Relatório Técnico

Núcleo de Computação Eletrônica

Chordal (2,1) - graphs -

Pavol Hell
Sulamita Klein
Fábio Protti
Loana Tito

NCE - 01/2000

Universidade Federal do Rio de Janeiro

Chordal $(2,1)$-graphs

Pavol Hell* Sulamita Klein† Fábio Protti‡ Loana Tito§

Abstract

A graph is said to be (k,l) if its vertex set can be partitioned into k independent sets and l cliques. The class of (k,l) graphs appears as a natural generalization of split graphs. In this paper, we describe a characterization for chordal $(2,1)$ graphs. This characterization leads to a $O(nm)$ recognition algorithm, where n and m are the numbers of vertices and edges of the input graph, respectively.

Keywords: (k,l) graphs, chordal graphs

1 Introduction

A graph is a split graph [6] if its vertices can be partitioned into an independent set and a clique. Split graphs are a well known class of perfect graphs. They can be recognized in polynomial time and admit polynomial time optimization algorithms [8]. Recently, generalizations of split graphs have appeared in the literature. Brandstädt [1] introduced the concept of (k,l) graphs, graphs that can be partitioned into k independent sets and l cliques. Notice that split graphs correspond to the case in which $k = l = 1$. The case $k = 3$ and $l = 0$ corresponds to the graph 3-colorability problem [7].

*School of Computing Sceince, Simon Fraser University, Burnaby, B.C., Canada, V5A1S6. E-mail: pavol@cs.sfu.ca

†IM e COPPE/Sistemas, Universidade Federal do Rio de Janeiro, RJ, 21945-970, Brasil. E-mail: sula@cos.ufrj.br

‡NCE, Universidade Federal do Rio de Janeiro, RJ, 20001-970, Brasil. E-mail: fabiop@nce.ufrj.br

§COPPE/Sistemas, Universidade Federal do Rio de Janeiro, RJ, 21945-970, Brasil. E-mail: loana@cos.ufrj.br

When $k \geq 3$ or $l \geq 3$, recognizing (k, l) graphs is a NP-complete problem [3]. In particular, Brandstädt studied the $(2,1)$ and $(2,2)$ graphs [2]. He gave a polynomial-time algorithm for the recognition of these generalized split graphs, however this algorithm is not correct [4]. A new version of it [3] has been proposed which runs in $O((n+m)^2)$ time, where n and m are the numbers of vertices and edges of the input graph, respectively. Feder *et al.* [5] studied the complexity of the more general problem of partitioning a graph in dense and sparse subgraphs (independent sets induce sparse graphs and cliques induce dense graphs). One of their results yields a polynomial-time algorithm for recognizing $(2,1)$ and $(2,2)$ graphs. Hoang and Le [9] proved that $(2,1)$ and $(2,2)$ graphs satisfy the Strong Perfect Graph Conjecture of Berge [8] and designed a polynomial-time algorithm to recognize perfect $(2,2)$ graphs.

In this paper we consider the class of chordal $(2,1)$ graphs. We describe a characterization for it that leads to a $O(nm)$ recognition algorithm.

Throughout this paper all graphs are finite, simple (i.e. without self-loops and multiple edges) and undirected. Let G be a graph. Denote its vertex set by $V(G)$ and its edge set by $E(G)$, and assume that $|V(G)| = n$ and $|E(G)| = m$. For a set X of vertices of G, denote by $G[X]$ the subgraph of G induced by X.

Denote by $N(v)$ the open neighborhood of a vertex v. If $S \subseteq V(G)$, denote by $N_S(v)$ the set of neighbors of v belonging to S, and define $\delta_S(v) = |N_S(v)|$. For $R \subseteq V(G)$, define $N_S(R) = \cup_{v \in R} N_S(v)$.

Let $S_1, S_2 \subseteq V(G)$. We say that S_1 and S_2 are *isolated* if $S_1 \cap S_2 = \emptyset$ and $N_{S_1}(S_2) = \emptyset$. In other words, S_1 and S_2 are disjoint and there is no edge linking a vertex of S_1 to a vertex of S_2.

A *clique* is a subset of vertices $C \subseteq V(G)$ inducing a complete subgraph. A *triangle* is a triple of vertices of G inducing a K_3. Write $T = xyz$ to mean that T is a triangle formed by vertices x, y, and z.

