Note

Recognizing clique graphs of directed edge path graphs

Marisa Gutierrez\textsuperscript{a,\ast}, Joao Meidanis\textsuperscript{b,1}

\textsuperscript{a}Departamento de Matemática, Universidad Nacional de La Plata, C. C. 172 (1900) La Plata, Argentina
\textsuperscript{b}Instituto de Computação, Universidade Estadual de Campinas, P.O. Box 6176, 13084-971 Campinas, Brazil

Abstract

Directed edge path graphs are the intersection graphs of directed paths in a directed tree, viewed as sets of edges. They were studied by Monma and Wei (J. Comb. Theory B 41 (1986) 141–181) who also gave a polynomial time recognition algorithm. In this work, we show that the clique graphs of these graphs are exactly the two sections of the same kind of path families, and give a polynomial time recognition algorithm for them.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Clique graphs; Intersection graphs

1. Introduction

In 1986, Monma and Wei published a thorough study of several classes of intersection graphs of path families of trees [7]. A total of six classes were studied, according to whether the underlying tree was undirected, directed, or directed and rooted, and also to whether the paths were seen as vertex- or edge-sets for the purposes of forming the intersection graph. Over the last decade, many papers appeared characterizing and solving the recognition problem for clique graphs of all of these path intersection graph classes except UE and DE (see Table 1).

The purpose of this work is to characterize and provide a polynomial time recognition algorithm for the clique graphs of the DE graphs, which are intersection graphs of
directed tree paths, viewed as sets of edges. We simplify the techniques used by Prisner and Szwarcfiter [8], and show that they can be used for other classes of graphs as well. Unfortunately, the techniques do not work for UE because those graphs are not clique-Helly.

The rest of the paper is organized as follows. Section 2 contains the basic definitions and provides the basis to apply these tools to other classes of graphs. Section 3 defines the path intersection graphs we will be using. Important properties needed in Section 4 are proved here as well. Finally, Section 4 contains the main results: characterization and polynomial time recognition algorithm for clique graphs of DE graphs.

2. Definitions

In this note, all graphs are simple, i.e., without loops or multiple edges. A graph is a pair \((V,E)\) where \(V\) and \(E\) are the vertex set and edge set of \(G\), respectively. An edge with \(u\) and \(v\) as extremes is noted by \(uv\) or \(vu\). Two graphs are isomorphic when they differ only by the names of their vertices. We will not distinguish isomorphic graphs and will generally write \(G = H\) when \(G\) and \(H\) are isomorphic. A set \(C\) of vertices of a graph \((V,E)\) is complete when any two vertices of \(C\) are adjacent. A maximal complete subset of \(V\) is called a clique. A class of graphs is a subset of graphs closed under isomorphism. We denote by Graph the class of all graphs.

A family is a pair \((I,F)\), where \(I\) is a finite, nonempty set and \(F\) is a mapping from \(I\) to the class of all sets such that \(F(i)\) is a finite, nonempty set for all \(i \in I\). We denote \(F(i)\) by \(F_i\) and a family \((I,F)\) by \((F_i)_{i \in I}\), or simply by \(F\). We call elements the elements of \(\bigcup_{i \in I} F_i\) and members the sets \(F_i\).

Two families \((F_i)_{i \in I}\) and \((A_i)_{i \in J}\) are isomorphic when there are two bijections \(a : I \mapsto J\) and \(b : \bigcup_{i \in I} F_i \mapsto \bigcup_{j \in J} A_j\) such that \(b(F_i) = A_{a(i)}\) for all \(i \in I\). We will write \(F = A\) when \(F\) and \(A\) are isomorphic. Families as defined here are analogous to hypergraphs \([1,2,4]\). A class of families is a subset of families closed under isomorphism. We denote by Family the class of all families. We use boldface for graph classes and slanted for family classes.

We define the intersection operator \(L : Family \mapsto Graph\) as follows. Given a family \(F = (F_i)_{i \in I}\), define \(L(F)\) as the graph \((V,E)\), where \(V = I\) and \(E = \{ij | i \neq j \text{ and } F_i \cap F_j \neq \emptyset\}\).

<table>
<thead>
<tr>
<th>Graph class</th>
<th>Clique class</th>
<th>Recognition solved by</th>
</tr>
</thead>
<tbody>
<tr>
<td>UV</td>
<td>DuallyChordal</td>
<td>Swarcfiter and Bornstein [10]</td>
</tr>
<tr>
<td>DV</td>
<td>DuallyDV</td>
<td>Prisner and Szwarcfiter [8]</td>
</tr>
<tr>
<td>RDV</td>
<td>DuallyRDV</td>
<td>Prisner and Szwarcfiter [8]</td>
</tr>
<tr>
<td>UE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>DuallyDE</td>
<td>This paper</td>
</tr>
<tr>
<td>RDE = RDV</td>
<td>DuallyRDV</td>
<td>Prisner and Szwarcfiter [8]</td>
</tr>
</tbody>
</table>
We define the family-of-cliques operator \( C : \text{Graph} \mapsto \text{Family} \) as follows. Given a graph \( G = (V,E) \), define \( C(G) \) as the family \((F_i)_{i \in I} \), where \( I \) is the set of all cliques of \( G \) and \( F_i = i \) for all \( i \in I \).

