DIMACS Technical Report 2004-30
June 2004

k-Interval-filament graphs

by
Fanica GAVRIL[1]
DIMACS
Rutgers University
96 Frelinghuysen Rd.
Piscataway, NJ 08854, USA

[1]Permanent address: CEMA(T1), P.O.B. 2250, Haifa 31021, ISRAEL

ABSTRACT

For a fixed k, an oriented graph is *k-transitive* if it is acyclic and for every directed path $p=u_1\rightarrow u_2\rightarrow\ldots\rightarrow u_{k+2}$ with $k+2$ vertices, p induces a clique if each of the two subpaths $u_1\rightarrow u_2\rightarrow\ldots\rightarrow u_{k+1}$ and $u_2\rightarrow\ldots\rightarrow u_{k+2}$ induces a clique. We describe an algorithm to find a maximum weight clique in a k-transitive graph.

Consider a hereditary family ***G*** of graphs. A graph $H(V,E)$ is called ***G****-k-mixed* if its edge set can be partitioned into two disjoint subsets $E1$, $E2$-$E3$ such that $H(V,E1)\in\boldsymbol{G}$, $H(V,E2)$ is transitive, $H(V,E2\text{-}E3)$ is k-transitive and for every three distinct vertices u,v,w if $u\rightarrow v\in E2$ and $(v,w)\in E1$ then $(u,w)\in E1$. The letter ***G*** is generic and can be replaced by names of specific families. We show that if the family ***G*** has a polynomial time algorithm to find a maximum clique, then, under certain restrictions there exists a polynomial time algorithm to find a maximum clique in a family of ***G***-k-mixed graphs.

Let I be a family of intervals on a line L in a plane PL such that every two intersecting intervals have a common segment. In PL, above L, construct to each interval $i(v)\in I$ a filament v connecting its two endpoints, such that for every two filaments u,v having $u\cap v\neq\phi$ and disjoint intervals $i(u)<i(v)$, there are no k filaments w with $i(u)<i(w)<i(v)$ which intersect neither u nor v, are mutually disjoint and have mutually disjoint intervals. This is a *family of k-interval-filaments* and its intersection graph is a *k-interval-filament graph*; their complements are (k-transitive) mixed graphs and have a polynomial time algorithm for maximum cliques. Now, when two filaments u,v do not intersect because $u\subset v$, and between u and v there are at most k-1 non-intersecting filaments $w_1,\ldots,w_{k-1}$ such that $w_i\subset w_{i+1}$ and intersect neither u nor v, we attach to each one of u,v a branch in the space above PL such that the two branches intersect. This is a *family of general-k-interval-filaments* and its intersection graph is a *general-k-interval-filament graph*; their complements are (k-transitive)-k-mixed graphs.

1. INTRODUCTION

We consider only finite graphs $G(V,E)$ with no parallel edges and no loops, where V is the set of vertices and E the set of edges. For $U \subseteq V$, $G(U)$ is the vertex subgraph defined by U. For $F \subseteq E$, $G(F)$ is the edge subgraph defined by F on V. Two vertices connected by an edge u—v are called *adjacent* and we denote this by (u,v), without regard for the orientation of the edge; $coG(V,coE)$ is *the complement of* G where $coE=\{(u,v)|\ u \neq v, (u,v) \notin E\}$. For $E1 \subseteq coE$ we denote $co[coG(E1)]$ by $G(coE1)$. A directed edge from u to v is denoted $u \rightarrow v$. We also denote $N_G(v)=\{u|\ (u,v) \in E\}$ and $N_G[v]=N_G(v) \cup \{v\}$. By a path $p=v_1$—...—v_k we always mean a simple path, denoted $p(v_1,v_k)$; p is an *induced path* if it has no chords. A *hole* $h=v_1$—...—v_k—v_1 is a chordless cycle with four or more vertices, denoted $h(v_1,v_k)$; h is *a dominating hole* if every vertex in V-h is adjacent to a vertex in h. A *subpath* of h, clockwise from v_i to v_j, is denoted $h(v_i,v_j)$. A subset of V is a *clique* (an *independent set*) if every two of its vertices are adjacent (not adjacent, respectively). Two subsets of a set *intersect* if they have a non-empty intersection; two subsets *overlap* if they intersect but none is contained in the other. In a rooted tree, with edges oriented from sons to father, the distance between two vertices u,v, where u is a successor of v, is the number of vertices in the directed path from u to v. The depth of a tree is the maximum distance from leaf to root.

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. Intersection graphs are of interest in various domains such as computer science, genetics and ecology [RO]. Intersection graphs of intervals on a line, subtrees on a tree and arcs on a circle are called *interval, chordal* and *circular-arc graphs* [GA3,GA2], respectively. An oriented graph $G(V,E)$ is called *transitive* if it is acyclic and for every three vertices $u,v,w \in V$, $u \rightarrow v, v \rightarrow w \in E$ implies $u \rightarrow w \in E$ [GRU,EPL]; its underlying undirected graph is called a *comparability graph*. A transitive graph can also be defined as follows: an oriented graph $G(V,E)$ is transitive if it is acyclic and for every directed path with three vertices $p=u_1 \rightarrow u_2 \rightarrow u_3$, p induces a clique if each of the two subpaths $u_1 \rightarrow u_2$, $u_2 \rightarrow u_3$ induces a clique. We define a generalization of the transitive graphs: For a fixed k, an oriented graph is *k-transitive* if it is acyclic and for every directed path $p=u_1 \rightarrow u_2 \rightarrow \ldots \rightarrow u_{k+2}$ with $k+2$ vertices, p induces a clique if each of the two subpaths $u_1 \rightarrow u_2 \rightarrow \ldots \rightarrow u_{k+1}$ and $u_2 \rightarrow \ldots \rightarrow u_{k+2}$

induces a clique. Thus, a graph is transitive iff it is *1*-transitive. Transitive graphs are perfect, but *k*-transitive graphs with $k\geq 2$ may not be perfect since odd holes are *k*-transitive.

A family ***H*** of graphs is *hereditary* if $H(V,E)\in\boldsymbol{H}$ implies $H(U)\in\boldsymbol{H}$ for every $U\subseteq V$; the families of intersection graphs are hereditary. We denote $\boldsymbol{coH}=\{coH|\ H\in\boldsymbol{H}\}$. Consider a hereditary family ***H*** of graphs. A graph *H(V,E)* is called ***H**-k-mixed* if its edge set can be partitioned into two disjoint subsets *E1*, *E2-E3* such that $H(V,E1)\in\boldsymbol{H}$, *H(V,E2)* is transitive, *H(V,E2-E3)* is *k*-transitive and for every three distinct vertices *u,v,w* if $u\rightarrow v\in E2$ and $(v,w)\in E1$ then $(u,w)\in E1$. The letter ***H*** is generic and can be replaced by names of specific families. When $E3=\phi$ we obtain the ***G***-mixed graphs defined in [GA4]. A vertex *u* is successor (predecessor) of *v* if $u\rightarrow v\in E2$ ($v\rightarrow u\in E2$, respectively). A vertex is an *E2-sink* if it has no outgoing *E2* edges.

Gavril [GA4] defined a new family of intersection graphs: Let *I* be a family of intervals on a line *L*, every two intersecting intervals having a common segment; we denote by *l(i),r(i)* the endpoints of an interval *i*. Let $V=\{v|\ i(v)\in I\}$ be a vertex set. In the plane *PL* containing *L*, above *L*, we construct to each interval $i(v)\in I$ a filament (a union of curves) *a(v)* connecting *l(i(v)),r(i(v))*, bounded by the perpendiculars to *L* at *l(i(v)),r(i(v))*, and not containing its endpoints. A filament *v* can have more than two endpoints subject to the conditions that it does not contain its endpoints, is continuous and all endpoints are in *L*; a filament can be a piece of (contained in) another filament. $FI=\{v|\ i(v)\in I\}$ is *a family of 2D-interval-filaments* and its intersection graph is a *2D-interval-filament graph*. Clearly, if two intervals are disjoint, their filaments do not intersect. The family of complements of *2D*-interval-filament graphs is exactly the family of cointerval mixed graphs [GA4]. The family of *2D*-interval-filament graphs contains the family of polygon-circle graphs and the latter includes the circular-arc, the circle trapezoid, the circle and the chordal graphs [GA2,FMW,GA1,GA3].

Consider a tree *T* in a plane *PL* and a family *ST* of subtrees of *T*. To each vertex *x* of *T* we add a branch *(x,y)*, adding it also to every subtree in *ST* containing *x*; in this way, the intersection of every two subtrees in *ST* contains an edge of *T*. Let *PP* be a surface perpendicular to *PL* such that $PP\cap PL=T$. On *PP*, above *T*, consider a family $FT=\{f_t|\ t\in ST\}$ of filaments, called *subtree filaments*, each f_t contained in *PP(t)* and connecting all the endpoints of *t*, such that if two subtrees overlap, their filaments intersect, and if two subtrees are contained one in another, their filaments may or may not intersect. Clearly, if two subtrees are disjoint, their filaments do not intersect. The intersection graph *G(V,E)* of

FT is *a subtree filament graph*. The family of complements of subtree-filament graphs is exactly the family of cochordal mixed graphs [GA4]. The subtree filaments are *3D*-interval filaments [GA7].

