A Fully Dynamic Algorithm for Recognizing and Representing Chordal Graphs

Yrysgul Tursunbay kyzy

A.P. Ershov Institute of Informatics Systems
Russian Academy of Sciences, Siberian Branch
6, Acad. Lavrentjev pr., Novosibirsk, 630090, Russia
ryskulya@gmail.com

Abstract. This paper considers the problem of recognition and representation of dynamically changing chordal graphs. The input to the problem consists of a series of modifications to be performed on a graph, where modifications can be additions or deletions of complete r-vertex graphs. The purpose is to maintain a representation of the graph as long as it remains a chordal graph and to detect when it ceases to be so.

1 Introduction

A graph G is said to be *chordal* if every cycle of length 4 or more contains a chord (an edge between two non-consecutive vertices in a cycle). From the practical point of view, chordal graphs have numerous applications in, for example, sparse matrix computation (e.g., see [1]), relational databases [2], and computational biology [3].

Several authors have studied the problem of dynamically recognizing and representing various graph families. [4] devises a fully dynamic recognition algorithm for chordal graphs which handles edge operations in $O(n)$ time. The authors of [5] improve the current complexities for maintaining a chordal graph by starting with an empty graph and repeatedly adding or deleting edges. They use their result to ameliorate the time bound for the biology-based problem of improving the matrix representation of an evolutionary tree (phylogeny) which contains errors. For proper interval graphs [6], each update can be supported in $O(d+\log n)$ time where d is the number of edges involved in the operation.

Unlike the existing works, we develop an algorithm for maintaining representations of chordal graphs under complete r-vertex graph insertions or deletions, where cliques have at least 3 vertices. Since a clique tree of a chordal graph has at most n nodes, each operation performs in $O(n)$ time.

2 Preliminaries

In this article, we assume that the reader has a moderate familiarity with graph theory. This section aims at defining notions and notations related to chordal graphs.

I. Virbitskaite and A. Voronkov (Eds.): PSI 2006, LNCS 4378, pp. 481–486, 2007.

Let $G = (V(G), E(G)) = (V, E)$ be a finite undirected and simple graph with $|V| = n$ vertices and $|E| = m$ edges. The subgraph of G *induced by* S is $G[S] = (S, E[S])$, where $E[S] = \{uv \in E \mid u, v \in S\}$. Let K_r be a complete r-vertex graph, where $r \geq 3$[1]. We define the following:

$$V(G) \cap V(K_r) = p, E(G) \cap E(K_r) = q,$$

$$G + K_r = \{V(G) + (V(K_r) \setminus p), E(G) + (E(K_r) \setminus q)\},$$
$$G - K_r = \{V(G) - (V(K_r) \setminus p), E(G) - E(K_r)\}.$$

A subset S of V is called a *separator* if $G[V \setminus S]$ is disconnected. S is a *uv-separator* if vertices u and v from $G[V \setminus S]$ are in different connected components of $G[V \setminus S]$; S is a *minimal uv-separator* if none of S subsets is a uv-separator. S is a *minimal separator* if S is a minimal uv-separator for all u and v from $G[V \setminus S]$.

A *clique* of a chordal graph G is a non-empty subset $C \subseteq V$ such that all the vertices of C are mutually adjacent. A clique K is *maximal* if K is not properly contained in another clique. A *clique tree* of G is a tree T such that its nodes have a 1-1 correspondence with maximal cliques of G, edges correspond to non-empty intersections of pairs of maximal cliques, and for all vertices v in G, the set of maximal cliques which contain v induces a subtree of T. It is worth remarking that a graph G is *chordal* iff it has a clique tree (see [3],[8],[9] for detailed explanation). We use u, v as vertex names of G and x, y as node names of T. The nodes x and y correspond to maximal cliques K_x and K_y of G. A clique tree has n nodes and each edge xy of T has the weight $w(xy) = |K_x \cap K_y|$. There are known algorithms to find a clique tree of a chordal graph in $O(m + n)$ time (see, e.g.,[1]).

