Algorithms on 2-Acyclic-Subtree Filament Graphs

Fanica GAVRIL

Computer Science Dept., Technion, Haifa 32000, Israel, gavril@cs.technion.ac.il.

ABSTRACT: Two intersecting subtrees of a graph $D$ are 2-acyclic if and only if their intersection is a subtree. Let $GI(V,F)$ be the intersection graph of a family of 2-acyclic subtrees of a graph $D$. In the present paper we define intersection graphs of 2-acyclic-subtree filaments on any graph $D$; the subtree filaments are similarly defined as the subtree filaments based on subtrees of a tree. We describe polynomial time algorithms for various problems on 2-acyclic-subtree filament graphs $G$ when their base graphs $GI$ have algorithms for related problems. The problems are to find maximum induced complete bipartite subgraphs, maximum weight holes of a given parity, minimum dominating holes, maximum weight antiholes and some others.

KEYWORDS: 2-acyclic-subtree filament graph, intersection graph, induced biclique, hole of given parity, dominating hole
1 Introduction

We consider only finite graphs $G(V,E)$ with no parallel edges and no self-loops, where $V$ is the set of vertices and $E$ the set of edges; $\text{co}G$ is the complement of $G$. For $U \subseteq V$, $G(U)$ is the vertex subgraph defined by $U$. Two vertices connected by an edge $u \overset{\rightarrow}{\rightarrow} v$ are called adjacent; a directed edge from $u$ to $v$ is denoted $u \overset{\rightarrow}{\rightarrow} v$. We also denote $N_G(v) = \{u \mid u \overset{\rightarrow}{\rightarrow} v \in E\}$, $\mathcal{N}_G(W) = \{u \mid u \overset{\rightarrow}{\rightarrow} v \in E, v \in W, u \notin V-W\}$ and $\mathcal{N}_G[W] = \mathcal{N}_G(W) \cup W$. By a path $p(v_1,v_k) = v_1 \overset{\rightarrow}{\rightarrow} \ldots \overset{\rightarrow}{\rightarrow} v_k$ we always mean a simple path; $p$ is an induced path if it has no chords. A hole $h(v_1,v_k) = v_1 \overset{\rightarrow}{\rightarrow} \ldots \overset{\rightarrow}{\rightarrow} v_k \overset{\rightarrow}{\rightarrow} v_1$ is a chordless cycle with four or more vertices. A hole or an induced path $h$ is dominating in $G$, if every vertex of $G$ is adjacent to a vertex of $h$. A subset of $V$ is a clique (an independent set) if every two of its vertices are adjacent (not adjacent, respectively).

A graph $G$ is an intersection graph of a family $S$ of distinct subsets of a set if there is a one-to-one correspondence between the vertices of $G$ and the subsets in $S$ such that two subsets intersect iff their corresponding vertices are adjacent; $S$ is a representation of $G$. A family of sets is called Helly if every subfamily of mutually intersecting sets has a non-empty intersection. Two distinct sets $b,d$ overlap if $b \cap d \neq \emptyset$, $b \nsubseteq d$ and $d \nsubseteq b$. An oriented graph $G(V,E)$ is transitive if it is acyclic and for every three vertices $u,v,w \in V$, $u \overset{\rightarrow}{\rightarrow} v, v \overset{\rightarrow}{\rightarrow} w \in E$ implies $u \overset{\rightarrow}{\rightarrow} w \in E$ [12]. A source (sink) of an oriented graph is a vertex with no incoming (outgoing, respectively) edges.

A subtree of a graph $D$ is an acyclic connected edge subgraph of $D$. Two subtrees intersect if they have at least one vertex in common. Two intersecting subtrees are called 2-acyclic iff their union is acyclic, iff their intersection is a subtree (connected) [4,10].

A graph is chordal iff it has no holes iff it is the intersection graph of a family of subtrees of a tree $T$ [3]; a chordal graph has $O(|V|)$ maximal cliques [3].