The composite operator \( K = LC \) is the clique operator, and \( K(G) \) is the clique graph of \( G \).

We define the dual operator \( D : \text{Family} \mapsto \text{Family} \) as follows. Given a family \( F = (F_i)_{i \in I} \), define \( D(F) \) as the family \((A_j)_{j \in J} \), where \( J = \bigcup_{i \in I} F_i \) and \( A_j = \{ i \in I \mid j \in F_i \} \).

We define the two-section operator \( S : \text{Family} \mapsto \text{Graph} \) as follows. Given a family \( F = (F_i)_{i \in I} \), define \( S(F) \) as the graph \((V,E)\) where \( V = \bigcup_{i \in I} F_i \) and \( E = \{ uv \mid \text{there is } i \in I \text{ such that } u,v \in F_i \} \).

A family \((F_i)_{i \in I}\) is called intersecting when \( F_i \cap F_j \neq \emptyset \) for all pairs \( i,j \in I \). A family \((F_i)_{i \in I}\) is Helly or has the Helly property when all its intersecting subfamilies of the form \((F_i)_{i \in I'}\), for \( \emptyset \neq I' \subseteq I \), have a non-empty intersection. We write Helly for the class of all Helly families.

A graph \( G \) is clique-Helly when \( C(G) \) is a Helly family. We denote by \( \text{Helly} \) the class of all clique-Helly graphs.

A family \( F \) is conformal when its dual is a Helly family. We call Conformal the class of all conformal families. It is known that a family \((F_i)_{i \in I}\) is conformal if and only if for each triple \( i,j,k \in I \) there is an index \( l \in I \) with

\[
(F_i \cap F_j) \cup (F_j \cap F_k) \cup (F_k \cap F_l) \subseteq F_l. \tag{1}
\]

Let \( F = (F_i)_{i \in I} \) be a family. We say that \( u \in \bigcup_{i \in I} F_i \) is separated by the family \( F \) when \( \bigcap_{u \in \bigcup_{i \in I} F_i} F_i = \{ u \} \). In this case we also say that \( F \) separates \( u \). A family is separating when it separates every element in \( \bigcup_{i \in I} F_i \). A family \((F_i)_{i \in I}\) is reduced when \( i \neq j \Rightarrow F_i \not\subseteq F_j \) for all pairs \( i,j \in I \). A family is reduced if and only if its dual is separating [1,2,4]. Call Separating (Reduced) the class of all separating (reduced) families.

It is straightforward to verify that \( SC = I \), the identity (we use the same symbol \( I \) for the identity in graphs and families). We also have \( DD = I \), \( LD = S \), and \( SD = L \). In addition, \( CS = I \) for families that are both conformal and reduced [1,2,4].

We define also another operator, called \( U \) (for “unit sets”), that acts as follows. Given a family \( F = (F_i)_{i \in I} \), add members of the form \( \{ u \} \) for each \( u \in \bigcup_{i \in I} F_i \). This operator separates a family while maintaining its image under \( S \), that is, \( U(F) \in \text{Separating} \) and \( SU(F) = S(F) \) for all families \( F \).

For a graph \( G = (V,E) \), the size of \( G \) is \( |G| = |V| + |E| \). A family \( F = (F_i)_{i \in I} \) has size \( |F| = |I| + \sum_{i \in I} |F_i| \). With these definitions, the operators \( L, D, S, \) and \( U \) are all polynomially computable. The operator \( C \) can be computed with time complexity \( O(n kc) \) by a result of Tsukiyama et al. [11], where \( n, k, \) and \( c \) are \( |V|, \binom{n}{k} - |E| \), and the number of cliques of \( G = (V,E) \), respectively.

The operators were defined for graphs and families, but they can be extended to classes in the standard way. For instance,

\[
L(\text{Class}) = \{ L(F) \mid F \in \text{Class} \}
\]

and so on. This can be done because all operators are invariant under isomorphisms.
3. The classes DE and duallyDE

Let $DTP-E$ be the family class defined as follows. A family $F$ belongs to this class when there is a directed tree $T$ such that each $F_i$ is the set of edges of a directed path of $T$. In this case the tree $T$ is an underlying tree of $F$. It is known that $DTP-E \subseteq Helly$ [7, Proof of Theorem 1]. Presently, we will show that $DTP-E \subseteq Conformal$ as well (Theorem 1). The graph class $DE$ is defined as $L(DTP-E)$, and $DuallyDE$ is defined as $S(DTP-E)$.