We use $I_j = K_j \cap N(K_j)$ to denote a minimal separator of a graph G, where $N(K_j)$ is a set of all nodes of a tree T adjacent with node j.

3 Algorithm

We consider how to implement modification operations. Some of these operations are identical to those in [4], but are repeated here so that the reader can have easy access to the full algorithm.

Our algorithm supports the following operations: **Insert Query** and **Delete Query** which return "yes" if a modified graph ($G+K_r$ and $G-K_r$, respectively) is chordal and "no" otherwise; **Insert** and **Delete** modify the clique tree T according to made modification.

We first deal with the insertion of a complete r-vertex graph K_r.

Lemma 1. ([10], [11]) *Let G be a connected chordal graph with its clique tree T. Then*

[1] For $r = 1$ an incremental dynamic algorithm which considers addition of vertices is presented in [7], for $r = 2$ a dynamic algorithm which deals with addition and deletion of edges is proposed in [4].

(i) *a set S is a minimal vertex separator of G iff $S = K_x \cap K_y$ for some edge $xy \in T$,*

(ii) *if $S = K_x \cap K_y$ for $xy \in T$, then S is a minimal uv-separator for any $u \in K_x \setminus S$ and $v \in K_y \setminus S$.*

Theorem 1. *Let G be a chordal graph without a complete r-vertex graph K_r. Then $G + K_r$ is chordal iff the following conditions are satisfied:*

(i) *G has a clique tree T with $u \in K_x, v \in K_y$ such that $u, v \in K_r$ for some edge xy in T,*

(ii) *there is a path from x to y in T such that $K_r \cap I_j \neq \varnothing$, where I_j is a set of vertices contained in this path.*

Proof. (i) Let $I = K_x \cap K_y \neq \varnothing$. Since uv is not an edge of G, we have $u \notin K_y, v \notin K_x$ and hence $u \in K_x - I, v \in K_y - I$. By Lemma 1, I is a uv-separator.

Let C be any cycle in $G + K_r$ with length ≥ 4 that contains uv where $uv \in q$. Let $P = C - uv$, so that P is a path from u to v of length ≥ 3. Since I is a uv-separator, P must contain a vertex $s \in I$. Then either su or sv is a chord of P, which means C has a chord. Hence, $G + K_r$ is a chordal graph.

(ii) Let an edge $xy \notin T$ where $u \in K_x, v \in K_y$ for $u, v \in p$, then there exists the path P from x to y in T and a minimal separator I_j of G, containing in this path. Suppose to the contrary that there exists any node z in a clique tree T, such that $K_r \cap I_z = \varnothing$, which is contained in the path P. Let $I_z = \{u', v'\}$ and $K_z \cap K_x = u', K_z \cap K_y = v'$, then $u', v' \notin K_r$. Since $uv \in G + K_r$ and there exist the edges $uu' \in K_x, u'v' \in K_z, v'v \in K_y$, $G + K_r$ has a chordless cycle (u, u', v', v). We get a contradiction. □

Insert Query(K_r)

If the conditions of Theorem 1 are satisfied, return "yes", otherwise return "no".

End Insert Query

We next show how to update a clique tree for $G + K_r$.

Insert(K_r)

1. Consider such edges of $G + K_r$ that $uv \notin G$ with $u \in K_x, v \in K_y$ such that $u, v \in K_r$ for any $xy \in T$. If such edges do not exist in $G + K_r$, we pass to item 2. Otherwise, let $I = K_x \cap K_y$ then $K = I \cup \{u, v\}$ is a clique in $G + K_r$. As K is not a clique of G, we must add to a new node z with $K_z = K$. We must consider whether cliques K_x, K_y are maximal in $G + K_r$. Since $v \notin K_x$, then $K_x \subset K_z$ iff $K_x = I \cup \{u\}$ iff $|K_x| = |I| + 1$. Similarly, $u \notin K_y$, then $K_y \subset K_z$ iff $K_y = I \cup \{v\}$ iff $|K_y| = |I| + 1$. Thus, comparing $|K_x|, |K_y|$ and $w(x, y) = |I|$ we determine, whether cliques K_x and K_y is maximal in $G + K_r$.