Let $GI(V,F)$ be the intersection graph of a family $FI$ of 2-acyclic subtrees of a graph $D$; we denote by $t(v)$ the subtree in $FI$ corresponding to vertex $v$. For every vertex $X$ of $D$ let $V_X = \{v \mid v \in V, X \in t(v)\}$; $V_X$ is a clique of $GI$, not necessarily maximal. For two vertices $X,Y$ of $D$, we denote $V_{XY} = (V_X - V_Y) \cup (V_Y - V_X)$. Let $M_{GI}$ be the 0-1 matrix whose columns correspond to the vertices of $D$ and the rows correspond to the subtrees in $FI$, such that $M_{GI}(t,X) = 1$ if and only if the subtree $t$ contains the vertex $X$. For a subtree $t$, let $M_{GI,t}$ be the submatrix of $M_{GI}$ defined by the columns $X$ (i.e., vertices of $D$) having $M_{GI}(t,X) = 1$. Reference [10] proves that a graph $GI(V,F)$ given as a matrix $M_{GI}$ of vertices (rows) vs. cliques (columns) is an
intersection graph of a family $FI$ of 2-acyclic subtrees of a graph $D$ iff for every subtree $t$, the graph $GI$, defined by $M_{GI}$, has an intersection representation by subtrees on $t$, that is $M_{GI}$ is chordal. Based on this, [10] describes a polynomial time algorithm to construct for a graph $GI$ given as a matrix $M_{GI}$, an intersection representation by a family of 2-acyclic subtrees of a graph $D$, when such a representation exists. Reference [4] proves that a graph $GI(V,F)$ given as a matrix $M_{GI}$ of vertices (rows) vs. maximal cliques (columns) is an intersection graph of a Helly family $FI$ of 2-acyclic subtrees of a graph $D$ iff for every vertex $v$ of $GI$, the subgraph $GI(N_{GI}(v))$ is chordal; such a graph has at most $O(|V|^2)$ maximal cliques, and for every two non-adjacent vertices $u,v$, the connected components of $GI(N_{GI}(u) \cap N_{GI}(v))$ are cliques. Reference [4] describes a polynomial time algorithm to construct for $GI$ such an intersection representation on a graph $D$ whose vertices correspond to the maximal cliques of $GI$.

In the present paper we define intersection graphs of 2-acyclic-subtree filaments on any graph $D$. For a better understanding we first assume that $D$ is planar drawn without edge intersections in a plane $PL$. Consider a family $FI$ of 2-acyclic subtrees of $D$, and let $GI(V,F)$ be its intersection graph. Let $PP$ be a surface perpendicular to $PL$ whose intersection with $PL$ is exactly $D$. For $t(v) \in FI$, let $PP(t(v))$ be the subsurface of $PP$ whose intersection with $PL$ is exactly $t(v)$. In $PP(t(v))$, above $D$, we connect all the endpoints of $t(v)$ by a continuous function $a(v):t(v) \to \mathbb{R}^+$ (i.e., $a(x)=0$ for every endpoint $x$ of $t(v)$) called a subtree filament, such that if $t(u), t(v)$ overlap the two filaments intersect, if $t(u), t(v)$ are disjoint, the two filaments do not intersect, and if $t(u) \subseteq t(v)$, the two filaments may or may not intersect. Let $G(V,E)$ be the intersection graph of this family of 2-acyclic-subtree filaments; $G(V,E)$ is called a 2-acyclic-subtree filament graph. Consider the set of edges $E2=\{u \to v \mid t(u) \subseteq t(v) \text{ and } a(u) \cap a(v)=\emptyset\}$; hence, $E=F-E2$. Let $E1=coF$. Thus $u \to v \in E1$ iff $t(u) \cap t(v)=\emptyset$ implying that $coE=E1 \cup E2$. We denote $IN_{G}(v)=\{u \mid a(u) \cap a(v)=\emptyset \text{ and } t(u) \subseteq t(v)\}$ and $OUT_{G}(v)=\{u \mid a(u) \cap a(v)=\emptyset \text{ and } t(v) \cap t(u)=\emptyset \text{ or } t(v) \subseteq t(u)\}$. The graph $G(V,E)$ and edge sets $E1, E2$ have the following properties:

**Property A**: The graph $HG(V,E2)$ is transitive by the containment relation of the surfaces delimited by the subtrees in $FI$.

**Property B**: For every three vertices $u,v,w$ of $G$, if $u \to v \in E2$ and $v \to w \in E1$ then $u \to w \in E1$, since $u \to v \in E2$ implies $t(u) \subseteq t(v)$ while $v \to w \in E1$ implies $t(v) \cap t(w)=\emptyset$, thus $t(u) \cap t(w)=\emptyset$. 

Methods for constructing subtree filaments are described in [2,5,7]. In fact, $G(V,E)$ is obtained by deleting from $GI(V,F)$ a transitive edge set $E_2$ between vertices $u,v$ having $t(u) \subset t(v)$. Consider two subtrees $t(v), t(w)$ (each with at least two vertices) whose intersection is only one vertex $X$ of $D$. Then, $a(v) \cap a(w) \neq \phi$ and for every subtree $t(u)$ having $t(u) \subset t(v)$ and $t(u) \cap t(v) \cap t(w) = \{X\}$ we must have $a(u) \cap a(v) \cap a(w) \neq \phi$; thus we cannot include the edge $u,v$ in $E_2$. Thus, for properly defining $E_2$, we must request either that the intersection of every two subtrees $t(u), t(v)$ contains an edge (every two rows of $M_{GI}$ have common ones in at least two rows) as in [2,5,7], or that if $|t(v) \cap t(w)| = 1$, then for every $t(u) \subset t(v)$ we have $a(u) \cap a(v) \neq \phi$.