Gavril [GA4] described a polynomial time algorithm to find maximum weight cliques in ***H*** mixed graphs when ***H*** has such an algorithm. Other polynomial time algorithms in *2D*-interval-filament and subtree-filament graphs were given for finding maximum weight cliques [GA4], maximum weight induced paths [GA5], maximum weight induced matchings [CA], and holes and antiholes of given parity [GA6].

In Section 2 we present an algorithm to find maximum weight cliques in *k*-transitive graphs and prove that the complements of the *k-interval filament* graphs (to be defined) are *k*-transitive mixed graphs. In Section 3 we define various other families of *k*-filament graphs. In Section 4 we show that given a family ***H*** of ***G***-*k*-mixed graphs where the family ***G*** has a polynomial time algorithm to find a maximum clique, there exists, under certain restrictions, a polynomial time algorithm to find a maximum clique in the family ***H*** of ***G***-*k*-mixed graphs, thus also in a family of (***G***-*k*-mixed)-*k*-mixed graphs.

In Sections 5,6 we describe polynomial time algorithms to find holes of a given parity and minimum dominating holes in *k*-interval-filament graphs and subtree filament graphs. When we discuss an ***H*** mixed graph $H(V,E1,E2)$, we assume that the partition $E1,E2$ is given, and when $H(V,E1)$ is the complement of an intersection graph, we assume that the intersection representation is also given.

2. ALGORITHMS FOR MAXIMUM WEIGHT CLIQUES IN *k*-TRANSITIVE AND *k*-INTERVAL-FILAMENT GRAPHS

Consider a transitive, i.e., a *1*-transitive, graph $G(V,E)$ and let $v_1,...,v_n$ be a topological ordering of its vertices; we denote $N^-(v)=\{u|\ u\rightarrow v\in E\}$. The algorithm to find a maximum weight clique in G works as follows: For every i we construct a maximum weight clique c_i in $G(V_i)$ such that $v_i\in c_i$. Assume that we have already $c_1,...,c_i$. Let us construct c_{i+1}. If $N^-(v_{i+1})=\phi$ then $c_{i+1}=\{v_{i+1}\}$. Assume that $N^-(v_{i+1})\neq\phi$. By the transitivity of G, for every $v_j\in N^-(v_{i+1})$, $c_j\cup\{v_{i+1}\}$ is a clique of $G(V_i)$. Thus, $c_{i+1}=c_j\cup\{v_{i+1}\}$ where c_j is a clique with maximum weight in $\{c_j|\ v_j\in N^-(v_{i+1})\}$. In this algorithm we remember with every c_i, its highest indexed vertex v_i, from which to continue to v_{i+1}. A similar algorithm works also for *k*-transitive graphs G. For every v_i and for every clique $\{v_{i,1},...,v_{i,k-1},v_i\}$ with k vertices in $G(V_i)$, we remember a maximum weight clique c of $G(V_i)$ whose k highest indexed vertices are $\{v_{i,1},\ldots,v_{i,k-1},v_i\}$. Then, by the definition of the k-transitive graphs, $c\cup\{v_{i+1}\}$ is a clique of $G(V_{i+1})$ iff $\{v_{i,1},\ldots,v_{i,k-1},v_i,v_{i+1}\}$ is a clique. Below is a formal description of the algorithm.

Consider a weighted k-transitive graph $G(V,E)$ and let $v_1,\ldots,v_n$ be a topological ordering of its vertices; for $U\subseteq V$ we denote $\omega(U)=\sum_{v\in U}\omega(v)$. For a clique c of G with $|c|\geq k$, we denote the set of its k highest indexed vertices $\{u_{|c|-k+1},\ldots,u_{|c|}\}$ by $tail(c)$. The algorithm for finding a maximum weight clique in G works by constructing for each v_i, $i=1,2,\ldots,n$, the set T_i of all triples $[b,pr(b),\Omega(b)]$ where: (i) b is a clique with k vertices in $G(V_i)$ such that $v_i\in b$; (ii) $\Omega(b)=\max\{\omega(c)|\ c$ *is a clique in* $G(V_i)$ *having* $|c|\geq k$ *and* $b=tail(c)\}$; (iii) $pr(b)$ is the highest indexed vertex in $c-b$, c being a clique giving the maximum to $\Omega(b)$;

In addition, with each v_i we keep a pair b_i,Ω_i defined as follows: $\Omega_i=\max\{\omega(c)|\ c$ *is a clique in* $G(V_i,E)$ *and* $v_i\in c\}$; let c be a clique for which $\Omega_i=\omega(c)$. If $|c|\geq k$ then $b_i=tail(c)$ and if $|c|<k$ then $b_i=c$. The algorithm for evaluating T_i,b_i,Ω_i for every i is given below.

1. ***for*** $i=1$ ***to*** n set $T_i=\phi$;
2. ***for*** $i=1$ ***to*** n ***do***
   $\Omega_i=0$;
3. let $C_i=\{b|\ b$ *is a clique of* $G(V_i),\ |b|=k,\ v_i\in b\}$;
4. ***for*** every $b\in C_i$
      let $\Omega(b)=\omega(b),\ pr(b)=\Lambda$;
      let v_r be the lowest indexed vertex in b;
5. ***for*** every $u\in N^-(v_i)\cap N^-(v_r)$ such that $b\cup\{u\}$ is a clique
         let $c=b\cup\{u\}-\{v_i\}$;
         let v_j be the highest indexed vertex in c;
6. find $[c,pr(c),\Omega(c)]$ in T_j;
7. ***if*** $\Omega(b)<\Omega(c)+\omega(v_i)$ ***then*** $\Omega(b)=\Omega(c)+\omega(v_i)$ and $pr(b)=u$;
8. add $[b,pr(b),\Omega(b)]$ to T_i;
9. ***if*** $\Omega_i<\Omega(b)$ ***then*** $\Omega_i=\Omega(b),\ b_i=b$;
10. ***for*** every maximal clique c of $G(V_i)$ fulfilling $|c|<k$, $v_i\in c$
      if $\Omega_i<\omega(c)$ ***then*** $\Omega_i=\omega(c)$ and $b_i=c$;
11. ***end***.

To find a maximum weight clique in $G(V,E)$, we find a vertex v_i having $\Omega_i = max_{1\leq j\leq n}\Omega_j$ and backtrack on the b_i's and the $pr(b_i)$'s.

Lemma 1. The algorithm finds a maximum weight clique in a k-transitive graph.

Proof. Let us first prove by induction that the sets T_i, b_i, Ω_i constructed by the algorithm are correct. Assume that this is true for $j<i$. In Steps 3-9 we consider every clique b of $G(V_i)$ such that $|b|=k$, $v_i \in b$, and every clique c in $G(N^-(v_i))$ such that $|c|=k$, $b=tail(c\cup\{v_i\})$; let v_j be the highest indexed vertex in c. In Step 6 we find $[c,pr(c),\Omega(c)]$ in T_j. Thus, by induction, $\Omega(c)$ is the weight of a maximum weight clique d in $G(V_j)$ having c as tail. Let us prove that $d\cup\{v_i\}$ is a clique of $G(V_i,E)$. Consider a vertex $u\in d-c$. The set $\{u\}\cup c$ is a clique and $\{u\}\cup c\cup\{v_i\}$ is a directed path starting in u with $k+2$ vertices, its first $k+1$ vertices being the clique $\{u\}\cup c$ and its last $k+1$ vertices being the clique $c\cup\{v_i\}$. Thus, by the definition of the k-transitivity $\{u\}\cup c\cup\{v_i\}$ is a clique, implying that u is adjacent to v_i. Hence, $d\cup\{v_i\}$ is a clique having maximum weight among all cliques in $G(V_i,E)$ which have $c\cup\{v_i\}$ as their highest indexed vertices. Thus, in Step 7, $\Omega(b)$ is the maximum weight of a maximum weight clique c' in $G(V_i,E)$ such that $b=tail(c')$ and $pr(b)$ is the highest indexed vertex in $c'-b$. Therefore the construction of T_i is correct. In Step 9 we evaluate the weight of a maximum weight clique c' in $G(V_i,E)$ such that $|c'|\geq k$ and $v_i\in c'$. In Step 10 we compare this weight with the weights of the maximal cliques c of $G(V_i,E)$ such that $|c|<k$ and $v_i\in c$, and take Ω_i as the maximum weight and b_i as the highest indexed k vertices of a maximum weight clique. Thus, the evaluation of Ω_i and b_i is correct.