   Replace xy in T with a new node z representing $K_z = I \cup \{u, v\}$ and add xz, yz, each with weight $|I| + 1$. Determine whether cliques K_x, K_y are maximal in $G+K_r$. If K_x, K_y are maximal then we pass to item 2. Otherwise, if K_x is not the maximal clique, remove xz and replace x with z, if K_y is not the maximal clique, remove yz and replace y with z.

2. Add a new node r to T corresponding to K_r. Connect r with other nodes i such that $K_i \cap K_r \neq \varnothing$, attribute to it weights $w(i, r)$. Moreover, if $K_w \subset K_i$ for some $w \in T$ then K_w is not maximal clique in $G + K_r$ and we must remove w from T.

End Insert

We will now examine a deletion of a complete r-vertex graph K_r.

Theorem 2. *Let G be a chordal graph which contains a complete r-vertex graph K_r. Then $G - K_r$ is chordal iff the following conditions are satisfied:*

(i) the edge $uv \in K_r$ is contained exactly in two maximal cliques of G;
(ii) G does not contain any cycle consisting of vertices of the set $I_r = K_r \cap N(K_r)$.

Proof. (i) It is known that $uv \in K_r$, i.e. K_r is one of the maximal cliques containing this edge. Then uv must be contained exactly in the one clique of G except K_r. Suppose to the contrary that $uv \in q$ is contained in two cliques $\{u, v, s\}$ and $\{u, v, t\}$, where $st \notin G$, which are different from K_r. Then these two cliques cannot be contained in one maximal clique of G. In this case the deletion of K_r leads to the appearance of a chordless cycle (u, s, v, t) in $G - K_r$. We get a contradiction.

(ii) Suppose to the contrary that I_r forms a cycle C. Note that I_r forms a cycle iff it contains all vertices of a complete r-vertex graph K_r. Consider a case when $|N(Kr)| \geq 2$. Let $K_x, K_y \in N(K_r), xy \in T$ and $I_r = (K_x \cap K_r, K_y \cap K_r)$. Since I_r forms a cycle C, it is clear that $K_x \cap K_y \neq \varnothing$. By the definition of a chordal graph, all the cycles of G have length 3. The deletion of K_r leads to the disappearance of a third edge for each clique $N(K_r)$. Since these cliques are connected between themselves by the common vertices or edges, the cycle of length $2 \cdot |N(K_r)|$ appears in $G - K_r$. It means that $G - K_r$ has a cycle with length ≥ 4. We get a contradiction. □

Delete Query(K_r)

If the conditions of Theorem 2 are satisfied, return "yes", otherwise return "no".

End Delete Query

We show how to update a clique tree for $G - K_r$.

Delete(K_r)

1. Consider edges uv of G such that $uv \notin G - K_r$, with $u \in K_x, v \in K_y$ and $u, v \in K_r$ for any $xy \in T$. If such edges do not exist in $G - K_r$, pass to item 2. Otherwise, T of G contains a node z corresponding $K_z = K$ (see item 1

of Insert for the definition of K_z). In $G - K_r$, the maximal clique K_z has split into the cliques $K_z^u = K_z - \{v\}$ and $K_z^v = K_z - \{u\}$ which may not be maximal.

Divide the set $N(K_z)$ into $N_u = \{x \in N(z) \mid u \in K_x\}, N_v = \{y \in N(z) \mid v \in K_y\}$ and $N_w = \{w \in N(z) \mid u, v \notin K_w\}$. Then K_z^u is not maximal in $G - K_r$ iff $\exists x \in N_u$ such that $K_z^u \subset K_x$ and $w(x, z) = k - 1$. Similarly, K_z^v is not maximal in $G - K_r$ iff $\exists y \in N_v$ such that $K_z^v \subset K_y$ and $w(y, z) = k - 1$.

Replace z with two nodes z_1 and z_2 respectively representing K_z^u and K_z^v and add the edge z_1z_2 with weight $w(z_1, z_2) = k - 2$. If $x \in N_u$, replace xz with xz_1. If $y \in N_v$, replace yz with yz_2. If $w \in N_w$, replace zw with z_1w or z_2w.