We can define subtree filaments for any graph $D$, similarly to the case for $D$ planar, by assuming that edge intersections (not at their $D$ vertices) of subtrees $t(u), t(v)$ is not an intersection of the subtrees and of the surfaces $PP(t(u)), PP(t(v))$. When $D$ is not planar, we can get a better picture of the intersections of subtree filaments by considering every subtree filament $a(v)$ with maximal $t(v)$, redrawn separately without edge intersections, and all the filaments $t(u)$ having $t(u) \subset t(v)$, which are in the surface $PP(t(v))$. The graph $GI$ is called the base graph of $G$.

The family of 2-acyclic-subtree filament graphs are contained in the family of complements of $H$-mixed graphs defined in [5,7] and contains the families of interval filament graphs, subtree filament graphs on a tree, circular-arc filament graphs and cactus-subtree filament graphs, which are intersection graphs of 2-acyclic-subtree filaments on a line, a tree, a circle or a cactus, respectively [5,7,8]. The subtree filament graphs and the interval filament graphs were introduced by Gavril [5] and he proved that they contain the families of polygon-circle, cocomparability and chordal graphs [3,14]. As proved in [5], when the base graph $GI$ has a polynomial time algorithm to find a maximum independent set, $G$ also has a polynomial time algorithm to find a maximum independent set.

The family of subtree filament graphs on a tree has polynomial time algorithms for maximum weight cliques, independent sets, induced bicliques, induced split graphs, holes and antiholes of given parity [5,7,8,9] and for maximum weight induced matchings [1].

We point out that the algorithms and proofs in this paper do not require an intersection representation by subtree filaments, but only the representation $FI$ constructed from the matrix $M$ of $GI$ [10], and the transitive set of edges $E_2$. Intersection graphs are of interest in various domains such as computer science, genetics and ecology [10,11,17,18].
A graph is a complete bipartite graph if its vertex set has a partition into two independent sets such that every two vertices in different independent sets are adjacent. The complement of a complete bipartite graph is a biclique. Gavril [9] describes polynomial time algorithms for maximum induced bicliques in the following families of graphs: polygon-circle graphs, 4-hole-free graphs, complements of interval-filament graphs and complements of subtree-filament graphs. These problems have applications when a given set of entities related by some property, must be clustered into cliques or independent sets by some strongly connected vs. non-connected or similarity vs. dissimilarity criteria; for example in Protein-Protein-Interaction (PPI) problems for co-clustering proteins according to some criterion [13,15,16,19].

In the present paper we describe polynomial time algorithms for various problems on 2-acyclic subtree filament graphs $G$ when their base graphs $GI$ have algorithms for specific related problems. In Section 2 we describe an algorithm to find a maximum induced complete bipartite subgraph when $GI$ is $K_{2,m+1}$-free and has a polynomial time algorithm to find a maximum independent set, for example when $GI$ is perfect and $K_{2,m+1}$-free. Note that the perfect $K_{2,2}$-free graphs are exactly the $(C_4,\text{odd hole})$-free graphs. In Section 3 we describe an algorithm for maximum weight holes of a given parity when $GI$ has an algorithm for maximum weight holes of given parity; such families of graphs $GI$ are described in [6]. In Section 4 we describe an algorithm for minimum dominating holes, when $GI$ has an algorithm for minimum dominating holes. In Section 5 we describe an algorithm for maximum weight antiholes of a given parity when $GI$ is a Helly 2-acyclic-subtree graph. In Section 6 we describe various algorithms when $GI$ fulfills that for every maximal clique $C$ of $GI$ there are two points $X,Y$ in $D$ such that $C \subseteq V_{XY}$; this family of graphs includes the circular-arc filament graphs and the cactus subtree filament graphs [8].