A graph is chordal if it does not contain chordless cycles with length greater than three. A graph G is chordal if and only if G has a perfect elimination scheme [8]. A perfect elimination scheme is an ordering $v_1, v_2, \ldots, v_n$ of the vertices of G such that $N(v_i) \cap V(G_i)$ is a clique, where $G_i = G[v_i, v_{i+1}, \ldots, v_n]$ for $1 \leq i \leq n$.

2 The characterization

In this section, we consider chordal graphs and present a characterization for chordal $(2, 1)$ graphs. The following lemma is a key result for our characterization.

Lemma 1 *Let G be a chordal graph. Let T be a triangle of G, and let C be a clique of G disjoint from T. Then at least one vertex of T is adjacent to all the vertices of $N_C(T)$.*

Proof. Let $X = N_C(T)$, and assume that $|X| = l$. The result is straightforward if $l \leq 2$. Assume $l > 2$ and write $X = \{v_1, v_2, \ldots, v_l\}$. Write $T = abc$. Assume by contradiction that no vertex of T is adjacent to all the vertices of X, that is, $\delta_X(y) < l$, for $y = a, b, c$. Therefore, there exists $i \in \{1, \ldots, l\}$ such that $(a, v_i) \notin E(G)$. Since $v_i \in X$, it has at least one neighbor in T. Assume w.l.o.g. that $(b, v_i) \in E(G)$. Since $\delta_X(b) < l$, there exists $j \in \{1, \ldots, l\}$, $j \neq i$, such that $(b, v_j) \notin E(G)$. Since v_j has at least one neighbor in T, (a, v_j) or (c, v_j) must belong to $E(G)$. But $(a, v_j) \notin E(G)$, otherwise v_j, v_i, b, and a would induce a C_4. Therefore, $(c, v_j) \in E(G)$. Moreover, $(c, v_i) \in E(G)$, otherwise v_j, v_i, c, and b would induce a C_4. Since $\delta_X(c) < l$ and $l > 2$, there exists $k \in \{1, \ldots, l\}$, $k \neq j$, $k \neq i$, such that $(c, v_k) \notin E(G)$. If $(a, v_k) \in E(G)$, then v_k, v_j, c, and a induce a C_4. Therefore, $(a, v_k) \notin E(G)$. Similarly, $(b, v_k) \notin E(G)$, otherwise v_k, v_j, c, and b would induce a C_4. This leads to a contradiction: $v_k \in X$ but $(y, v_k) \notin E(G)$ for $y = a, b, c$. $\square$

Let $C \neq \emptyset$ be a clique in a graph G. If $C \cap T \neq \emptyset$ for every triangle T of G, say that C is a *t-clique* of G. We are now ready now to present the main result of this work:

Theorem 2 *Let G be a chordal graph. Then the following three statements are equivalent:*
(i) G is $(2, 1)$;
(ii) there exists a t-clique C in G;
(iii) G does not contain two isolated triangles.

Proof. The equivalence (i)$\leftrightarrow$(ii) is immediate, since G is $(2, 1)$ if and only if there exists a clique $C \subseteq V(G)$ intersecting every odd cycle of G. The implication (ii) $\rightarrow$ (iii) is also simple, since (ii) implies that any two distinct triangles either intersect or are joined by an edge. The remainder of the proof consists of showing that the implication (iii) $\to$ (ii) is true.

Assume that (iii) holds. Let $t(G)$ denote the number of triangles of G. We will prove that there is a t-clique C in G, by induction on $t(G)$. If $t(G) = 0$, the result holds trivially. Assume that the result is valid for all chordal graphs G such that $t(G) < k$. Let us show that it is also valid when $t(G) = k \geq 1$. Since G is chordal, G has a perfect elimination scheme $v_1, v_2, \dots, v_n$. Let i $(1 \leq i \leq n-1)$ be the smallest index such that $t(G_{i+1}) < t(G_i)$. Observe that there indeed exits such an index i, since $t(G_n) = 0$ and $t(G_1) = k \geq 1$. Moreover, $t(G_i) = t(G_1)$.