If K_z^u and K_z^v are maximal cliques then pass to item 2. Otherwise, if K_z^u is not maximal because $K_z^u \subset K_{x_i}$ for some $x_i \in N_u$ then remove x_iz_1 and replace z_1 with x_i. Similarly, if K_z^v is not maximal because $K_z^v \subset K_{y_i}$ for some $y_i \in N_v$ then remove y_iz_2 and replace z_2 with y_i.

2. Remove r corresponding K_r from T.

End Delete

Corollary 1. *If $I_r = K_r \cap N(K_r)$ forms two or more different paths P_i, then K_r is a separator of G.*

Proof. Let P_1 and P_2 be two paths formed by $I_r = K_r \cap N(K_r)$. Let $K_x, K_y \in N(K_r)$ and $I_{r'} = K_x \cap K_r, I_{r''} = K_y \cap K_r$. Assume that $I_{r'} \subset P_1$ and $I_{r''} \subset P_2$. Then we have $K_x \cap K_y = \varnothing$. It means that deleting K_r leads to the appearance of two connected components, where cliques K_x and K_y are contained in the different connected components. Hence K_r is a separator of graph G. □

We use a clique tree T of a chordal graph G for performing the described operations. Since T has at most n nodes, each operation runs in $O(n)$ time.

4 Conclusions

In this paper, we described a fully dynamic algorithm, which considers new modifications of graphs, i.e. insertions or deletions of complete r-vertex graph, where $r \geq 3$. The proposed algorithm could be a suitable addition to the algorithm of Ibarra [10] for the maintenance of chordal graphs. Also, if it is known that the edges which should be added to the input graph G form a clique, then we are able to implement the algorithm more efficiently than if we were to add or delete the edges one by one.

Acknowledgment. The author would like to thank V.A. Evstegneev for the statement of the problem and I.B. Virbitskaite for her helpful comments and advice.

References

1. J. R. S. Blair and B. Peyton. *An introduction to chordal graphs and clique trees.* In Graph Theory and Sparse Matrix Computation, volume 56 of IMA, pp. 1-29. Ed.A.George and J.R.Gilbert and J.W.H.liu), Springer, 1993.

2. C. Beeri, R. Fagin, D. Maier and M. Yannakasis. *On the desirability of acyclic database schemes.* Journal of the ACM, 30:479-513, 1983.
3. P. Buneman. *A characterization of rigid circuit graphs.* Discrete Mathematics, 9:205-212, 1974.
4. L. Ibarra. *Fully dynamic algorithms for chordal graphs.* In Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99), SIAM, Philadelphia, 1999, pp.923-924.
5. A. Berry, A. Sigayret, and J. Spinrad. *Faster dynamic algorithms for chordal graphs, and an application to Phylogeny.* In 31st Int. Workshop on Graph Theoretical Concepts in Computer Science (WG05), number 3787 in Lecture Notes in Computer Science, pp.445-455.
6. P. Hell, R. Shamir, and R. Sharan. *A fully dynamic algorithm for recognizing and representing proper interval graphs.* in Proceedings of t he 7th Annual European Symposium on Algorithms, Lecture Notes in Computer Science 1643, Springer-Verlag, 1999, pp. 527-539.
7. A. Berry, P. Heggernes and Y. Villanger. *A vertex incremental approach for dynamically maintaining chordal graphs.* Discrete Mathematics, 3063 (2006), pages 318-336.
8. F. Gavril. *The intersection graphs of subtrees in trees are exactly the chordal graphs.* J.Combinatorial Theory B, 1974, 16: pp.47-56.
9. J. R. Walter. *Representations of chordal graphs of a tree.* J.Graph Theory, 1978, 2: pp.265-267.
10. C. Ho and R. C. T. Lee. *Counting clique trees and computing perfect elimination schemes in parallel.* Information Processing Letters, 1989, 31: pp.61-68.
11. M. Lundquist. *Zero patterns, chordal graphs, and matrix completions.* PhD thesis, Dept.of Mathematical Sciences, Clemson University, 1990.