2 Algorithms for maximum induced complete bipartite subgraphs and bicliques

First we describe an algorithm to find a maximum induced complete bipartite subgraph in a 2-acyclic-subtree filament graph $G(V,E)$ when its base graph $GI$ is $K_{2,m+1}$-free and $G$ (thus also $G$ [5]) has a polynomial time algorithm to find a maximum independent set. Note that $GI$ is $K_{2,m+1}$-free iff for every two non-adjacent vertices $u,v$, the maximum independent set of $GI(N_G(u) \cap N_G(v))$ has at most $m$ vertices. Consider two non-adjacent vertices $u,v$ having $t(u) \subseteq t(v)$ and let $U_{u,v} = \{w \mid t(u) \subseteq t(w) \subseteq t(v), w \not\in N_G(u) \cup N_G(v)\} \cup \{u,v\}$.

**Lemma 1.** Every vertex $z \in N_G(u) \cap N_G(v)$ is adjacent to every vertex $w \in U_{u,v}$. 
Proof. Assume that there are two non-adjacent vertices \( z \in N_G(u) \cap N_G(v) \) and \( w \in U_{u,v} \). If \( t(w) \subset t(z) \) then \( t(u) \subset t(w) \subset t(z) \) and if \( t(z) \subset t(w) \) then \( t(z) \subset t(w) \subset t(v) \). Also, if \( t(w) \cap t(z) = \emptyset \), then \( t(u) \cap t(z) = \emptyset \), since \( t(u) \subset t(w) \). All three cases contradict the fact that \( z \) is adjacent to both \( u \) and \( v \) but not to \( w \). □

Let \( B(IND1, IND2, E) \) be a complete bipartite subgraph of \( G \):

Assume that \( IND1 \) has only one vertex \( u \); then \( IND2 \subseteq N_G(u) \).

Assume that the clique \( coG(IND1) \) has only \( E2 \) edges, that is every two vertices \( w,z \in IND1 \) have \( t(w) \subset t(z) \); let \( u,v \) be the unique source and sink of \( coG(IND1) \). Then, by Lemma 1, \( IND1 \subseteq U_{u,v} \) and \( IND2 \subseteq N_G(u) \cap N_G(v) \).

Assume that \( IND1 \) has two vertices \( u,v \) having \( t(u) \cap t(v) = \emptyset \); hence \( IND2 \subseteq N_G(u) \cap N_G(v) \). The clique \( coG(IND2) \) may contain both \( E1 \) and \( E2 \) edges. Let \( SO \) and \( SI \) be the set of sources and sinks, respectively, of \( coG(IND2) \) relative to \( E2 \). Each one of \( coG(SO) \) and \( coG(SI) \) contains no \( E2 \) edges and each one of \( G(SO) \) and \( G(SI) \) contains no \( E1 \) edges. Thus, \( SO \) and \( SI \) are independent sets of \( GI(N_G(u) \cap N_G(v)) \), and since \( GI \) is \( K_{2,m+1} \)-free it follows that each one of \( SO \) and \( SI \) has at most \( m \) vertices. Therefore, \( IND2 \) is defined by a set of sources \( SO \), \( |SO| \leq m \), and a set of sinks \( SI \), \( |SI| \leq m \). Let \( U_{SO,SI} = \{ w \mid w \in N_G[S(O \cup SI)], t(u) \subset t(w) \subset t(v), u \in SO, v \in SI \} \), \( NN_G(SO,SI) = \cap_{u \in SO,SI} N_G(u) \). Then, by Lemma 1, \( IND2 \subseteq (SO \cup SI) \subseteq U_{SO,SI} \) and \( IND1 \subseteq NN_G(SO,SI) \).

The algorithm for a maximum induced complete bipartite subgraph of \( G \) works as follows: First, it considers every vertex \( u \), sets \( IND1 = \{ u \} \), and finds a maximum independent set \( IND2 \) in \( G(N_G(u), E) \) by the algorithm in [5]. Next, it considers every possible (independent) sets of sources and sinks \( SO, SI \), each with at most \( m \) vertices, finds a maximum independent set \( IND3 \) in \( G(U_{SO,SI}, E) \) by the algorithm in [5], sets \( IND2 = (SO \cup SI) \cup IND3 \) and finds a maximum independent set \( IND1 \) in \( G(\cap_{SO,SI} N_G(SO,SI)) \). Among all such pairs \( IND1, IND2 \) it chooses the pair with a maximum number of vertices.