Clearly, G_{i+1} is chordal and does not contain two isolated triangles (otherwise, G_i and G would also contain these triangles). Moreover, $t(G_{i+1}) < k$. Therefore, G_{i+1} satisfies the induction hypotheses, that is, there exists a t-clique C in G_{i+1}. Now we will show how to obtain from C a t-clique in G_i.

Let $N_i(v_i) = N(v_i) \cap V(G_{i+1})$, and let $p = |N_i(v_i)|$. Since $t(G_{i+1}) < t(G_i)$, v_i forms a triangle with two vertices of G_{i+1}. Therefore, $p \geq 2$. Let us divide the proof in cases.

Case 1: $|C \cap N_i(v_i)| \geq p - 1$.

If C contains at least $p-1$ neighbors of v_i in G_{i+1}, then C is also a t-clique in G_i, since C intersects all the triangles of G_{i+1} and also all the triangles containing v_i.

Case 2: $|C \cap N_i(v_i)| \leq p - 2$.

Assume that C contains at most $p-2$ neighbors of v_i in G_{i+1}. In this case, C contains in fact exactly $p-2$ of such neighbors, since the existence of three vertices in $N_i(v_i) \backslash C$ would imply that C does not intersect the triangle formed by them, a contradiction.

Let $x, y \in N_i(v_i) \backslash C$. Let T be the triangle $v_i x y$, and let L be clique formed by the set of vertices in C which are neighbors of v_i, x, or y, that is, $L = N_C(T)$. Note that L contains $p-2$ neighbors of v_i in G_{i+1}. Occasionally, L might be empty. Let $\mathcal{T}$ be the collection of triangles of G_{i+1} containing no vertices of L, $V(\mathcal{T})$ the subset of vertices belonging to triangles of $\mathcal{T}$, and $W = \{w \in V(\mathcal{T}) \mid w \in N(T) \text{ and } w \in Q \text{ for all } Q \in \mathcal{T}\}$.

Case 2.1: $\mathcal{T} = \emptyset$.

In this case, $W = \emptyset$ and L intersects every triangle in G_{i+1}, that is, L is a t-clique in G_{i+1}. By Lemma 1, there exists a vertex $r \in T$ which is adjacent to all the vertices in L. Thus, $L \cup \{r\}$ is a clique intersecting every triangle in G_i, that is, $L \cup \{r\}$ is a t-clique in G_i.

Case 2.2: $\mathcal{T} \neq \emptyset$ and $W \neq \emptyset$.

Let $w \in W$. Then there exists $t \in T$ such that t is adjacent to w, by the definition of W. We claim that $L \cup \{w\}$ is a clique. Clearly, $w \notin C$. Assume by contradiction that there exists a vertex $u \in L$ such that u is not adjacent to w. Let $z \in C \backslash L$ such that z is adjacent to w. There indeed exists such a vertex z, since w belongs to some triangle $T_1 \in \mathcal{T}$ and T_1 is intersected by C. In addition, z is not adjacent to any vertex in T, since $z \notin L$ by the definition of $\mathcal{T}$. Since $u \in L$, there exists $t' \in T$, not necessarily distinct from t, such that u is adjacent to t'. Observe that the subgraph induced by w, t, t', z, u contains either a C_4 or a C_5. This is a contradiction, since G is chordal. Therefore, $L \cup \{w\}$ is indeed a clique. Moreover, $L \cup \{w\}$ is a t-clique in G_{i+1}. By Lemma 1, there exists a vertex $r \in T$ which is adjacent to all the vertices in $L \cup \{w\}$. Thus, $L \cup \{r, w\}$ is a t-clique in G_i.

Case 2.3: $\mathcal{T} \neq \emptyset$ and $W = \emptyset$.