Class $DTP-V$ is defined analogously, with $F_i$ being sets of vertices of directed paths in a directed tree. We define the graph classes $DV = L(DTP-V)$, and $DuallyDV = S(DTP-V)$.

The behavior of $K$ in some classes of intersection graphs appears in a recent paper [5]. In particular, it is shown that $K(DV) = DuallyDV$ and $K(DuallyDV) = DV$.

**Theorem 1.** $DTP-E \subseteq Conformal$.

**Proof.** We will use the characterization of conformal families mentioned in Section 2, Eq. (1). Let $F$ be a family of $DTP-E$, $T$ an underlying tree of $F$ and $F_i,F_j,F_k$ members of $F$. If either $F_i \cap F_k \subseteq F_j$, or $F_i \cap F_j \subseteq F_k$, or $F_i \cap F_j \subseteq F_k$, we are done. Suppose then that there are edges $x \in F_i \cap F_k - F_j$, $y \in F_j \cap F_k - F_i$, and $z \in F_j \cap F_k - F_j$. Because $F$ is Helly, we know that there is an edge $w \in F_i \cap F_j \cap F_k$. But then it is impossible to arrange the edges so that path $F_i$ contains $y$, $w$, $z$ and not $x$, path $F_j$ contains $x$, $w$, $y$ and not $z$, and path $F_k$ contains $x$, $w$, $z$ and not $y$. In fact, it is easy to see that edge $w$ must be between the other mentioned edges $(x,y,z)$ in each of the paths $F_i,F_j,F_k$. Removing $w$ from the underlying tree $T$, we end up with two connected components but each of $x,y,z$ would have to lie in a distinct component, which is impossible. $\square$

In the following result we prove that every family of edge sets of a directed path can be made separating or reduced without modifying its image under $S$ or $L$.

**Theorem 2.**

$$L(DTP-E) = L(DTP-E \cap \text{Separating}),$$
$$L(DTP-E) = L(DTP-E \cap \text{Reduced}),$$
$$S(DTP-E) = S(DTP-E \cap \text{Separating}),$$
$$S(DTP-E) = S(DTP-E \cap \text{Reduced}).$$

**Proof.** The first equality $L(DTP-E) = L(DTP-E \cap \text{Separating})$ is a consequence of the Clique-Tree Theorem [7, Theorem 1], which states: if a graph $G \in DE$, then there is a tree where each vertex corresponds to a clique of $G$ such that the family $DC(G)$ belongs to $DTP-E$ with this tree as an underlying tree. Since $DC(G)$ is a separating family the result follows.

For the second statement suppose that $F$ is a family that belongs to $DTP-E$, $T$ is an underlying tree of $F$ and $F_i,F_j$ are two members of $F$ such that $F_i \subseteq F_j$. Suppose that the set $F_i$ corresponds to a path ending in a vertex $u$ in $T$. Construct a tree $T'$ adding
a new vertex $v$ to $T$ and an edge $uv$. Construct also a family $F'$ which is equal to $F$ except that $F_j$ is replaced by $F'_j = F_j \cup \{uv\}$. Notice that $F'_j$ is not contained in any other $F_i$ of $F$ and that $L(F') = L(F)$. Repeating a similar operation for any member contained in another in $F$ we obtain a reduced family in $DTP-E$ with the same image under $L$ as $F$.

The last two statements are true because $DTP-E$ is closed under $U$ and under removal of contained members, respectively. □

**Theorem 3.** $K(\text{DE}) = \text{DuallyDE}$, and $K(\text{DuallyDE}) = \text{DE}$.

**Proof.**

\[
K(\text{DE}) = K(L(DTP-E)) \\
= K(L(DTP-E \cap \text{Separating})) \quad \text{by definition} \\
= LCS(DTP-E \cap \text{Separating}) \quad \text{by Theorem 2} \\
= LD(DTP-E \cap \text{Separating}) \quad \text{because $DTP-E \subseteq \text{Helly}$ and} \\
\quad \text{CS = H for conformal and} \\
\quad \text{reduced families} \\
= S(DTP-E \cap \text{Separating}) \quad \text{because $LD = S$} \\
= S(DTP-E) \quad \text{by Theorem 2} \\
= \text{DuallyDE} \quad \text{by definition.}
\]

Analogously, we can prove the other equality, as follows: $K(\text{DuallyDE}) = LCS(DTP-E)$

= $LCS(DTP-E \cap \text{Reduced}) = L(DTP-E) = \text{DE}$.

Class $\text{DE}$ is properly sandwiched between $\text{DV}$ and $\text{Helly}$, as shown in Fig. 1.