Consider a maximum weight clique c of G; let v_i be its highest indexed vertex and let $b=tail(c)$. If $|c|<k$, then c is found in Step 10 and $\omega(c)=\Omega_i$. If $|c|\geq k$ then by Steps 3-9, the triple $[b,pr(b),\Omega(b)]$ appears in T_i, and Ω_i is set equal to $\omega(c)$. Thus, the algorithm finds c or a clique of equal weight.□

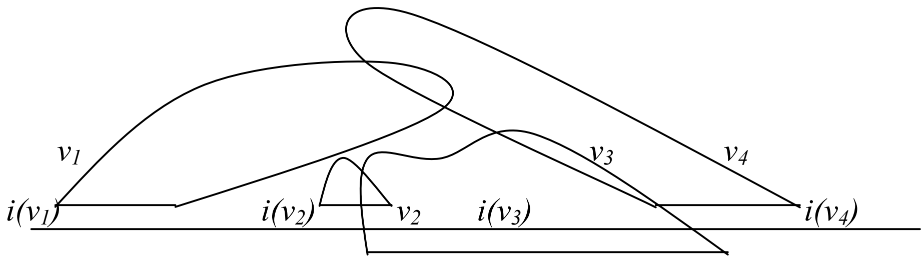
By keeping every T_i in lexicographic order and inserting appropriate pointers, we can perform the searching in Step 6 in $k|V|$ steps. Thus, the algorithm works in time $O(k|V|^{k+2})$. When $k=1$, every T_i has exactly one triple $[\{v_i\},pr(\{v_i\}),\Omega(\{v_i\})]$ and the algorithm is identical to the one in [EPL] working in time $O(|E|)$.

Consider a family I of intervals on a line L and a vertex set $V=\{v|\ i(v)\in I\}$. In the plane containing L, above L, we construct to each interval $i(v)\in I$ a filament v connecting its two endpoints; two filaments may intersect even when their intervals are disjoint (Figure 1). Let $F=\{v|\ i(v)\in I\}$ and let $G(V,E)$ be the intersection graph of F. For $u,v\in F$, we denote $i(u)<i(v)$ whenever $i(u)\cap i(v)=\phi$ and $i(u)$ appears at the left of $i(v)$. Consider two filaments

u,v such that $u \cap v \neq \phi$ and $i(u)<i(v)$. Let $K(u,v)$ be the maximum size of a subfamily D of F fulfilling: (i) every $w \in D$ intersects neither u nor v and has $i(u)<i(w)<i(v)$; (ii) every pair $w,z \in D$ has $w \cap z=\phi$ and $i(w) \cap i(z)=\phi$.

Let $k=1+max\{K(u,v)|\ u,v \in F,\ u \cap v \neq \phi \text{ and } i(u)<i(v)\}$. We say that F is a *k-interval filament family* and its intersection graph G is a *k-interval filament graph*. In other words, F is a *k-interval filament* family if for every two filaments u,v having $u \cap v \neq \phi$ and $i(u)<i(v)$, there are at most $k\text{-}1$ filaments w with $i(u)<i(w)<i(v)$ which intersect neither u nor v, are mutually disjoint and have mutually disjoint intervals. Clearly, the *2D*-interval-filament graphs are *1*-filament graphs. Let $s(v)$ be the region bounded by $v \cup i(v)$. Two filaments u,v intersect iff $s(u),s(v)$ overlap. The intersection graph GS of $\{s(v)|\ v \in V\}$ is called a *k-interval graph* since it contains the edges between any two intersecting intervals, and some additional edges between disjoint intervals (Figure 1). The complement of GS can be seen as an ***H***-*k*-mixed graph $coGS(V,\phi,E12\text{-}E4)$ where $coGS(E12\text{-}E4)$ is a *k*-transitive graph having $u \rightarrow v \in E12$ when $i(u)<i(v)$.

Figure 1



Theorem 2. The complement of a *k-interval filament* graph G is a co-*k*-interval mixed graph and a co-*k*-interval mixed graph is a *k*-transitive mixed graph $coG(V,E1,E2)$ where $coG(V,E1)$ is a *k*-interval graph (hence it is *k*-transitive), $coG(V,E2)$ is transitive, and both $coG(V,E1)$, $coG(V,E2)$ have a common topological ordering.

Proof. Let G be a *k-interval filament* graph represented by a family F of *k-interval filaments*. The edge set coE of $coG(coE)$ can be partitioned into two disjoint subsets $E1=\{u \rightarrow v|\ u,v \in V,\ u \cap v=\phi \text{ and } i(u)<i(v)\}$, $E2=\{u \rightarrow v|\ u,v \in V,\ u \cap v=\phi \text{ and } i(u) \sqsubset i(v)\}$. For two vertices u,v we have $(u,v) \in coE1$ iff $u \cap v \neq \phi$ or $i(u) \sqsubset i(v)$, iff $s(u) \cap s(v) \neq \phi$. Hence, $G(coE1)=GS$. Consider three vertices $u,v,w \in V$ such that $u \rightarrow v \in E2$ and $(v,w) \in E1$. Then $i(u) \sqsubset i(v)$, $u \cap v=\phi$ and $i(v) \cap i(w)=\phi$, $v \cap w=\phi$ implying $u \cap w=\phi$, $i(u) \cap i(w)=\phi$, i.e., $(u,w) \in E1$. Therefore, the complement of G is a co-*k*-interval mixed graph.

Let $coG(V,E1,E2)$ be a co-k-interval mixed graph where $G(coE1)$ is a k-interval graph GS and $E1=\{u\rightarrow v|\ s(u)\cap s(v)=\phi \text{ and } i(u)<i(v)\}$. Clearly $coG(E2)$ is transitive and $coG(V,E1)$, $coG(V,E2)$, $coG(V,coE)$ are acyclically oriented with a common topological ordering. Consider a directed path $p(u_1,u_{k+2})$ with $k+2$ vertices in $coG(V,E1)=coGS$ such that each of the two subpaths $p(u_1,u_{k+1})$ and $p(u_2,u_{k+2})$ induces a clique. By the definition of $E1$, $i(u_1)<i(u_2)<...<i(u_{k+2})$. Assume that $u_1\rightarrow u_{k+2}\notin E1$, i.e., $i(u_1)\cap i(u_{k+2})\neq\phi$. Since the subpaths $p(u_1,u_{k+1})$ and $p(u_2,u_{k+2})$ of p are cliques, it follows that $u_2,...,u_{k+1}$ intersect neither u_1 nor u_{k+2}, are mutually non-intersecting and $i(u_2)<...<i(u_{k+1})$, contradicting the fact that F is a *k-interval filament* family. Thus $u_1\rightarrow u_{k+2}\in E1$ and $\{u_1,u_2,...,u_{k+2}\}$ is a clique of $coG(V,E1)$. Therefore, $coG(V,E1)=GS$ is k-transitive and $coG(V,E1,E2)$ is a k-transitive mixed graph.□

By Theorem 2, the complement $coG(V,E1,E2)$ of a *k-interval filament* graph G is a k-transitive mixed graph with $coG(V,E1)$,$coG(V,E2)$ having a common topological ordering. Thus, we can find a maximum weight independent set in G in time $O(k|V|^{k+3})$, using the algorithms in [GA4] and above.

3. VARIOUS FAMILIES OF *k*-FILAMENT GRAPHS

Reference [GRU] proved that a cocomparability graph, i.e., the complement of a *1*-transitive graph, is an intersection graph of piecewise linear curves between two parallel lines. In a similar way, we can obtain an intersection representation for the complements of a k-transitive graph. Given a co-k-transitive graph G, we construct the transitive closure coH of coG and construct a representation of the co-transitive graph H as an intersection of a family of curves between two parallel lines $L1,L2$, in a plane PL. Then, for every two vertices u,v not adjacent in coG but having $u\rightarrow v\in coH$ (in the intersection representation of H the curves u,v do not intersect, u appears below v, and between u and v there are at most $k-1$ mutually non-intersecting curves which intersect neither u nor v) we attach to each one of u,v a filament in the space above PL such that the two filaments intersect. The graph G is clearly the intersection graph of the curves with the newly attached filaments.

We can use the above intersection model of co-k-transitive graphs for a farther generalization of the k-interval filament graphs. Consider a graph $coG(V,E1,E2-E3)$ such that the complement of $coG(V,E1)$ is a k-interval graph (represented by a family of interval filaments on a line L in a plane PL), $coG(V,E2-E3)$ is a k-transitive graph and $coG(V,E2)$ is transitive where $u \rightarrow v \in E2$ implies $i(u) \sqsubset i(v)$. Then, for every two vertices u,v not adjacent in $coG(V,E2)$ but having $u \rightarrow v \in E3$ (in the intersection representation of G the filaments u,v do not intersect yet, u appears inside v and between u and v there are at most k-1 mutually containing and non-intersecting filaments which intersect neither u nor v) we attach to each one of u,v a filament in the space above PL such that the two filaments intersect. The graph G is clearly the intersection graph of the filaments with the newly attached filaments and is called a *general-k-interval-filament graph*.

