Relatório Técnico

Núcleo de Computação Eletrônica

Chordal (1,l) - and (k,1) - graphs

Pavol Hell
Sulamita Klein
Fábio Protti
Loana Tito

NCE - 06/2000

Universidade Federal do Rio de Janeiro

Chordal $(1, l)$- and $(k, 1)$-graphs

Pavol Hell, [a] Sulamita Klein, [b] Fábio Protti, [c] Loana Tito [d]

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

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

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

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

Abstract

A graph is said to be a (k, l)-graph 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 work, we characterize chordal $(k, 1)$- and $(1, l)$-graphs.

Key words: (k, l)-graphs, chordal graphs

1 Introduction

Brandstädt [1] introduced the concept of (k, l)-graphs. A graph G is a (k, l)-graph if its vertices can be partitioned into k independent sets and l cliques. (k, l)-graphs appear as a natural generalization of split graphs [5], which are precisely $(1, 1)$-graphs. In [2], an $O((n+m)^2)$ recognition algorithm for $(2, 1)$-graphs and $(2, 2)$-graphs is proposed. When $k \geq 3$ or $l \geq 3$, recognizing (k, l)-graphs is an NP-complete problem. Feder *et al.* [3] designed an algorithm that applies in a more general context when a graph is to be partitioned into two parts, one in some sense 'dense' and the other in some sense 'sparse' (if 'dense' means a clique and 'sparse' means bipartite, we recover the case of $(2, 1)$-graphs). Here we consider the classes of chordal $(1, l)$- and $(k, 1)$-graphs.

Let G be a graph. If $S, S' \subseteq V(G)$, we denote by $N(S, S')$ the set of vertices of S' which are either in S or adjacent to vertices of S. In addition, we say that S and S' are *isolated* in G if $N(S, S') = \emptyset$. If vertices $v_1, v_2, ..., v_k$ form a k-clique C, we write $C = v_1 v_2 \ldots v_k$.

2 Main results

In this section we present characterizations for chordal $(k, 1)$-graphs and chordal $(1, l)$-graphs. The following observation will be useful:

Lemma 1. Let G be a chordal graph, and let C and C' be two cliques of it. Then some vertex of C is adjacent to all the vertices of $N(C, C')$.

Proof. We shall prove that, in fact, the neighborhoods of the vertices of C in C' are linearly ordered. Suppose that two vertices $v_1, v_2 \in C$ have incomparable neighborhoods in C', i.e., that neither of the sets $N(\{v_1\}, C'), N(\{v_2\}, C')$ contains the other. Then there exist vertices $u_1, u_2 \in C'$ such that u_1 is adjacent to v_1 but not to v_2, and u_2 is adjacent to v_2 but not to v_1. This is impossible, since u_1, u_2, v_2, v_1 would induce a chordless four-cycle. The lemma follows by taking the vertex in C with maximal neighborhood in C'. q.e.d.

A simple necessary condition for G to be a $(k, 1)$-graph is that G does not contain two isolated $(k + 1)$-cliques. It turns out for chordal graphs that this condition is also sufficient. Let us define a *r-intersector* in G as a clique which intersects every r-clique of G.

Theorem 2. The following statements are equivalent for a chordal graph G: (i) G is a $(k, 1)$-graph; (ii) G contains a $(k + 1)$-intersector; (iii) G does not contain two isolated $(k + 1)$-cliques.

Proof. The implications (i) $\rightarrow$ (ii) and (ii) $\rightarrow$ (iii) are immediate. The implication (ii) $\rightarrow$ (i) is also simple: if C is a $(k + 1)$-intersector in G, then the size of a maximum clique in $G \setminus C$ is at most k, and since G is perfect, $G \setminus C$ is k-partite. The remainder of the proof consists of showing that (iii) implies (ii). Let G be a chordal graph and C a clique in G which intersects the greatest number of $(k+1)$-cliques in G. Suppose that some $(k+1)$-clique C_1 is disjoint from C. Lemma 1 guarantees that there is a vertex $u \in C_1$ adjacent to all vertices of $L = N(C_1, C)$. The clique $L \cup \{u\}$ cannot intersect more $(k+1)$-cliques than C, so there exist a $(k+1)$-clique $v_1 v_2 \dots v_{k+1}$ where $v_{k+1} \in C \setminus L$. By (iii), one of the vertices $v_1, \dots, v_k$, say v_1, is either in C_1 or adjacent to a vertex of C_1. Let A be the set of all vertices v_1 such that some $(k+1)$-clique $v_1 \dots v_{k+1}$, where $v_{k+1} \in C \setminus L$, is disjoint from L, and v_1 is either in C_1 or adjacent to a vertex of C_1. We claim that each $v_1 \in A$ is adjacent to all vertices of L: Indeed, if $y \in L$ is not adjacent to v_1, then there is a $t \in C_1$ (possibly $t = u$) such that y, v_{k+1}, v_1, t, u, y is a chordless cycle. (In case $v_1 \in C_1$, note that $v_1 \neq u$ as $v_{k+1} \notin L$, but we do take $t = u$.) A similar argument shows that v_1 is adjacent to u: take any four-cycle $v_1, t \in C_1, u, y \in L$. We also claim that any two $y, y' \in A$ are adjacent in G: Otherwise, suppose that $v'_1 \dots v'_{k+1}$, where $v'_{k+1} \in C$, is the $(k + 1)$-clique through v'_1. Then there would be a chordless cycle obtained by taking v_1, the vertices v_{k+1}, v'_{k+1} (possibly just one vertex if

$v_{k+1} = v'_{k+1}$), followed by v'_1, then one or two vertices of C_1 and back to v_1. Now the set $L \cup A \cup \{u\}$ intersects C_1 and also all the $(k+1)$-cliques that were intersected by C, contradicting the maximality of C. q.e.d.