Since the $K$ operator alternates between: $\text{DV}$ and $\text{DuallyDV}$; $\text{DE}$ and $\text{DuallyDE}$; but leaves $\text{Helly}$ fixed [6], it follows that $\text{DuallyDE}$ is properly sandwiched between $\text{DuallyDV}$ and $\text{Helly}$ (Fig. 2).

On the other hand, notice that $\text{DE}$ is different from $\text{DuallyDE}$ because $K_{3,3} \in \text{DuallyDE} \setminus \text{DE}$, and the cage $K(K_{3,3})$ is in $\text{DE}$ but not in $\text{DuallyDE}$. Indeed, the
cage is the intersection graph of the nine distinct two-edge directed paths of the directed tree of Fig. 3, so it is in DE.

Since \( K^2(K_{3,3}) = K_{3,3} \), \( K_{3,3} \) is in the \( K \)-image of DE, then it is in DuallyDE. In addition, \( K_{3,3} \) cannot be a DE graph because DE graphs with \( n \geq 4 \) vertices have at most \( \lceil 3(n-4)/2 \rceil \) cliques [7, Theorem 5]. Observe that this proves that \( K(K_{3,3}) \) cannot be in DuallyDE, because \( K^2(K_{3,3}) = K_{3,3} \). \( \square \)
4. Characterization and algorithm

Inspired by the techniques of Prisner and Szwarcflter [8], we rephrase them in terms of operators and apply them to a different class: \textbf{DE}. For instance, Prisner and Szwarcflter define the graph $G'$ obtained from $G$ by adding a new vertex $v'$ and an edge $vv'$ for each $v \in V(G)$; in operator notation, $K(G')$ is $LUC(G)$. We feel that the operator notation has the advantage of highlighting the important properties of the classes that make the theorems work (properties such being separated, reduced, and so on [see Section 2]). Applications to other graph classes readily follow [3].

**Theorem 4.** $G \in \text{DuallyDE} \iff G$ is clique-Helly and $LUC(G) \in \text{DE}.$

**Proof.** $(\Rightarrow)$ $G$ is clique-Helly because all \textbf{DE} graphs are clique-Helly [7] and $K(\text{Helly}) = \text{Helly}$ [6]. If $G \in \text{DuallyDE}$, we can write $G = S(F)$, where $F \in \text{DTP-E}$ is conformal and reduced (Theorems 1 and 2). Then $LUC(G) = LUCS(F) = LU(F) \in \text{DE}$, since $DTP-E$ is closed under $U$.

(\iff) We will prove that $K(LUC(G)) = G$ and thus $G$ will be a graph in \textbf{DuallyDE} by Theorem 3:

$$K(LUC(G)) = \text{LCSDUC}(G) \text{ because } K = \text{LC, } I = \text{SD}$$
$$= \text{LDUC}(G) \text{ because } C(G) \in \text{Helly then }$$
$$\text{UC}(G) \in \text{Helly } \cap \text{ Separating and } CS = I$$
$$\text{for conformal and reduced families}$$
$$= \text{SU}(G) \text{ LD = S}$$
$$= \text{SC}(G) \text{ because } SU = S$$
$$= G \text{ since } SC = I.$$  

**Theorem 5.** If $G \in \text{DuallyDE}$ and $n = V(G)$ then there are at most $n(n+1)/2$ cliques in $G$.

**Proof.** By Theorems 1 and 2, $G$ can be written as $S(F)$, where $F \in \text{DTP-E}$ is conformal and reduced. Then each clique of $G$ is a member of $F$. Since there are at most $n(n+1)/2$ paths in the underlying tree of $F$, the result follows. □

The recognition algorithm we propose for \textbf{DuallyDE} consists in verifying if $G$ is clique-Helly, then computing $LUC(G)$ and verifying whether $LUC(G) \in \text{DE}$. Theorem 4 guarantees the correctness of this procedure. Since recognizing clique-Helly graphs and \textbf{DE} graphs can be done in polynomial time [7,9], and the number of cliques of a \textbf{duallyDE} graph is also polynomial by Theorem 5, the entire procedure takes polynomial time. Of course, one has to stop the algorithm and give a negative answer in case $G$ fails to be clique-Helly, or if more than $n(n+1)/2$ cliques are generated while computing $C(G)$. The actual complexity depends on the complexity of recognizing \textbf{DE}, which, as far as we know, has not been studied in detail so far.
Acknowledgements

We thank Jayme Szwarcfiter for making available to us a copy of the manuscript [8] prior to publication. We thank Vitae, FOMEC, FAPESP, and PRONEX/CNPq (Project PRONEX 664107/1997-4)) for financial support. The sponsors do not necessarily share the concepts or opinions expressed in this work, for which the authors are solely responsible.

References