There is also a generalization of the subtree filament graphs similar to the one for general-*k*-interval filament graphs. Consider a rooted tree T drawn in a plane PL, such that the edges of T are oriented from sons to father. Consider a family ST of rooted subtrees of T. We say that a subtree $t(u)$ appears below a subtree $t(v)$ in T if there is a directed path in T from the root of $t(u)$ to the root of $t(v)$. To each vertex x of T we add a branch (x,y), adding it also to every subtree in ST containing x; in this way, the intersection of every two subtrees in ST contains an edge of T. As proved in [GA3], a graph $G(V,E)$ is chordal iff it can be represented as an intersection graph of such a family ST of subtrees on a tree T. For $t(u),t(v) \in ST$, we denote $t(u)<t(v)$ whenever $t(u) \cap t(v)=\phi$ and $t(u)$ appears below $t(v)$ in T; the relation "<" among the subtrees in ST is transitive and defines a transitive graph $coG(V,E2)$. Two subtrees are called incomparable if $t(u) \cap t(v)=\phi$ and none of them appears below the other. The graph $coG(V,E1)$ defined by the incomparability relation among the subtrees in ST is called an *incomparability graph*. Therefore, if G is chordal, then the edge set of its complement coG can be partitioned into two subsets $E1,E2$, such that $coG(V,E1)$ is an incomparability graph, $coG(V,E2)$ is a transitive graph and for every three distinct vertices u,v,w if $u \rightarrow v \in E2$ and $(v,w) \in E1$ then $(u,w) \in E1$. Therefore the cochordal graphs are incomparability mixed graphs. A maximum weight clique of an incomparability graph can be found in time $O(|V|)$, assuming that T and ST are given, as follows: We go in T (from its leaves toward its root) on the roots of the subtrees, and for every subtree t with root s and roots $s_1,..,s_j$ immediatly below s in T, we take $\omega(s)=max\ [\omega(t), \omega(s_1)+...+\omega(s_j)]$.

Let PP be a surface perpendicular to PL such that $PP \cap PL=T$. On PP, above T, consider a family $FT=\{f_t|\ t \in ST\}$ of filaments, called *subtree filaments*, each f_t connecting all the endpoints of t, such that if two subtrees overlap, their filaments intersect, and if two subtrees are contained one in another, their filaments may or may not intersect. When we request that each filament f_t be contained in $PP(t)$ then if two subtrees are disjoint, their filaments do not intersect and the intersection graph $G(V,E)$ of FT is *a subtree filament* *graph*. The complements *coG(V,E1,E2)* of subtree-filament graphs are exactly the cochordal mixed graphs [GA4], where *coG(E1)* is a cochordal graph and *coG(E2)* is transitive. But, as shown above, the edge set *E1* of a cochordal graph *coG(V,E1)* also has a partition *E11,E12*, such that *coG(V,E11)* is an incomparability graph and *coG(V,E12)* is transitive. Therefore, the edge set of the complement *coG* of a subtree filament graph has a partition *coG(V,E11,E12,E2)* such that *coG(V,E11)* is an incomparability graph, *coG(V,E11,E12)* is a cochordal graph and *coG(V,E12)*, *coG(V,E2)* are transitive graphs.

Let us now allow intersections between two filaments *u,v* in *PP* also when $t(u)<t(v)$, under the restriction that there are at most *k-1* filaments *w* with $t(u)<t(w)<t(v)$ which intersect neither *u* nor *v*, are mutually disjoint and have mutually disjoint subtrees; let *E4* be the subset of *E12* defined by these intersections. Let us call their intersection graph *coG(V,E11,E12-E4,E2)* a *k-subtree filament graph*. Let *s(v)* be the region bounded by $v \cup t(v)$. Two filaments *u,v* intersect iff *s(u),s(v)* overlap. The intersection graph *GT* of $\{s(v) | v \in V\}$ is called a *k-chordal graph* and it contains the edges between any two intersecting subtrees, and some additional edges between disjoint subtrees. The complement of *GT* can be seen as a ***H***-*k*-mixed graph *coGT(V,E11,E12-E4)* where *coGT(V,E11)* is an incomparability graph, *coG(V,E11,E12)* is a cochordal graph and *coGT(E12-E4)* is a *k*-transitive graph having $u \rightarrow v \in E12$ whenever $i(u)<i(v)$; the set *E2* is irrelevant for *GT*. The complement of a *k*-chordal graph *GT* is an incomparability k-mixed graph. Therefore, the complement of a *k*-subtree filament graph *coG(V,E11,E12-E4,E2)* is a (co-*k*-chordal) mixed graph.

As we did for the k-interval filament graphs, we can add to subtree filaments additional filaments in the space outside the plane *PP*, between filaments containing one another, to obtain a complement of a *k*-transitive graph. In this way, we obtain intersection graphs *G*, whose complements *coG* have a partition of the edge set *coG(V,E11,E12-E4,E2-E3)* such that *coG(V,E11)* is an incomparability graph, *coG(V,E11,E12-E4)* is a co-*k*-chordal graph, *coG(V,E12-E4)*, *coG(V,E2-E3)* are *k*-transitive graphs and *coG(V,E12), coG(V,E2)* are transitive graphs; these graphs are (incomparability-*k*-mixed)-*k*-mixed graphs. Let *F* be such a family of filaments: *F* is a *general-k-subtree filament family* and its intersection graph *G* is a *general-k-subtree filament graph*. In other words, *F* is a *general-k-subtree filament* family if for every two filaments *u,v* having $u \cap v \neq \phi$ if $t(u)<t(v)$ then there are at most *k-1* filaments *w* with $t(u)<t(w)<t(v)$ which intersect neither *u* nor *v*, are mutually disjoint and have mutually disjoint intervals, and if $t(u) \subset t(v)$ but $u \cap v \cap PP = \phi$ ($u \rightarrow v \in E4$)

then there are at most *k-1* non-intersecting filaments $w_1,...,w_{k-1}$ such that $w_i \subset w_{i+1}$ and intersect neither *u* nor *v*. Clearly, the interval-filament graphs are *1*-filament graphs. When *T* is a path, *E11=ϕ* and *coG(V,ϕ,E12-E4,E2-E3)* is the complement of a *k*-interval filament graph. When *E11=E12=ϕ*, then *coG(V,E2-E3)* is a co-*k*-transitive graph. We summarize these families of intersection graphs in Table 1:

Table 1:

Graph	**Complement**	**Characterization of complement**
Interval graph	*coG(V,ϕ,E2)*	*ϕ* mixed graph
k-Interval graph	*coG(V,ϕ,E12-E4)*	*ϕ* *k*-mixed graph (*coG(V,ϕ,E12)* is cointerval)
Interval-filament graph	*coG(V,ϕ,E12,E2)*	cointerval mixed graph (*coG(V,ϕ,E12)* is cointerval graph)
k-Interval-filament graph	*coG(V,ϕ,E12-E4,E2)*	co-*k*-interval mixed graph (*coG(V,ϕ,E12,E2)* is cointerval filament graph)
General-k-interval-filament graph	*coG(V,ϕ,E12-E4,E2-E3)*	(co-*k*-interval)-*k*-mixed graph (*coG(V,ϕ,E12,E2)* is cointerval filament graph)
Chordal graph	*coG(V,E11,E12)*	incomparability mixed graph
k-Chordal graph	*coG(V,E11,E12-E4)*	incomparability *k*-mixed graph (*coG(V,E11,E12)* is cochordal)
Subtree filament graph	*coG(V,E11,E12,E2)*	cochordal mixed graph
k-Subtree filament graph	*coG(V,E11,E12-E4,E2)*	(co-*k*-chordal) mixed graph
General-*k*-subtree filament graph	*coG(V,E11,E12-E4,E2-E3)*	(co-*k*-chordal)-*k*-mixed graph

Similarly, consider a hereditary family ***G*** of graphs. A graph *H(V,E)* is (***G***-*k-mixed)-k-mixed* if it has a partition of the edge set *H(V,E11,E12-E4,E2-E3)* such that *H(V,E11)* ∈***G***, *H(V,E12-E4), H(V,E2-E3)* are *k*-transitive graphs, *H(V,E12), H(V,E2)* are transitive graphs, for every three distinct vertices *u,v,w* if $u \rightarrow v \in E2$ and $(v,w) \in E11 \cup E12$ then $(u,w) \in E11 \cup E12$, and for every three distinct vertices *u,v,w* if $u \rightarrow v \in E12$ and $(v,w) \in E11$ then $(u,w) \in E11$. The letter ***G*** is generic and can be replaced by names of specific families. Below, we show that when the family ***G*** has a polynomial time algorithm to find a maximum clique and certain restrictions are placed on the *k*-transitive graphs *H(V,E12-E4)*, *H(V,E2-E3)*, there exists a polynomial time algorithm to find a maximum clique in the family of (***G***-*k*-mixed)-*k*-graphs *H(V,E11,E12-E4,E2-E3)*. Therefore, the family of *k*-

subtree filament graphs, with the additional restrictions, has a polynomial time algorithm to find a maximum independent set.

4. ALGORITHMS FOR MAXIMUM WEIGHT CLIQUES IN *k*-MIXED GRAPHS

Let ***G*** be a family of hereditary graphs and consider a ***G***-mixed graph *G(V,E1,E2)*, where *G(V,E2)* is a transitive graph and ***G*** has a polynomial time algorithm to find a maximum weight clique. Let *c* be a clique of *G(V,E1,E2)* having only one *E2*-sink. By eliminating in *c(E2)* the sinks one by one we obtain an edge subgraph $T_c(E2)$ such that if $u \rightarrow v \in T_c(E2)$ then for no vertex *w* of *c* there exists $u \rightarrow w \rightarrow v \in c(E2)$.

Lemma 3. The subgraph T_c is a tree called the *underlying tree of c*.