For every clique \( C \) of \( G \), denote the subgraph \( \cup_{v \in C} t(v) \) of \( D \) by \( t(C) \). We consider now the problem of finding maximum bicliques in 2-acyclic-subtree filament graphs \( G(V,E) \) for which there exists a constant \( k \) such that every clique \( C \) of \( G \) has a subset \( A_C \) with at most \( k \) vertices fulfilling: \( t(C) = t(A_C) \) and there are no edges of \( G \) between a vertex \( u \) having \( t(u) \subset t(C) \) and a vertex \( v \) having \( t(A_C) \subset t(v) \) and \( A_C \cap N_G[v] = \emptyset \). For a biclique \( B(C1,C2) \), either \( t(C1) = t(A_C1) \subset t(C2) \) or \( t(C1) \cap t(C2) = \emptyset \). For such a biclique \( B(C1,C2) \), we look for the clique \( C1 \). Thus, for every clique \( A \) (candidate for \( A_{C1} \)) with at most \( k \) vertices let \( Z_A = \{ u \mid t(u) \subset t(A) \} \) and \( W_A = \{ w \mid A \cap N_G[w] = \emptyset \) and \( t(A) \subset t(w) \) or \( t(A) \cap t(w) = \emptyset \);
hence $C_1 \subseteq Z_A$ and $C_2 \subseteq W_A$. In the subtree-filament subgraphs $G(Z_A)$, $G(W_A)$, we can find maximum cliques, by the algorithm in [5]. Among them we take the biclique with a maximum number of vertices.

### 3 ALGORITHM FOR MAXIMUM WEIGHT HOLES OF GIVEN PARITY

In this Section we describe an algorithm for maximum weight holes of a given parity in 2-acyclic subtree-filament graphs $G(V,E)$, when its base graph $GI(V,F)$ has an algorithm for maximum weight holes of given parity; such families of graphs $GI$ are described in [6]. The algorithm is similar to the one in [7] described for 3D-interval-filament graphs. Recall that $E \cup E_2 = coE_1 = F$.

**Lemma 2.** Let $p(v_i,v_k)$ be an induced path in $G$ and consider some $i$, $1 \leq i \leq k-2$. If $v_i \rightarrow v_k \in E_2$, then for every $v_j$, $1 \leq j \leq k-2$, we have $v_j \rightarrow v_k \in E_2$.

**Proof.** Assume that $v_i \rightarrow v_k \in E_2$. By Property B, we cannot have $v_{i+1} \rightarrow v_k \in E_1$, since this would imply $v_i \rightarrow v_{i+1} \in E_1$ and $v_i \rightarrow v_{i+1} \in E$. Thus, $v_{i+1} \rightarrow v_k \in E_2$; and so on, for every vertex $v_j$, $1 \leq j \leq k-2$, to the right and left of $v_i$ in $p$. □

**Lemma 3.** Any hole $h(v_i,v_k)$ of $G$, which is not a hole of $GI$, has two non-adjacent vertices $v_i,v_j$ such that $v_i \rightarrow v_j \in E_2$ and has no three vertices $v_i,v_j,v_k$ such that $v_i \rightarrow v_j,v_j \rightarrow v_k \in E_2$.

**Proof.** A hole $h(v_i,v_k)$ of $G$ has all its edges in $E \subseteq coE_1$. If $h$ is not a hole of $GI$, then $h$ has in $GI$ a chord in $E_2 \subseteq coE$, since $h$ is a hole of $G$. Thus, $h$ has two non-adjacent vertices $v_i,v_j$ such that $v_i \rightarrow v_j \in E_2$. Assume that $h$ has three vertices $v_i,v_j,v_k$ such that $v_i \rightarrow v_j,v_j \rightarrow v_k \in E_2$; w.l.o.g. assume that $i<k<j$. But, Lemma 2 applied to the path $v_i \rightarrow v_j \in E_2$ and the path $p(v_i,v_j)$ implies that for every $v_r \in p(v_i,v_j)$, we have $v_r \rightarrow v_j \in E_2$, hence $v_k \rightarrow v_j \in E_2$, contradicting $v_j \rightarrow v_k \in E_2$. □

**Lemma 4.** Consider a hole $h(v_i,v_k)$ of $G$. Then, either $h$ is a hole of $GI$ or $h$ has a vertex $v_i$ such that for every $v_j \in h(v_{i+2},v_{i-2})$, we have $v_j \rightarrow v_i \in E_2$ and $h(v_{i+2},v_{i-2})$ is an induced path of $GI$.

**Proof.** Assume that $h$ is not a hole of $GI$; hence, by Lemma 3, $h$ has two non-adjacent vertices $v_i,v_r$ such that $v_r \rightarrow v_i \in E_2$. By Lemma 2 applied to the path $h(v_{i+2},v_{i-2})$, for every $v_j \in h(v_{i+2},v_{i-2})$, we have $v_j \rightarrow v_i \in E_2$. By Lemma 3, there are no $E_2$ edges between non-adjacent vertices in $h(v_{i+2},v_{i-2})$, thus $h(v_{i+2},v_{i-2})$ is an induced path of $GI$. □

**Lemma 5.** A hole $h$ of $GI$ has no $E_2$ edges of $H_0(V,E_2)$ and is a hole of $G$. An induced path $p(v_i,v_k)$ of $GI$ can have only its first and last edges in $E_2$: $v_i \rightarrow v_2,v_k \rightarrow v_{k-1} \in E_2$. 

