominating Set

Initialization: $D = \phi$

for $i = 1; i \le n; i++$ **do**

  $S_1 = Adj[v_i] \cap D$

  $S_2 = Adj^2[v_i] \cap D$

  case $|S_1| \ge 1 \vee |S_2| \ge 2$ **do**

    No update needs to be done;

  end

  case $S_1 = \phi \wedge S_2 = \phi$ **do**

    $D = D \cup Max(v_i)$;

  end

  case $S_1 = \phi \wedge S_2 = \{v_r\}$ **do**

    $v_j = Max(v_i), v_k = Max(v_j)$;

    $S = \{v_{i+1}, \cdots, v_{j-1}\}$

    $T = \{v_x | d(v_x, v_i) = 2 \wedge x > j\} = \{v_l, \cdots, v_k\}$

    if $\exists$ *a vertex in* T *that dominate* S **then**

      Choose Maximum vertex from T dominating S and let it be v_t

      $D = D \cup \{v_t\}$

    else

      Let v_s be the vertex in S that is not dominated by $v_l \in T$

      $D = D \cup \{Max(v_s)\}$

    end

  end

end

Algorithm 2: Disjunctive Domination on Interval Graph

The correctness of the algorithm has to be proved yet.