Proof. Assume that T_c contains three vertices *u,v,w* such that $u \rightarrow w \in E2$, $u \rightarrow v \in E2$. Since *c* is a clique, it follows that $v \rightarrow w \in E2$ or $w \rightarrow v \in E2$, contradicting the definition of T_c, or $(w,v) \in E1$ contradicting the definition of ***G***-mixed graphs, since $u \rightarrow w \in E2$ and $(w,v) \in E1$ imply $(u,v) \in E1$. Hence T_c is a tree.□

The clique *c* consists of the transitive closure of T_c relative to *E2* and additional *E1* edges between every two vertices of T_c which are not successors one to another.

A clique *c* of *G(V,E1,E2)* is called a *k,q,v-clique* if it has a unique *E2*-sink *v*, $|c| \leq q$ and T_c has depth at most *k*. For a clique *cc* with unique *E2*-sink *u*, containing *v*, we say that *c* is the *v,k-subclique of cc* if *c* is exactly the subset of *cc* containing *v* and all its successors in T_{cc} at depth at most *k*.

Consider a ***G***-*k*-mixed graph *G(V,E1,E2-E3)* obtained from a ***G***-mixed graph *G(V,E1,E2)* by deleting a subset *E3* of *E2* edges such that *G(V,E2-E3)* is a *k*-transitive graph. For a vertex *v* let $Y(v)=\{x_1,\ldots,x_p\}$ be the set of successors of *v* having *E3* edges towards *v* and its predecessors. Let *X(v)* be the set containing $v,x_1,\ldots,x_p$ and all the vertices *x* having $x_i \rightarrow x \rightarrow v \in E2$. A clique *c* of *G(V,E1,E2)* with *v* as unique *E2*-sink may not be a clique of *G(V,E1,E2-E3)* since some *E2* edges of *c* may be in *E3*; but by the definition of *X(v)*, for a vertex $x \in X(v) \cap c$, *X(v)* contains also every *y* for which there are *E2* paths from *x* to *y* and from *y* to *v*, it follows that $T_c(X(v) \cap c)$ is a subtree of T_c rooted at *v*. For obtaining a polynomial time algorithm to find maximum weight cliques in such graphs *G(V,E1,E2-E3)* we must add an additional restriction.

Restriction *q*: For every vertex *v* of *G*, a maximum clique of *G(Y(v),E1,E2)* has size at most *q*, *q* constant.

Let c be a clique of $G(X(v),E1,E2)$ with unique $E2$-sink v and tree T_c of depth at most k. For every source u of c there is a source $s(u)\in Y(v)$ which is its successor. Two sources u,w of c have $(u,w)\in E1$, thus $(s(u),s(w))\in E1$ because of the $E2$ paths from $s(u)$ to u and from $s(w)$ to w. Thus, the set of sources of $Y(v)$ which are successors of the sources of c forms a clique. Hence, by Restriction q, T_c has at most q leaves and being of depth at most k, it has at most $q*k$ vertices. Also, for every clique c of $G(V,E1,E2)$, the clique $c\cap X(v)$ has at most $q*k$ vertices.

Consider a clique c of $G(V,E1,E2)$ with unique $E2$-sink v and underlying tree T_c; the clique $c\cap X(v)$ has at most $q*k$ vertices. Let $v_1,...,v_r$ be the sons of v in T_c. Assume that for every v_i, $1\leq i\leq r$, every subtree of T_c with root v_i defines a clique of $G(V,E1,E2\text{-}E3)$.

Lemma 4. The set c is a clique of $G(V,E1,E2\text{-}E3)$ iff for every son v_i of v in T_c, $1\leq i\leq r$, the v_i,k-subclique c_i of $X(v)\cap c$ fulfills that $c_i\cup\{v\}$ is a clique of $G(V,E1,E2\text{-}E3)$.

Proof. If c is a clique of $G(V,E1,E2\text{-}E3)$ then for every subtree t of T_c, $t\cup\{v\}$ defines a clique of $G(V,E1,E2\text{-}E3)$.

Conversely, assume that for every i, $1\leq i\leq r$, the v_i,k-subclique c_i of $X(v)\cap c$ fulfills that $c_i\cup\{v\}$ is a clique of $G(V,E1,E2\text{-}E3)$, but c is not a clique of $G(V,E1,E2\text{-}E3)$. Thus, for two vertices u,w in c there exists $u\rightarrow w\in E3$. Assume that $w\neq v$. Since $u\rightarrow w\in E2$, there is an i, $1\leq i\leq r$, such that $w=v_i$ or $u\rightarrow w\rightarrow v_i\in E2$. Hence, u,w are contained in the clique defined by the subtree of T_c with root v_i, contradicting the fact that u,w are not adjacent in c. Therefore $w=v$. W.l.o.g. assume that u is the closest successor of v in T_c having $u\rightarrow v\in E3$ and let $p(u,v_i,v)$ be the path in T_c from u to v. Since the v_i,k-subclique c_i of $X(v)\cap c$ fulfills that $c_i\cup\{v\}$ is a clique of $G(V,E1,E2\text{-}E3)$, it follows that $p(u,v_i,v)=u_1\rightarrow u_2\rightarrow\ldots\rightarrow u_{k+1}\rightarrow\ldots \rightarrow v_i\rightarrow v$, $u=u_1$, has at least $k+2$ vertices (we can have $u_{k+1}=v_i$). The directed path $p(u_1,u_{k+1},v)$ obtained from $p(u,v_i,v)$ by going from its vertex u_{k+1} directly to v fulfills that its subpaths $p(u_1,u_{k+1})$ and $p(u_2,u_{k+1},v)$ of p are cliques and since $G(V,E1,E2\text{-}E3)$ is k-transitive it follows that u and v are adjacent. Therefore c is a clique of $G(V,E1,E2\text{-}E3)$.□

We describe an algorithm to find a maximum weight clique in the graph $G(V,E1,E2\text{-}E3)$. We assume that the vertices of V are indexed in a topological ordering $v_1,...,v_n$ relative to $E2$ (i.e., $v_j\rightarrow v_i\in E2$ implies $j<i$) and that $v_i\rightarrow v_n$ for every $1\leq i<n$, otherwise such a vertex v_n with $\omega(v_n)=0$ is added. We denote $V_i=\{v|\ v\rightarrow v_i\in E2\}\cup\{v_i\}$; by the definitions, $X(v_i)\subseteq V_i$.

For every v_i, $i=1,...,n$, we construct $X(v_i)$ and for every k,q,v_i-clique cc of $G(X(v_i),E1,E2)$ we evaluate the weight of a maximum weight clique $c_{i,cc}$ in $G(V_i,E1,E2\text{-}E3)$ such that cc is the v_i,k-subclique of $c_{i,cc}\cap X(v_i)$. There are at most $|V|^q$ such cliques cc and when considering vertices v_s, $s>i$, by the definition of $X(v_i)$, we have to consider only these subsets to avoid possible $E3$ edges from vertices in $X(v_i)$. Assume that for every $j=1,...,i-1$ we constructed $X(v_j)$ and for every k,q,v_j-clique cc of $G(X(v_j),E1,E2)$ we found a maximum weight clique $c_{j,cc}$ in $G(V_j,E1,E2-E3)$ fulfilling that cc is the v_j,k-subclique of $c_{j,cc}\cap X(v_j)$.

Now we consider v_i. We construct $X(v_i)$. We consider every k,q,v_i-clique cc of $G(X(v_i),E1,E2)$. Let $w_1,...,w_s$ be the $E2$-sinks of cc in $cc-\{v_i\}$. For every w_j and clique $c_j=cc\cap X(w_j)$, consider every k,q,w_j-clique $cc(w_j)$ in $X(w_j)$ fulfilling that $cc(w_j)\cup\{v_i\}$ is a clique and $cc(w_j)\cap cc=c_j$. Among all these k,q,w_j-cliques $cc(w_j)$ we take the one with its corresponding maximum weight clique $C(cc(w_j))$ having maximum weight; we replace $\omega(w_j)$ by $\omega(C(cc(w_j)))$. By Lemma 4, $C(cc(w_j))\cup\{v_i\}$ is a clique, since $(cc(w_j)\cap cc)\cup\{v_i\}$ is a clique. For any vertex $w_r\in V_i-X(v_i)$ we consider all k,p,w_r-cliques in $X(w_r)$, we take a maximum weight clique $C(cc(w_r))$ with w_r its $E2$-sink, among those corresponding to these k,p,w_r-cliques $cc(w_r)$. Since $C(cc(w_r))\cap X(v_i)=\phi$, $C(cc(w_r))\cup\{v_i\}$ is a clique and by the construction, there are no $E3$ edges from $C(cc(w_r))$ towards vertices v having $v_i\rightarrow v\in E2$. We replace $\omega(w_r)$ by $\omega(C(cc(w_r)))$. We find a maximum weight clique cm_i in $G(cc\cup(V_i-X(v_i))-\{v_i\},E1)$ using the polynomial time algorithm of the family ***G***. Let $c_{i,cc}=(\cup\{C(cc(w_j))|\ v_j\in cm_i\})\cup\{v_i\}$. By Lemma 4, $c_{i,cc}$ is a clique of $G(V_i,E1,E2-E3)$. We replace $\omega_{cc}(v_i)$ by $\omega(c_{i,cc})=\omega(cm_i)+\omega(v_i)$ and we insert pointers from v_i to every $v_j\in cm_i$ and corresponding $C(cc(w_j))$. We remark that a vertex $w_r\in V_i-X(v_i)$ can also be a vertex which is a successor of a vertex in $X(v_i)-cc$: if $w_r\in cc(w_j)-cc$ then $w_r\rightarrow w_j\in E2$, $(w_r,w_j)\notin E1$ and not both w_r,w_j will be chosen in cm_i. Let c_i be the maximum weight clique taken over all k,q,v_i-cliques cc of $G(X(v_i),E1,E2)$.