Technion - Computer Science Department - Technical Report CS-2011-12 - 2011
**Proof.** Let \( h(v_1, v_k) \) be a hole of \( GI \) and assume \( v_1 \rightarrow v_2 \in E_2 \). Since \( v_2, v_k \) are not adjacent in \( GI \) it follows that \( v_2 \rightarrow v \in E_1 \), and by Property B, it follows that \( v_1 \rightarrow v_k \in E_1 \), contradicting the fact that \( h(v_1, v_k) \) is a hole of \( GI \). Therefore \( h \) has no \( E_2 \) edges and is a hole of \( G \).

Let \( p(v_1, v_k) \) be an induced path of \( GI \) and assume that \( v_i \rightarrow v_{i+1} \in E_2 \) for some \( i, 2 \leq i \leq k-1 \). Since \( v_{i-1}, v_{i+1} \) are not adjacent in \( GI \) it follows that \( v_{i-1} \rightarrow v_{i+1} \in E_1 \); hence, by Property B, it follows that \( v_i \rightarrow v_{i+1} \in E_1 \), contradicting the fact that \( v_i \rightarrow v_{i+1} \in p(v_1, v_k) \subset E_1 \). Similarly, if \( v_{i+1} \rightarrow v_i \in E_2, 1 \leq i \leq k-2 \). Therefore \( p(v_1, v_k) \) can have only its first and last edges in \( E_2 \), with the orientation \( v_1 \rightarrow v_2, v_k \rightarrow v_{k-1} \in E_2 \).

Assume that \( GI \) has a polynomial time algorithm to find a maximum weight hole of a given parity. The algorithm to find a maximum weight hole \( h(v_1, v_k) \) of a given parity in \( G \), works as follows: By Lemmas 4,5, either \( h \) is a hole of \( GI \), to be found directly, or \( h \) has a vertex \( v_i \) such that for every \( v_j \in h(v_{i+2}, v_{i-2}) \), \( v_j \rightarrow v_i \in E_2 \) and \( h(v_{i+2}, v_{i-2}) \) is an induced path of \( GI \). We take every vertex of \( G \) as candidate for \( v_i \) of \( h \). We take every two non-adjacent vertices in \( NG(v_i) \) as candidates for \( v_{i-1}, v_{i+1} \). We take as candidates for \( v_{i-2}, v_{i+2} \) every \( v \in ING(v_i) \cap NG(v_{i-1}) \), \( w \in ING(v_i) \cap NG(v_{i+1}) \) such that either \( u, w \) are adjacent in \( G(V, E) \) or \( u \rightarrow w \in E_1 \), that is \( t(u) \cap t(w) = \emptyset \). The vertex \( v_i \) is represented by a tree \( t(v_i) \) while \( t(v_j) \cap t(v_{i+1}) \), \( t(v_j) \cap t(v_{i-1}) \) and every \( t(v) \) with \( v \in ING(v_i) \), are subtrees of \( t(v_i) \). The intersection graph of the family of subtree filaments \( \{ t(v_j) \cap t(v_{i-1}), t(v_j) \cap t(v_{i+1}) \} \cup \{ v \in ING(v_i) \} \) of the subtree \( t(v_i) \) is \( G((NG(v_i) \cup \{ v_{i-1}, v_{i+1} \}), E) \) and its base graph is \( GI((NG(v_i) \cup \{ v_{i-1}, v_{i+1} \}), coE_1) \). Hence an induced path from \( v_{i-2} \) to \( v_{i+2} \) in \( t(v_j) \) is an induced path of the interval graph defined by the intersection of the subtrees in \( GI((NG(v_i), coE_1) \) with the unique path connecting \( t(v_{i-2}) \) to \( t(v_{i+2}) \) in \( t(v_i) \) such that its first and last edges are in \( GI \) and not in \( E_2 \). A maximum weight induced path of a given parity in an interval graph can be found by the algorithm in [6]. The algorithm works in time \( O(|V|^2 + |V|^c) \) where \( O(|V|^c) \) is the time needed to find a maximum weight hole of a given parity in \( GI \).