The following argument shows that this case leads to a contradiction, and thus cannot occur. Since G does not contain two isolated triangles, every $Q \in \mathcal{T}$ contains a vertex belonging to $N(T)$. Since $W = \emptyset$, there exist distinct $T_1, T_2 \in \mathcal{T}$ and distinct vertices w_1, w_2 such that $w_1 \in T_1$, $w_2 \in T_2$, and $w_1, w_2 \in N(T)$. Let $t_1, t_2 \in T$, not necessarily distinct, such that $(t_1, w_1), (t_2, w_2) \in E(G)$. Observe that $w_1, w_2 \notin C$. Let $z_1, z_2 \in C \backslash L$, not necessarily distinct, such that $(z_1, w_1), (z_2, w_2) \in E(G)$. These two vertices must exist, since T_1 and T_2 are intersected by C. Moreover, z_1 and z_2 are not adjacent to any vertex in T, since $z_1, z_2 \notin L$. We conclude this case by observing that:

a) if $(w_1, w_2) \in E(G)$, then $t_1 \neq t_2$, since otherwise the triangle $w_1 w_2 t_1$ would not be intersected by C. Moreover, w_i $(i = 1, 2)$ cannot be adjacent to t_1 and t_2 simultaneously, since otherwise $w_i t_1 t_2$ would not be intersected by C. This implies that the subgraph induced by w_1, w_2, t_1, t_2 is a C_4, a contradiction.

b) if $(w_1, w_2) \notin E(G)$, then the subgraph induced by $w_1, w_2, t_1, t_2, z_1, z_2$ contains either a C_4, a C_5, or a C_6, another contradiction. This completes the proof. $\square$

3 The algorithm

Algorithm: recognition of chordal $(2,1)$ graphs
Input: a perfec elimination scheme $v_1, \ldots, v_n$ for a chordal graph G

```
C ← {v_n}
i ← n − 1
while i ≥ 1 and C ≠ ∅ do
        let p = |N_i(v_i)|
        if |C ∩ N_i(v_i)| ≥ p − 1
           then
                  i ← i − 1
           else
                  let x, y ∈ N_i(v_i)\C
                  let T be the triangle v_i x y
                  L ← N_C(T)
                  if |C\L| ≥ 3
                     then
                            let T_1 be a triangle formed by three vertices in |C\L|
                            C ← ∅
                     else
                            𝒯 ← triangles of G_{i+1} containing no vertices of L
                            V(𝒯) ← vertices belonging to triangles of 𝒯
                            W ← {w ∈ V(𝒯) | w ∈ N(T) and w ∈ T_1 for all T_1 ∈ 𝒯}
                            if (𝒯 = ∅) or (𝒯 ≠ ∅ and W ≠ ∅)
                               then
                                      if W ≠ ∅ then
                                                  let w ∈ W
                                                  W ← {w}
                                      let r ∈ {x, y} such that L ∪ W ∪ {r} is a clique
                                      C ← L ∪ W ∪ {r}
                                      i ← i − 1
                               else
                                      let T_1 ∈ 𝒯 such that T_1 is isolated from T
                                      C ← ∅
end_while
if C = ∅
   then return T, T_1  { two isolated triangles }
   else  return C      { a t-clique in G }
end_algorithm
```

The algorithm takes as input a perfect elimination scheme $v_1, v_2, \ldots, v_n$ for a chordal graph G, and returns either a t-clique C in G, if G is $(2,1)$, or two isolated triangles in G, otherwise.

At the beginning, $C = \{v_n\}$ is set as a t-clique for G_n. Next, the scheme is scanned backwards from v_{n-1} to v_1, if $n > 1$. Each new iteration in the body of the main loop (lines 4-30) tries to update C in such a way that it becomes a t-clique for G_i. If the tentative succeeds, i is decreased and the process continues. Otherwise, two isolated triangles are found, and the algorithm stops. The correctness of the algorithm is dealt with in the next theorem.

Theorem 3 *Given a chordal graph G and a perfect elimination scheme $v_1, \ldots, v_n$ for it as input, the algorithm returns either a t-clique in G if G is $(2,1)$, or two isolated triangles in G otherwise.*

