ENUMERATING THE MAXIMAL CLIQUES OF A CIRCLE GRAPH

Jayme L. Szwarcfiter
Magali M. Araújo Barroso (*)

NCE-14/88

Dezembro/88

Universidade Federal do Rio de Janeiro
Núcleo de Computação Eletrônica
Caixa Postal 2324
20001 - Rio de Janeiro - RJ
BRASIL

(*) Universidade Federal de Minas Gerais, Departamento de Ciência da Computação.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

RESUMO

Descrevemos a noção de orientação localmente transitiva de um um grafo não direcionado, como uma generalização de orientação transitiva ordinária . Como aplicação, obtemos um algoritmo para a geração de todas as cliques maximais de um grafo circular G, cuja complexidade é $O(n(m+\alpha))$, onde n,m e α são o número de vértices, arestas e cliques maximais de G. Em adição, mostramos que o número exato de tais cliques pode ser computado em tempo $O(nm)$.

ABSTRACT

We describe the notion of locally transitive orientations of an undirected graph, as a generalization of ordinary transitive orientations. As an application, we obtain an algorithm for generating all maximal cliques of a circle graph G in time $O(n(m+\alpha))$, where n,m and α are the number of vertices, edges and maximal cliques of G. In addition, we show that the actual number of such cliques can be computed in $O(nm)$ time.

1. INTRODUCTION

We describe an algorithm for generating all maximal cliques of a circle graph having time complexity $O(n(m+\alpha))$, where n,m and α are respectively the number of vertices, edges and maximal cliques of the graph. The algorithm is an application of a special orientation of a graph called locally transitive, a generalization of transitive orientations.

Clearly, the maximal cliques of a general graph can be enumerated by the algorithm of Tsukiyama et al. [7], but it would require $O(nm'\alpha)$ time, where m' is the number of edges of the complement of the graph. As for cliques of circle graphs, the following are some known results.

Rotem and Urrutia [6] generate all the maximum (largest maximal) cliques in $O(n(n+\alpha^*))$ time, α^* being the number of such cliques. Algorithms for computing one maximum clique include that of Rotem and Urrutia [6], Gavril [3] and Buckingham [1], respectivily of time complexities $O(n^2)$, $O(n^3)$ and $O(n+m \log w)$, where w is the size of the maximum clique. A maximum weighted clique can be computed by the algorithms of Hsu [4] and Buckingham [1] in time $O(n^2+m \log\log n)$ and $O(n+m\delta)$, respectively where δ is the maximum degree of the graph.

In addition, we show that the exact number α of maximal cliques of a circle graph can be computed in $O(nm)$ time. Observe that α can be large compared to n. In fact, the examples by Moon and Moser [5] of graphs having a maximum number of maximal cliques are circle graphs. Therefore it might be highly inneficient to compute α by generating all maximal cliques and counting them. This is a motivation for describing a polynomial time algorithm for computing such number.

The following are the terminology and notation employed.

G denotes an undirected graph with vertex set $V(G)$ and edge set $E(G)$. If $Z \subset V(G)$ then $G<Z>$ denotes the subgraph induced in G by Z. A clique is a complete subgraph of G and a maximal clique is one not properly contained in any other. A circle graph G is the intersection graph of a set of n chords of a circle C. We may assume that no two chords of C share a common endpoint. A circle sequence S of G is the sequence of the 2n distinct endpoints of the chords of C that we obtain as we walk around C in some fixed direction, starting from a chosen point of C. Denote by $S_1(v)$ and $S_2(v)$ respectively the first and second instances in S of the chord of C corresponding to $v \in V(G)$. Write $S_i(v) < S_j(w)$ whenever $S_i(v)$ proceeds $S_j(w)$ in S. Hence $S_1(v) < S_2(v)$, for all $v \in V(G)$.

$\vec{G}$ denotes an acyclic orientation of G. $A_v(\vec{G})$ and $A_v^{-1}(\vec{G})$ are the subsets of vertices incident to edges leaving and entering v, respectively. For $v,w \in V(G)$, v is an ancestor of w in $\vec{G}$ whenever the digraph contains a v-w path. In this case, w is a descendant of v. Denote by $D_v(\vec{G})$ the set of all descendants of v. If $w \in D_v(\vec{G})$ and $v \neq w$, then v is a proper ancestor of w and the latter a proper descendant of v. $\vec{G}$ is called *transitive whenever* $(v,w),(w,z) \in E(\vec{G})$ *implies* $(v,z) \in E(\vec{G})$. The transitive reduction of $\vec{G}$ is the spanning subdigraph of $\vec{G}$ formed exactly by the edges which are not implied by transitivity.

In Section 2 we present the locally transitive orientations of a graph. The method described for all maximal clique enumeration employs such orientations. In Section 3 we show that a circle graph always admits a locally transitive orientation and the latter is easily obtained from the circle sequence of the graph. The actual algorithms are described in Section 4, whereas the results are summarized in the last section.