In the remainder of this section, we shall characterize chordal $(1,l)$-graphs. We first propose a greedy algorithm for recognizing chordal $(1,l)$-graphs.

A greedy algorithm for recognizing chordal $(1,l)$-graphs

Let G be a chordal graph. The greedy algorithm labels the vertices of G with $0, 1, 2, \ldots$ in such a way that the vertices labeled 0 form an independent set, and the vertices labeled j, $j > 0$, form a clique C_j. Let $v_1, \ldots, v_n$ be a perfect elimination ordering for G. The algorithm proceeds as follows: label vertex v_1 with 0, and having labeled vertices $v_1, \ldots, v_{i-1}$, try to label v_i with the largest label already used; if not possible, use a new label for it. To illustrate this idea, let us show that for split graphs this algorithm works. This amounts to attempting to label the vertices of G with 0, 1. If the greedy algorithm cannot scan the ordering to the end using 0 and 1 only, we will show that the graph is not a split graph by exhibiting an induced $2K_2$ contained in G. (A chordal graph is split if and only if it does not contain $2K_2$ as an induced subgraph [5].) Suppose that the algorithms arrives at a vertex v_i which cannot be labeled either 0 or 1. Let u_1 the first vertex labeled 1. Thus, there must exist a vertex labeled 0 to which u_1 is adjacent, for otherwise it would have been labeled 0. Denote this vertex by w_1. Since v_i cannot be labeled 0, there must exist a vertex $v_j \neq w_1$ labeled 0, $j < i$, such that $(v_j, v_i) \in E(G)$. Write $v_i = u_2$ and $v_j = w_2$. Let us show that the vertices u_2, w_2, u_1, w_1 induce a $2K_2$. Indeed, $(w_2, w_1) \notin E(G)$, as both vertices have label 0; $(u_2, u_1) \notin E(G)$, for otherwise, as u_1 is adjacent to all vertices labeled 1 between u_1 and u_2, by the elimination ordering u_2 would have been labeled 1; $(u_2, w_1) \notin E(G)$, for otherwise u_2 and u_1 would be adjacent by the elimination ordering; finally, $(w_2, u_1) \notin E(G)$, for otherwise either w_2 would have been labeled 1 (in case w_2 is located to the right of u_1 in the ordering) or u_2 and u_1 would be adjacent (in case w_2 is located to the left of u_1). Now we are able to prove the following property of the greedy algorithm:

Claim. Each time the greedy algorithm uses a new label r for a vertex v_i, v_i belongs to an induced subgraph H_r of G isomorphic to rK_2 (a graph formed by r mutually disjoint edges). Moreover, the edges of H_r are of the form $(u_j, w_j), 1 \leq j \leq r$, where u_j is the first vertex labeled j and w_j is labeled 0. (In this case, it is clear that $v_i = u_r$.)

Proof. By induction on r. For $r = 1$ the result is immediate. The case $r = 2$ has already been considered. Suppose now that the greedy algorithm uses a new label $r > 2$ for v_i. Let v_j be the first vertex labeled $r - 1$. By the induction hypothesis, v_j belongs to an induced subgraph H_{r-1} of G isomorphic to $(r-1)K_2$ such that its edges are $(u_j, w_j), 1 \leq j \leq r-1$, as described above. By the elimination ordering, it is not difficult to see that there are no edges linking v_i to $u_j, w_j (1 \leq j \leq r-1)$. Write $v_i = u_r$. As u_r cannot be labeled 0, let w_r be a vertex labeled 0 to which u_r is adjacent. It is clear that w_r is not adjacent to $w_j, 1 \leq j \leq r-1$, since all of them are labeled 0. It remains to show that w_r is not adjacent to any $u_j, 1 \leq j \leq r-1$. Suppose that $(w_r, u_j) \in E(G)$ for some $j \leq r-1$. If w_r appears to the right of u_j, then w_r would be labeled j, since u_j is adjacent to all vertices labeled j between u_j and w_r; otherwise, if w_r appears to the left of u_j, then u_j and u_r would be adjacent since both are neighbors of w_r. q.e.d.

Corolary. Let G be a chordal graph. Then G is $(1, l)$ if and only if it does not contain $(l+1)K_2$ as an induced subgraph.

Proof. The necessity is immediate. The sufficiency is based on the fact that if G is not $(1, l)$ then the greedy algorithm must necessarily arrive at a vertex v_i which cannot receive any label from $0, 1, \ldots, l$. This implies that G contains an induced $(l+1)K_2$ by the previous claim. q.e.d.

Observe that the greedy algorithm may be used to find the minimum l for which a chordal graph G is $(1, l)$ by scanning the entire ordering. We conclude by remarking that it is possible to prove the following result: a chordal graph G is (k, l) if and only if G does not contain $(l+1)K_{k+1}$ as an induced subgraph.

References

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

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

[3] 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.

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

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