**4 Algorithm for Minimum Dominating Holes**

In the present Section we describe an algorithm for finding minimum dominating holes in 2-acyclic subtree-filament graphs \( G(V, E) \), when the base graph \( GI \) has an algorithm for minimum dominating holes. The algorithm is similar to the one in [8] for 3D-interval-filament graphs. Using Lemmas 4,5, we characterize dominating holes:
Lemma 6. Consider a dominating hole $h$ of $G$. Then, either $h$ is a hole of $GI$ which dominates $G$ or $h$ has a vertex $v_i$ such that $\{v_{i-1}, v_{i+1}\}$ dominates $GI(\text{OUT}_G(v_i))$ and $h$ is a dominating hole of the intersection graph of the family of subtree filaments $\{t(v_i), t(v_i) \cap t(v_{i+1}), t(v_i) \cap t(v_{i+1}) \cup \{t(u) \mid v \in \text{IN}_G(v_i)\}\}$ of the subtree $t(v_i)$.

Proof. Consider a dominating hole $h$ of $G$. By Lemma 4, $h$ is a hole of $GI$, or $h$ has a vertex $v_i$ such that $h-\{v_i, v_{i-1}, v_{i+1}\}$ is an induced path of $GI(\text{IN}_G(v_i))$ with its first and last edges are in $GI$ and not in $E_2$. In the first case, the hole $h$ of $GI$ dominates $G$. In the second case, $h-\{v_i, v_{i-1}, v_{i+1}\}$ cannot dominate vertices in $\text{OUT}_G(v_i)$, thus, the vertices in $\text{OUT}_G(v_i) \cup \text{IN}_G(v_i)$ are dominated by $\{v_i, v_{i-1}, v_{i+1}\}$. Therefore, $h$ is a dominating hole containing $v_i$ of the intersection graph of the family of subtrees filaments $\{t(v_i), t(v_i) \cap t(v_{i+1}), t(v_i) \cap t(v_{i+1}) \cup \{t(u) \mid v \in \text{IN}_G(v_i)\}\}$ of the subtree $t(v_i)$. □

The algorithm to find a minimum dominating hole $h$ in a 2-acyclic-subtree filament graph $G$ works as follows: We find a minimum dominating hole in $GI$. For every vertex $v$ of $G$ and every two non-adjacent vertices $u, w$ in $\text{IN}_G(v)$ such that $\{v, u, w\}$ dominates $\text{OUT}_G(v) \cup \text{IN}_G(v)$, we find a minimum dominating hole, containing $v, u, w$ in the intersection graph of the family of subtrees $\{t(v), t(v) \cap t(u), t(v) \cap t(w) \cup \{t(x) \mid x \in \text{IN}_G(v)\}\}$ of the subtree $t(v)$, by the algorithm described in [8]; among these dominating holes we take the minimum one. The algorithm works in time $O(\sqrt{|V|}^7 + |V|^c)$ where $O(\sqrt{|V|}^c)$ is the time needed to find a minimum dominating hole in $GI$.

5 Antiholes of Given Parity in 2-Acyclic-Subtree Filament Graphs when $GI$ is Helly

In the present Section we describe an algorithm for maximum weight antiholes of a given parity in 2-acyclic-subtree filament graphs $G(V,E)$ whose base graph $GI$ is a Helly 2-acyclic-subtree graph. The algorithm is similar to the one in [7] described for subtree-filament graphs on a tree.

The problem is to find a hole $h(v_i, v_k)$ of a given parity in the complement $\text{co}(G, E1 \cup E2)$ of $G(V,E)$. If $h(v_i, v_k)$ contains no $E1$ edges, then $h(v_i, v_k)$ is a hole of the comparability graph $H_G(V,E2)$ which being perfect, has only even holes.

Assume now that $h$ has an $E1$ edge $(v_i, v_k)$, hence $v_i, v_k$ are not adjacent in $G(V,E)$. Thus, the path of $h$ connecting $v_i, v_k$ is contained in $\text{IN}_G(v_i) \cap \text{IN}_G(v_k)$. By [Lemma 2, 4] the connected components of $GI(\text{IN}_G(v_i) \cap \text{IN}_G(v_k))$ are cliques (say $m$ cliques) and the path of $h$ connecting $v_i, v_k$ is a path in $\text{co}(G(\text{IN}_G(v_i) \cap \text{IN}_G(v_k)), E1 \cup E2)$. The path is formed by $m$ subpaths
in the $E_2$ comparability graphs defined by every clique in $GI(N_G(v_1)\cap N_G(v_k))$ and $m-1$ $E_1$ edges (between the comparability graphs defined by different cliques) connecting these subpaths. Thus, it remains to find induced paths of given parity in comparability graphs.