When $i=n$, $\omega(c_n)$ is the weight of a maximum weight clique in $G(V,E1,E2-E3)$ and such a clique can be found by backtracking on the pointers from v_n.

Lemma 5. In the above algorithm, every $c_{i,cc}$ is a maximum weight clique of $G(V_i,E1,E2-E3)$ such that cc is the v_i,k-subclique of $c_{i,cc}\cap X(v_i)$, and c_i is a maximum weight clique of $G(V_i,E1,E2-E3)$.

Proof. Consider a k,q,v_i-clique cc of $G(X(v_i),E1,E2)$. By the induction hypothesis on the algorithm, $c_{i,cc}=(\cup\{C(cc(w_j))|\ w_j\in cm_i\})\cup\{v_i\}$, for every $w_j\in V_i-\{v_i\}$, $C(cc(w_j))$ is a clique of $G(V_j,E1,E2-E3)$, $\omega(w_j)=\omega(C(cc(w_j)))$, and cm_i is a maximum weight clique of $G(V_i-\{v_i\},E1)$.

Let us prove that $c_{i,cc}$ is a clique of $G(V_i,E1,E2)$. Consider $x,z\in c_{i,cc}$, such that $x\in c_j$, $z\in c_k$, $v_j,v_k\in cm_i$ and $v_j\neq v_k$. If $x=v$, $z=v_k$, then $(x,z)\in E1$ since cm_i is a clique of $G(V_i-\{v_i\},E1)$.

If $z \neq v_k$ then $z \rightarrow v_k \in E2$. Thus, by the definition of the ***G***-mixed graphs $z \rightarrow v_k \in E2$ and $(v_j,v_k) \in E1$ imply $(z,v_j) \in E1$. If in addition $x \neq v_j$, then $x \rightarrow v_j \in E2$, and again by the definition of the ***G***-mixed graphs, $x \rightarrow v_j \in E2$ and $(v_j,z) \in E1$ imply $(x,z) \in E1$. Also, by Lemma 4, every $C(cc(w_j)) \cup \{v_i\}$ is a clique, since the w_j,k-subclique $cc(w_j) \cup \{v_i\}$ is a clique of $G(V_i,E1,E2-E3)$. Therefore $c_{i,cc}$ is a clique of $G(V_i,E1,E2)$.

Let us prove that $c_{i,cc}$ is a maximum weight clique of $G(V_i,E1,E2-E3)$ such that cc is the v_i,k-subclique of $c_{i,cc} \cap X(v_i)$. Consider a maximal clique d of $G(V_i,E1,E2-E3)$ such that cc is the v_i,k-subclique of $d \cap X(v_i)$; let the set of sinks of $d-\{v_i\}$ in $G(d \cap X(v_i),E2)$ be A_d. By the algorithm, for every sink w_j of $cc-\{v_i\}$ (and thus also of $d \cap X(v_i)-\{v_i\}$), the subclique of d rooted at w_j has weight smaller or equal to the one of $C(cc(w_j))$ and we can replace it with $C(cc(w_j))$ obtaining a clique with greater weight than d; thus we can assume that they have equal weight. Let $V_i=V_i-X(v_i)$, $d'=d-X(v_i)$, $c'=c_{i,cc}-X(v_i)$, d'' the set of $E2$-sinks of d' and c'' the set of $E2$-sinks of c', hence $c''=cm_i-X(v_i)$. Then, c' is a maximum weight clique of $G(W_i,E1,E2-E3)$ and by the induction hypothesis c'' is a maximum weight clique of $G(W_i,E1)$ and d'' is a clique of $G(W_i,E1)$, implying that $|d''| \leq |c''|$, therefore $|d| \leq |c_{i,cc}|$.

Let us prove that c_i is a maximum weight clique of $G(V_i,E1,E2-E3)$. Let d be a maximal clique of $G(V_i,E1,E2-E3)$. Then $d \cap X(v_i)$ is a clique of $G(X(v_i),E1,E2)$. Let d' be the v_i,k-subclique of $d \cap X(v_i)$. By Restriction q, d' has at most q vertices and is one of the k,q,v_i-cliques cc considered in the algorithm, implying that $|dd| \leq |c_{i,cc}| \leq |c_i|$. Therefore c_i is a maximum weight clique of $G(V_i,E1,E2-E3)$.□

By Lemma 5, c_n is a maximum weight clique of $G(V,E1,E2-E3)$. The algorithm works in time $O(|V|F(|V|,|E1|)+|V|^{q*k})$ where $F(|V|,|E1|)$ is the time complexity of the algorithm to find a maximum weight clique for a graph in ***G***.

5. HOLES OF GIVEN PARITY IN *k*-INTERVAL FILAMENT GRAPHS

We need a number of Lemmas.

Lemma 6. Consider a graph $G(V,E)$ such that $coG(V,E1,E2)$ is a ***coG*** mixed graph.

a) If an induced path $p(v_1,v_r)$ in G has some v_i such that $v_i \rightarrow v_r \in E2$, then for every v_j, $1 \leq j \leq r-2$, there exists $v_j \rightarrow v_r \in E2$.

b) Any hole $h(v_1,v_r)$ of G, which is not a hole of $G(coE1)$, has two non-adjacent vertices v_i,v_j such that $v_i \rightarrow v_j \in E2$ and has no three vertices v_i,v_j,v_s such that $v_i \rightarrow v_j \rightarrow v_s \in E2$.

c) Any hole $h(v_1,v_r)$ of G which is not a hole of $G(coE1)$ has a vertex v_i in h such that for every $v_j \in h(v_{i+2},v_{i-2})$, we have $v_j \to v_i \in E2$ and $h(v_{i+2},v_{i-2})$ is an induced path of $G(coE1)$.

Proof. a) Consider an induced path $p(v_1,v_r)$ in G having a v_i such that $v_i \to v_r \in E2$. By the definition of the ***coG*** mixed graphs, we cannot have $(v_{i+1},v_r) \in E1$, since this would imply $(v_i,v_{i+1}) \in E1$. Thus, $v_{i+1} \to v_r \in E2$; and so on, for every vertex v_j, $1 \leq j \leq r-2$, to the right and left of v_i in p.

b) Consider a hole $h(v_1,v_r)$ of G. If h is not a hole of $G(coE1)$, then h has two non-adjacent vertices v_i,v_j such that $v_i \to v_j \in E2$. Assume that h has three vertices v_i,v_j,v_s such that $v_i \to v_j \to v_s \in E2$; w.l.o.g. assume that $i<s<j$. But, (a) applied to $v_i \to v_j \in E2$ and the path $h(v_i,v_j)$ implies that for every $v \in h(v_i,v_{j-2})$, we have $v \to v_j \in E2$, hence $v_s \to v_j \in E2$.

c) Since h is not a hole of $G(coE1)$, h has two non-adjacent vertices v_i,v_j such that $v_j \to v_i \in E2$. Thus, by (a) 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 $v_j \to v_i \in E2$. By (b), there are no $E2$ edges between vertices in $h(v_{i+2},v_{i-2})$, thus $h(v_{i+2},v_{i-2})$ is an induced path of $G(coE1)$.□

Lemma 7. Consider a graph $G(V,E)$ such that $coG(V,E1,E2)$ is a ***coG*** mixed graph. Any hole h of $G(coE1)$ has no $E2$ edges and thus is also a hole of $G(V,E)$. Any induced path $p(v_1,v_r)$ of $G(coE1)$ has no $E2$ edges except may be for the first and last edges which can be $v_1 \to v_2, v_r \to v_{r-1} \in E2$; thus $p(v_2,v_{r-1})$ is an induced path of $G(V,E)$.

Proof. Assume that h has an edge $v_i \to v_j \in E2$. Since h is a hole in $G(coE1)$ it has $(v_{i-1},v_j) \in E1$, implying by the definition of ***coG*** mixed graphs that $(v_{i-1},v_i) \in E1$, contradicting the fact that v_{i-1},v_i are adjacent in $G(coE1)$. Similarly for a path $p(v_1,v_r)$ of $G(coE1)$ which cannot have $E2$ edges except for the first and last edges which can be $v_1 \to v_2, v_r \to v_{r-1} \in E2$.□

Lemma 8. Let $G(V,E)$ be a graph whose complement $coG(V,E1,E2-E3)$ is a ***coG***-*k*-mixed graph. Then a hole $h(v_1,v_r)$ of G which is not a hole of $G(coE1)$ has a vertex v_i such that for every $v_j \in h(v_{i+2},v_{i-2})$, there exists $v_j \to v_i \in E2$.