2. LOCALLY TRANSITIVE ORIENTATIONS

Let G be an undirected graph, $|V(G)| > 1$, and $\vec{G}$ an acyclic orientation of it. Let $v,w \in V(G)$, v a proper ancestor of w in $\vec{G}$, and denote by $Z(v,w) \subset V(G)$ the subset of vertices which are simultaneously descendants of v and ancestors of w in $\vec{G}$. An edge $(v,w) \in E(\vec{G})$ *induces local transitivity* when $\vec{G}\langle Z(v,w)\rangle$ is a transitive digraph. Clearly, in this case the vertices of any path from v to w in $\vec{G}\langle Z(v,w)\rangle$ induce a clique in G. In addition, (v,w) *induces maximal local transitivity* (or shortly, is a *maximal edge*) when there is no edge $(v',w') \in E(\vec{G})$ different from (v,w) such that simultaneously v' is an ancestor of v and w' a descendant of w in $\vec{G}$. The orientation $\vec{G}$ is *locally transitive* when each of its edges induces local transitivity. As an example, the digraph of figure 1(b) is a locally transitive orientation of the graph shown in 1(a).

The application of locally transitive orientations to the enumeration of maximal cliques is based on the following theorem.

Theorem 1: Let G be an undirected graph, $\vec{G}$ a locally transitive orientation of it and $\vec{G}_R$ the transitive reduction of $\vec{G}$. Then there exists a one-to-one correspondence between maximal cliques of G and v-w paths in $\vec{G}_R$, for all maximal edges $(v,w) \in E(\vec{G})$.

Proof: Let C be a maximal clique of G. Then the subdigraph induced in $\vec{G}$ by C is a tournment and therefore has a spanning path $\vec{P}$. Because $\vec{G}$ is locally transitive and C a maximal clique it follows that no edge of $\vec{P}$ can be implied by transitivity. Hence $\vec{P}$ is also a spanning path of C in $\vec{G}_R$. Suppose. $\vec{P}$ is the path in $\vec{G}$ from vertex v to w. Then $(v,w) \in E(\vec{G})$ must be a maximal edge, otherwise C is not a maximal clique. Therefore we can choose $\vec{P}$ as the v-w path corresponding to C satisfying the conditions of the theorem. Conversely, any maximal edge $(v,w) \in E(\vec{G})$ defines a clique C formed by the vertices of a v-w path in $\vec{G}_R$. C must be a maximal clique, otherwise (v,w) is not a maximal edge □.

(a) (b)

Figure 1: A Locally Transitive Orientation

3. CIRCLE GRAPHS

Theorem 1 suggests a method for enumerating the maximal cliques of a graph G, provided a locally transitive orientation $\vec{G}$ is given. The problem remains how to construct $\vec{G}$. Below we show that if G is a circle graph then it is always simple to compute such an orientation.

The next lemma is immediate.

Lemma 1: Let G be a circle graph, S a circle sequence of it and $v_1,\ldots,v_k \in V(G)$ satisfying $S_1(v_i) < S_1(v_{i+1})$, $1 \leq i < k$. Then $\{v_1,\ldots,v_k\}$ induces a clique in G if and only if $S_1(v_k) < S_2(v_2) < \ldots < S_2(v_k)$.

Let G be a circle graph and S a circle sequence of it. A S_1-orientation $\vec{G}$ of G is one in which every directed edge $(v,w) \in E(\vec{G})$

satisfies $S_1(v) < S_1(w)$.

Lemma 2: If (v_1, v_k) is an edge of a S_1-orientation $\vec{G}$ of G then each path $v_1, \ldots, v_k$ induces a clique in G.

Proof: If follows $S_1(v_1) < \ldots < S_1(v_k)$. Also, since $(v_1, v_k) \in E(\vec{G})$ we conclude that $S_1(v_k) < S_2(v_1)$. Now, for i=1,...,k-1, $(v_i, v_{i+1}) \in E(\vec{G})$ and $S_1(v_i) < S_1(v_{i+1})$ imply $S_2(v_i) < S_2(v_{i+1})$. Therefore, $S_1(v_k) < S_2(v_1) < \ldots < S_2(v_k)$. By lemma 1 it follows that $v_1, \ldots, v_k$ induces a clique in G □.

Theorem 2: Any S_1-orientation is locally transitive.

Proof: Clearly, a S_1-orientation $\vec{G}$ of the circle graph G is always acyclic. Suppose it is not locally transitive. Then there exists an edge $(v,w) \in E(\vec{G})$ which does not induce local transitivity. That is, there is a path from v to w in $\vec{G}$ which does not induce a clique in G, contradicting lemma 2 □.