Proof. First, assume that G is $(2,1)$. We then need to show that the algorithm returns a t-clique C for G. The proof is by induction on n. If $n = 1$, then the algorithm sets $C = \{v_n\}$ in line 1, skips the **while** in lines 3-31, and finally returns C in line 34. If $n > 1$, then the algorithm finds a t-clique C for G_2, which is a chordal $(2,1)$ graph with $n-1$ vertices. Consider now the last iteration, in which $i = 1$. By Theorem 2, one of the Cases 1, 2.1, or 2.2 must occur, since $G_1 = G$ does not contain two isolated triangles. Moreover, the test in line 12 cannot be true, since otherwise the triangle T_1 defined in line 14 is isolated from T. Therefore, one of the tests in lines 5 (corresponding to Case 1) or 20 (corresponding to Cases 2.1 and 2.2) must be true. If the test in line 5 is true, then C does not need to be updated, since it is also a t-clique for $G_1 = G$. On the other hand, if the test in line 5 is false, then the test in line 20 must be true, and C is set to $L \cup \{r\}$ (if $W = \emptyset$) or to $L \cup \{r, w\}$ (if $W \neq \emptyset$). In either case, C is set as a t-clique for $G_1 = G$, and the algorithms returns it in line 34.

Assume now that G is not $(2,1)$. Thus, by Theorem 2, G contains two isolated triangles abc and def. Take a, b, c, d, e, f in such a way that they are the six rightmost vertices forming two isolated triangles in the perfect elimination scheme $v_1, \ldots, v_n$. Let v_i be the leftmost vertex in the scheme such that $v_i \in A = \{a, b, c, d, e, f\}$. Assume without loss of generality that $v_i = a$.

Observe that G_{i+1} is $(2,1)$, by the choice of A. Therefore, the algorithm finds a t-clique C for G_{i+1}. When starting the next iteration, in which the vertex $v_i = a$ is included, there exist two isolated triangles in G_i. This implies that none of the tests in lines 5 and 15 can be true, since otherwise C would be updated as a t-clique for G_i, which is a contradiction by Theorem 2. Hence, the algorithm executes either the block **then** in lines 13-15 or the block **else** in lines 28-30 (which corresponds to Case 3 of Theorem 2). In either case, a triangle $T_1 \in \mathcal{T}$ isolated from $T = v_ixy = abc$ is chosen in line 14 or 29, and the algorithm returns T and T_1 in line 33. $\square$

A straightforward analysis of the algorithm shows that it runs in $O(nm)$ time. It is sufficient to show that the complexity of a single iteration in the body of the main loop (lines 4-30) is $O(m)$. Lines 4-11 clearly require $O(n)$ time. After computing L in line 11, observe that if the set $C \backslash L$ contains three distinct elements, then the triangle T_1 defined in line 14 is isolated from T, and the algorithm must stop. Thus, if the algorithm reaches the **else** in line 16, $C \backslash L$ contains at most two elements, say z_1 and z_2. Moreover, every triangle of $\mathcal{T}$, if any, is either of the form z_1z_2w, where $z_1 \neq z_2$ and $w \in N(z_1) \cap N(z_2)$, or of the form zw_1w_2, where $z \in C \backslash L$, $w_1, w_2 \in N(z)$, and (w_1, w_2) is an edge. Therefore, computing $\mathcal{T}$ requires $O(m)$ time. Lines 18-30 require time no greater than this.

Acknowledgements: We would like to thank Prof. Yoshiko Wakabayashi who provided a valuable suggestion which improved the presentation of the algorithm.

References

[1] A. Brandstädt. Partitions of graphs into one or two independent sets and cliques. Technical Report 105, Fern Universität, Hagen, Germany, january 1991.

[2] A. Brandstädt. Partitions of graphs into one or two independent sets and cliques. *Discrete Mathematics* 152 (1996) 47–54.

[3] A. Brandstädt. The complexity of some problems related to graph 3-colorability. *Discrete Applied Mathematics* 89 (1998) 59–73.

[4] A. Brandstädt. Corrigendum. *Discrete Mathematics* 186 (1998), p. 295.

[5] T. Feder, P. Hell, S. Klein, and R. Motwani. Complexity of graph partition problems. In F. W. Thatcher and R. E. Miller, eds., *Proceedings of the 31st Annual ACM Symposium on Theory of Computing - STOC'99*, pages 464–472. Plenum Press, New York, 1999.

[6] S. Földes and P. L. Hammer. Split graphs. In F. Hoffman *et al.*, eds., *Proceedings of the 8th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 311–315. Louisiana State University, Baton Rouge, Louisiana, 1977.

[7] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science* 1 (1976) 237–267.

[8] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[9] C. T. Hoang and V. B. Le. Recognizing perfect 2-split graphs. Submitted manuscript, 1998.