The algorithm to find an induced path of a given parity from a vertex $v_1$ to any other vertex in a comparability graph $H(V,F)$ oriented as a transitive graph, works as follows: The directions of the edges in an induced path of $H(V,F)$ are alternating, otherwise a chord appears in the path. We label with path parity even every vertex $v_2$ adjacent to $v_1$ by an edge $(v_1,v_2)$ or $(v_2,v_1)$ and insert a pointer from $v_2$ to $v_1$. We continue labeling the vertices not labeled with both parities. Assume that we found an induced path $p$ of a given parity from $v_1$ to a vertex $v$ and we have pointers from $v$ along $p$ to $v_1$, to recover $p$. If the last edge on $p$ is $(x,v)$, we label every $w$ having $(w,v)$ with the parity of $|p|+1$. If the last edge on $p$ is $(v,x)$, we label every $w$ having $(v,w)$ with the parity of $|p|+1$. From $v$ we backtrack on $p$ to find the vertex $u$ closest to $v_1$ and adjacent to $w$. In the first case, the edge between $u$ and $w$ is oriented $(w,u)$, because $(u,w)$ and $(w,v)$ would imply $(u,v)$. Similarly, in the second case, the edge between $u$ and $w$ is oriented $(u,w)$. Thus, the path $q$ obtained by going on $p$ from $v_1$ to $u$ and then directly to $w$ has the same parity as $|p|+1$, since both end in backward edges or in forward edges; we insert a pointer from $w$ to $u$. The algorithm requires $O(|V|^4)$ time.

This entire algorithm for antiholes requires $O((\max \ m)|V|^8)$ time. Note that finding a maximum induced path in a comparability (even a bipartite) graph is NP-hard.

### 6 Algorithms in 2-acyclic-subtree filament graphs $G$ fulfilling that every maximal clique of $GI$ is contained in some $V_{XY}$

In this Section we describe various algorithms for 2-acyclic-subtree filament graphs $G(V,E)$ fulfilling that for every maximal clique $C$ of $GI$ there are two points $X,Y$ in $D$ such that $C \subseteq V_{XY}$. This family of graphs includes the circular-arc filament graphs and the cactus subtree filament graphs [8]. They include also the Helly 2-acyclic-subtree filament graphs, since every clique $C$ of $GI$ is contained in some $V_X$. The algorithm is a generalization of the algorithm in [8].

Consider two vertices $X,Y$ of $D$. For every two non-adjacent vertices $v,u$ of $G$ such that $v \in V_X \cap V_{XY}$, $u \in V_Y \cap V_{XY}$, we orient the $E_1$ edge $v \rightarrow u$ as $v \rightarrow u$. In addition, we reverse the orientation of the edges in $coG(V_Y \cap V_{XY}, E_2)$; by reversing the orientation of the edges in a transitive graph we obtain a transitive orientation, since if an obstruction to transitivity
occurs, its reversing gives an obstruction in the original orientation. Let $E3$ denote the new
orientation of $coG(V_{XY}, E1 \cup E2)$.

**Lemma 7.** The new orientation $E3$ of $coG(V_{XY})$ is transitive.

**Proof.** In $E3$ consider three vertices $v, u, w$ having $v \rightarrow u \rightarrow w$. Assume that $u \in V_{XY} \cap V_{XY}$; then $v \in V_{XY} \cap V_{XY}$ and $t(v) \subset t(u)$. If $w \in V_{XY} \cap V_{XY}$, then $v \rightarrow w$ by the orientation of $E2$. If $w \in V_{XY} \cap V_{XY}$, then $t(u) \cap t(w) = \phi$ implying that $t(v) \cap t(w) = \phi$, $v \rightarrow w \in E1$ and $v \rightarrow w \in E3$.

Assume now that $u \in V_{XY} \cap V_{XY}$; then $w \in V_{XY} \cap V_{XY}$ and $t(w) \subset t(u)$. If $v \in V_{XY} \cap V_{XY}$, then $w \rightarrow v$ by the transitivity of the reverse of $E2$. If $v \in V_{XY} \cap V_{XY}$, then $t(v) \cap t(u) = \phi$ implying that $t(v) \cap t(w) = \phi$, $v \rightarrow w \in E1$ and $v \rightarrow w \in E3$. □

Therefore, every graph $G(V,E)$ in this family has $O(|V|^2)$ cocomparability subgraphs $G(V_{XY},E)$, which can be found in polynomial time, such that every clique of $G$ is contained in one of them. Thus, a maximum clique of $G$ can be found in polynomial time by finding maximum cliques in every cocomparability subgraph $G(V_{XY},E)$. When the family GI has also an algorithm for maximum independent set, the algorithms described in [8] for maximum induced split graph and for clique intersecting all maximum independent sets, work as well for this family of graphs.

**References**