4. THE ALGORITHMS

The algorithm for finding all maximal cliques of a circle graph G can now be described as follows:

1. Construct a S_1-orientation $\vec{G}$ of G
2. Construct the transitive reduction $\vec{G}_R$
3. Find all maximal edges of $\vec{G}$
4. For each maximal edge $(v,w) \in E(\vec{G})$
   find all v-w paths in $\vec{G}_R$ (each of them defines a maximal clique of G).

We discuss below the steps involved in this algorithm.

A S_1-orientation $\vec{G}$ of the given circle graph can be easily obtained from its circle sequence. The sequence itself can be computed in O(nm) time, following the characterization by Gabor, Hsu and Supowit [2]. The amount of time required by the construction of $\vec{G}_R$ is less than O(nm). Step 3 of the above algorithm can be implemented following an observation that whenever $\vec{G}$ is a locally transitive orientation then $(v,w) \varepsilon E(\vec{G})$ is a maximal edge iff

$$A_v(\vec{G}) \cap A_w(\vec{G}) = A_v^{-1}(\vec{G}) \cap A_w^{-1}(\vec{G}) = \Phi$$

There is no difficulty to check this condition and therefore obtain all maximal edges in O(nm) time. For implementing step 4 of the algorithm, we define for each $v \varepsilon V(\vec{G})$, $Z(v) \subseteq V(\vec{G})$ as the subset of vertices simultaneously descendants of v and ancestors of w, for every w such that (v,w) is a maximal edge. Clearly, the required v-w paths in $\vec{G}_R$ taken from some chosen vertex v are exactly the source-sink paths in $\vec{G}_R<Z(v)>$. Each of these paths can be obtained in O(n) time, using a simple unconstrained dept-first search of $\vec{G}_R<Z(v)>$, starting at the source v. The construction of $\vec{G}_R<Z(v)>$ takes O(m) time for each vertex v. Therefore the overall complexity of the maximal clique finding algorithm is $O(n(m+\alpha))$.

Now, we proceed to describe the algorithm for computing the number of maximal cliques α of a circle graph G, as follows.

1. Perform steps 1,2 and 3 of the previous (maximal clique finding) algorithm.
2. For each $v \varepsilon V(\vec{G})$, let W(v) be the subset of vertices w such that (v,w) is a maximal edge, and construct $\vec{G}'_R<Z(v)>$ as the digraph obtained from $\vec{G}_R<Z(v)>$ by adding a new vertex w' and and edge (w,w') for each $w \varepsilon W(v)$.
3. For each $v \varepsilon V(\vec{G})$, compute the number $\alpha(v)$ of source-sink paths in $\vec{G}'_R<Z(v)>$, as follows:

$$\alpha(u) = \begin{cases} 1, & \text{if } u \text{ is a sink of } \vec{G}'_R\langle Z(v)\rangle, \text{ otherwise} \\ \Sigma\, \alpha(t), & \text{where } t \in A_u(\vec{G}_R\langle Z(v)\rangle) \end{cases}$$

4. $\alpha = \Sigma\alpha(v)$.

As noted before, step 1 can be implemented in O(nm) time and the same applies for step 2. For each considered digraph $\vec{G}_R\langle Z(v)\rangle$, we can obtain $\alpha(v)$ in O(m) time if we compute $\alpha(u)$ backwards from the sinks to the source v. Therefore the everall time bound is O(nm).

The correctness of this algorithms follows directly from the one-to-one correspondence between maximal cliques of G and source-sink paths in $\vec{G}_R\langle Z(v)\rangle$, for each $v \in V(\vec{G})$.

5. CONCLUSIONS

We have described an algorithm for finding all maximal cliques of a circle graph G in time $O(n(m+\alpha))$, where n, m and α are the number of vertices, edges and maximal cliques of G, respectively. The maximal cliques are generated in lexicographical ordering. In addition, we have also formulated a method for computing the total number of maximal cliques of G in overall time O(nm). Both algorithms can be applied to any graph (not necessarily a circle one) to which a locally transitive orientation is known.

REFERENCES

1. M.A. Buckingham, "Circle graphs", Comp.Sc.Rep. NSO-21, Courant Institute of Mathematical Sciences, New York, NY, 1980;

2. C.P. Gabor, W.L. Hsu and K.J. Supowit, Recognizing circle graphs in polynomial time, Proc. 26th Ann.Symp. on Foundations of Computer Science, IEEE Computer Society, Los Angeles, CA, 1985;

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

4. W.L. Hsu, Maximum weight clique algorithms for circular-arc graphs and circle graphs, SIAM J. Comput. 14 (1985), 224-231;

5. J.W. Moon and L.Moser, On cliques in graphs, Israel J. Math. 3 (1965), 23-28;

6. D. Rotem and J. Urrutia, Finding maximum cliques in circle graphs, Networks 11 (1981), 269-278;

7. S. Tsukiyama, M. Ide, H. Ariyoshi and I. Shirakama, A new algorithm for generating all the maximal independent sets, SIAM J. Comput. 6 (1977), 505-517.