Proof. Consider a hole $h(v_1,v_r)$ of G which is not a hole of $G(coE1)$; hence, h has two non-adjacent vertices v_i,v_j such that $v_j \to v_i \in E2-E3$. Assume that v_i is an $E2$ sink of h. We cannot have $(v_{j+1},v_i) \in E1$, since, by the definition of the ***coG***-*k*-mixed graphs, this would imply $(v_{j+1},v_j) \in E1$. Thus $v_{j+1} \to v_i \in E2-E3$ (we cannot have $v_i \to v_{j+1} \in E2-E3$ since v_i is an $E2$ sink of h); similarly for v_{j-1}. Continuing in this way clockwise and counterclockwise from v_j in h, we obtain that for every vertex $v_s \in h(v_{i+2},v_{i-2})$ we have $v_s \to v_i \in E2-E3$. If

$(v_{i+1},v_i)\in E3$ or $(v_{i-1},v_i)\in E3$, then $v_{i+1}\rightarrow v_i\in E3$ or $v_{i-1}\rightarrow v_i\in E3$, respectively, since v_i is an $E2$ sink of h.□

The algorithm to find a hole $h(v_1,v_r)$ of a given parity in the complement G of a ***coG*** *mixed graph $coG(V,E1,E2)$*, when the family of $G(coE1)$ graphs has a polynomial time algorithm to find an induced path of a given parity between two vertices, works as follows: By Lemmas 6,7, h is a hole of $G(coE1)$, which can be found directly, or h has a vertex v_i such that for every $v_j\in h(v_{i+2},v_{i-2})$, $v_j\rightarrow v_i\in E2$ and $h(v_{i+2},v_{i-2})$ is an induced path of $G(coE1)$. We take every vertex of G as candidate for v_i of h, every two non-adjacent vertices in $N_G(v_i)$ as candidates for v_{i-1},v_{i+1} and every $v\in N_G(v_{i-1})$-$N_G[v_i]$, $w\in N_G(v_{i+1})$-$N_G[v_i]$, $v\rightarrow v_i\in E2$, $w\rightarrow v_i\in E2$, as candidates for v_{i-2},v_{i+2}. In $G((V\text{-}N_G[v_i,v_{i-1},v_{i+1}])\cup\{v_{i-2},v_{i+2}\}, coE1)$ we delete the vertices w which do not have $w\rightarrow v_i\in E2$. In the remaining graph GG we cannot have $v_{i-2}\rightarrow w\in E2$ or $v_{i+2}\rightarrow w\in E2$ since this would imply $v_i\rightarrow w\in E2$, by Lemma 6(a). In GG we find an induced path of the needed parity from v_{i+2} to v_{i-2}.

Gavril [GA5,GA6] described algorithms to find induced holes and paths of given parities for interval filament graphs and subtree filament graphs.

Consider a hole $h(v_1,v_r)$ of a k-interval graph GS (with complement $coGS(V,\phi,E12\text{-}E4)$) such that the intervals $i(v_1)<i(v_r)$ have the leftmost right endpoint and rightmost left endpoint, respectively, among all the intervals of h on L. Since an interval graph has no holes, in $h(v_1,v_r)$ the vertices v_1,v_r must be adjacent by an edge $v_1\rightarrow v_r\in E4$, $i(v_1)\cap i(v_r)=\phi$, and for every $w\in h(v_3,v_{r-1})$ it exists $i(v_1)<i(w)<i(v_r)$. Thus, $r<2k+3$, otherwise h would contain k mutually non-adjacent vertices, not adjacent to v_1,v_r, corresponding to k filaments v_j with $i(v_1)<i(v_j)<i(v_r)$ which intersect neither v_1 nor v_r, are mutually disjoint and have mutually disjoint intervals. Therefore, *a hole h of a given parity in a k-interval graph GS can be found* by considering every combination of $2k+2$ or less vertices and checking if it is a hole of a given parity.

The algorithm to find a hole $h(v_1,v_r)$ of a given parity in a k-interval filament graph G whose complement (Theorem 2) $coG(V,\phi,E12\text{-}E4,E2)$ is a co-k-interval mixed graph ($u\rightarrow v\in E2$ whenever $i(u)\subset i(v)$ and $u\cap v=\phi$) works as follows: By Lemmas 6,7, h is a hole of the k-interval graph $GS=G(co(E12\text{-}E4))$ having at most $2k+2$ vertices, or h has a vertex v_1 such that for every $v_j\in h(v_3,v_{r-1})$, there exists $v_j\rightarrow v_1\in E2$. The induced path $h(v_3,v_{r-1})$, $i(v_3)<i(v_{r-1})$, fulfils that all $E12$ edges are from left to right, otherwise by replacing in h every $E4$ edge $u\rightarrow w$ by a filament $u\cup w$ we obtain a hole with three mutually non intersecting filaments u,w,v_1, u inside w inside v_1, in contradiction to Lemma 6(b). To find

such a path $h(v_3,v_{r-1})$, $v_3 \rightarrow v_1 \in E2, v_{r-1} \rightarrow v_1 \in E2$, we proceed as follows: In $G((V-N_G[v_1,v_2,v_r]) \cup \{v_3,v_{r-1}\}, coE1)$ we delete the vertices w which do not have $w \rightarrow v_1 \in E2$. In the remaining graph GG we cannot have $v_3 \rightarrow w \in E2$ or $v_{r-1} \rightarrow w \in E2$ since this would imply $v_1 \rightarrow w \in E2$, by Lemma 6(a). In the remaining graph GG we find an induced path of the needed parity from v_3 to v_{r-1}. In GG we label every pair v_3,v_4, where $v_4 \in N_{GG}(v_3)$, as a path of even parity. For each parity and every induced path X with $2k$ vertices and last two vertices u_X, w_X, we find and remember one induced path from v_3 to u_X to w_X whose last $2k$ vertices form X. The next vertex (filament) v is added only to those paths whose set X fulfils that w_X intersects v, no other vertex in X intersects v and $i(u_X)<i(v)$ (all $E12$ edges are from left to right). Thus, we can find in G a hole of a given parity $h(v_1,v_r)$ with $r \geq 2k+3$ using this algorithm, and with $r<2k+3$ directly, k being constant.

The above algorithms can be adjusted to find maximum weighted holes of a given parity, when the family $G(coE1)$ has a polynomial time algorithm to find a maximum weight induced path of a given parity between two vertices. The algorithms work in time $O(|V|^{2k+4})$.

6. MINIMUM DOMINATING HOLES IN *k*-INTERVAL FILAMENT GRAPHS

Lemma 9. Consider a graph $G(V,E)$ such that $coG(V,E1,E2)$ is a ***coG*** mixed graph. A minimum dominating hole $h(v_1,v_r)$ of G either is a minimum dominating hole of $G(coE1)$ or it has a vertex v_i such that for every $v_j \in h(v_{i+2},v_{i-2})$, we have $v_j \rightarrow v_i \in E2$ and $h(v_{i+2},v_{i-2})$ is a minimum dominating induced path of $G((V-N_G[v_i,v_{i-1},v_{i+1}]) \cup \{v_{i-2},v_{i+2}\}, coE1)$ with no first and last $E2$ edges.

Proof. Consider a minimum dominating hole $h(v_1,v_r)$ of G. By Lemma 6(c), either h is a dominating hole of $G(coE1)$ or h has a vertex v_i such that for every $v_j \in h(v_{i+2},v_{i-2})$, $v_j \rightarrow v_i \in E2$ and $h(v_{i+2},v_{i-2})$ is an induced path of $G(coE1)$. In the first case, h is also a minimum dominating hole of $G(coE1)$ since by Lemma 7 any hole of $G(coE1)$ is also a hole of G. In the second case, $h(v_{i+2},v_{i-2})$ is a minimum dominating induced path of $G((V-N_G[v_i,v_{i-1},v_{i+1}]) \cup \{v_{i-2},v_{i+2}\}, coE1)$ with no first and last $E2$ edges, since by Lemma 7 only the first and last edges of an induced path of $G(coE1)$ can be $E2$ edges.□

The algorithm to find a minimum dominating hole $h(v_1,v_r)$ in the complement G of a ***coG*** *mixed graph $coG(V,E1,E2)$*, when the family of $G(coE1)$ graphs has a polynomial time algorithm to find a minimum dominating induced path between two vertices, works as follows: By Lemma 9, h is a minimum dominating hole of $G(coE1)$, which we can find directly, or h has a vertex v_i such that for every $v_j \in h(v_{i+2},v_{i-2})$, $v_j \rightarrow v_i \in E2$ and $h(v_{i+2},v_{i-2})$ is a minimum dominating induced path of $G((V\text{-}N_G[v_i,v_{i-1},v_{i+1}]) \cup \{v_{i-2},v_{i+2}\}, coE1)$ with no first and last $E2$ edges. We take every vertex of G as candidate for v_i of h, every two non-adjacent vertices in $N_G(v_i)$ as candidates for v_{i-1},v_{i+1} and every $v \in N_G(v_{i-1})\text{-}N_G[v_i]$, $w \in N_G(v_{i+1})\text{-}N_G[v_i]$, $v \rightarrow v_i \in E2$, $w \rightarrow v_i \in E2$, as candidates for v_{i-2},v_{i+2}. In $G((V\text{-}N_G[v_i,v_{i-1},v_{i+1}]) \cup \{v_{i-2},v_{i+2}\}, coE1)$ we delete the vertices w which do not have $w \rightarrow v_i \in E2$. In the remaining graph GG we cannot have $v_{i-2} \rightarrow w \in E2$ or $v_{i+2} \rightarrow w \in E2$ since this would imply $v_i \rightarrow w \in E2$, by Lemma 6(a). In GG we find a minimum dominating induced path from v_{i+2} to v_{i-2}, if such a dominating induced path exists. Among all choices of vertices v_i,v_{i-1},v_{i+1}, we take the one giving a minimum dominating hole.

A minimum dominating induced path p in an interval graph GI from a vertex v_1 to a vertex v_r, $i(v_1)<i(v_r)$, can be found as follows: We consider as candidates for v_2,v_{r-1} every two vertices $v_2 \in N_{GI}(v_1)$, $v_{r-1} \in N_{GI}(v_r)$ such that every w having $i(w)<i(v_1)$ or $i(v_r)<i(w)$ is contained in $N_{GI}(v_2,v_{r-1})$, and we find a shortest path from v_2 to v_{r-1} in the interval subgraph $GI(V\text{-}N_{GI}[v_1,v_r])$. Among all choices of v_2,v_{r-1} we take the one giving a minimum induced dominating path.

Consider a chordal graph G represented as an intersection graph of a family ST of subtrees of a tree T. *A minimum dominating induced path p in G from a vertex v_1 to a vertex v_r, can be found as follows*: W.l.o.g. we assume that: $t(v_1)<t(v_r)$, T is rooted at the root of $t(v_r)$ and every subtree of ST is on or below the path PT from the root of $t(v_1)$ to the root of $t(v_r)$. The roots of every subtree in ST which intersects PT is in PT and their intersections with PT form a family of intervals. For every vertex $v_2 \in N_G(v_1)$ such that $t(v_1) \cup t(v_2)$ intersects all subtrees below $t(v_1)$ we consider v_2 as candidate for next vertex in the path. We go on PT from left to right, on the left endpoints of the intervals, from $r(i(v_1))$ to $l(i(v_r))$. Assume that for every two intersecting subtrees $t(v_{j-1})$, $t(v_j)$ having $l(i(v_{j-1}))<l(i(v_j))$ in PT we found already a minimum induced path ending in v_{j-1},v_j which dominates all subtrees below the subpath of PT from $l(i(v_1))$ to $r(i(v_{j-1}))$. We consider now a new subtree $t(v_{j+1})$ on PT. Then, for every subtree $t(v_j)$ such that $v_j \in N_G(v_{j+1})$ we consider every subtree $t(v_{j-1})$ such that $v_{j-1} \in N_G(v_j)$ and check that $t(v_{j+1})$ dominates the subtrees below $t(v_j)$ which are not dominated by $t(v_j)$, $t(v_{j-1})$. Among all the vertices v_{j-1} we take the one with a minimum induced path and assign it to the pair v_j, v_{j+1}. When $v_j \in N_G(v_r)$, we consider every subtree $t(v_{j-1})$ such that $v_{j-1} \in N_G(v_j)$ and check that $t(v_r)$ dominates the

subtrees below $t(v_j)$ which are not dominated by $t(v_j)$, $t(v_{j-1})$. Among all the vertices v_{j-1} we take the one with a minimum induced path and assign it to the pair v_j,v_r; this is a minimum dominating induced path in G from v_1 to v_r.

Since interval filament graphs and subtree filament graphs G are ***coG***-mixed graphs $coG(V,E1,E2)$ where $coG(E1)$ are interval and chordal graphs, respectively, we can find for them minimum dominating holes by the above algorithms for ***G***-mixed graphs, interval graphs and chordal graphs.

By Section 5, a hole h of a k-interval graph GS has at most $2k+2$ vertices. Thus, *a minimum dominating hole h in a k-interval graph GS* can be found by considering every combination of $2k+2$ or less vertices and checking if it is a dominating hole.

The algorithm to find a minimum dominating hole $h(v_1,v_r)$ *in a k-interval filament graph G* whose complement $coG(V,\phi,E12\text{-}E4,E2)$ is a co-k-interval mixed graph works as follows: By Lemma 9, h is a minimum dominating hole of the k-interval graph $G(co(E12\text{-}E4))$ having at most $2k+2$ vertices, or h has a vertex v_1 such that for every $v_j \in h(v_3,v_{r-1})$, there exists $v_j \rightarrow v_1 \in E2$ and $h(v_3,v_{r-1})$ is a minimum dominating induced path in $G((V\text{-}N_G[v_1,v_2,v_r]) \cup \{v_3,v_{r-1}\}, coE1)$. We consider as candidates for v_2,v_r every two vertices $v_2,v_r \in N_G(v_1)$, such that every w having $i(w) \cap i(v_1)=\phi$ or $i(v_1) \subset i(w)$ is contained in $N_G[v_1,v_2,v_r]$. We consider as candidates for v_3,v_{r-1} every $v \in N_G(v_2)\text{-}N_G[v_1]$, $w \in N_G(v_r)\text{-}N_G[v_1]$ having $v \rightarrow v_1, w \rightarrow v_1 \in E2$. The induced path $h(v_3,v_{r-1})$, $i(v_3)<i(v_{r-1})$, is contained in the region bounded by $v_1 \cup i(v_1)$ and fulfils that all $E12$ edges are from left to right, otherwise by replacing in h every $E4$ edge $u \rightarrow w$ by a filament $u \cup w$ we obtain a hole with three mutually non intersecting filaments u,w,v_1, u inside w inside v_1, in contradiction to Lemma 6(b). In $G((V\text{-}N_G[v_1,v_2,v_r]) \cup \{v_3,v_{r-1}\}, coE1)$ we delete the vertices w which do not have $w \rightarrow v_1 \in E2$ and in the remaining graph GG we find a minimum dominating induced path from v_3 to v_{r-1}. As v_4 we consider every vertex in $N_{GG}(v_3)$ which intersects every filament w which has $i(w)<i(v_4)$ and is not intersected by v_3. For every induced path X with $2k$ vertices and last two vertices u_X, w_X, we find and remember a minimum induced path from v_3 to u_X to w_X whose last $2k$ vertices form X which dominates all vertices having intervals at the left of $r(i(u_X))$. The next vertex (filament) v is added only to those paths whose set X fulfils that w_X intersects v, no other vertex in X intersects v, $i(u_X)<i(v)$ and v intersects all filaments having intervals between $r(i(u_X))$ and $r(i(w_X))$ and are not intersected by w_X. Among all the vertices u_X we take the one with a minimum induced path and assign it to the pair w_X, v. When $w_X \in N_{GG}(v_r)$, we consider every u_X and check that v_r intersects the filaments with intervals

at the right of $r(i(u_X))$ and not intersected by w_X. Among all the vertices u_X we take the one with a minimum induced path and assign it to the pair w_X,v_r; this is a minimum dominating induced path in *GG* from v_1 to v_r. Thus, we can find in *G* a hole of a given parity $h(v_1,v_r)$ with $r \geq 2k+3$ using this algorithm, and with $r<2k+3$ directly, *k* being constant.

REFERENCES

[CA] K. Cameron, Induced matchings in intersection graphs, Discrete Math. 278(2004)1-9.

[EPL] S. Even, A. Pnueli and A. Lempel, Permutation graphs and transitive graphs, J. Assoc. Comput. Mach. 19(1972)400-410.

[FMW] S. Felsner, R. Muller and L. Wernish, Trapezoid graphs and generalizations, geometry and algorithms, Discrete Appl. Math. 74(1997)13-32.

[GA1] F. Gavril, Algorithms for a maximum clique and a maximum independent set of a circle graph, Networks 3(1973)261-273.

[GA2] F. Gavril, Algorithms on circular-arc graphs, Networks 4(1974)357-369.

[GA3] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, J. Comb. Theory Ser. B (1974)47-56.

[GA4] F. Gavril, Maximum weight independent sets and cliques in intersection graphs of filaments, Inform. Proc. Lett., 73(2000)181-188.

[GA5] F. Gavril, Algorithms for maximum weight induced paths, Inform. Proc. Lett., 81(2002)203-208.

[GA6] F. Gavril, Perfect interval-filament graphs, submitted for publication.

[GA7] F. Gavril, Genaralizations of interval-filament graphs, DIMACS TR 2004-04, DIMACS, Rutgers University, 2004.

[GRU] M. C. Golumbic, D. Rotem and J. Urrutia, Comparability graphs and intersection graphs, Discrete Math. 43(1983)37-46.

[JK] S. Janson and J. Kratochvil, Thresholds for classes of intersection graphs, Discrete Math. 108(1992)307-326.

[RO] F. S. Roberts, Discrete Mathematical Models with Applications to Social, Biological and Environmental Problems, Prentice-Hall Englewood Cliffs, NJ.