Information Processing Letters 124 (2017) 21–25

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Computing the clique-separator graph for an interval graph in linear time

Louis Ibarra

U.S. Dept. of Defense, Fort Meade, MD 20755, USA

ARTICLE INFO

Article history:
Received 16 November 2016
Accepted 10 April 2017
Available online 20 April 2017
Communicated by R. Uehara

Keywords:
Graph algorithms
Interval graphs
Chordal graphs

ABSTRACT

We present an algorithm to compute the clique-separator graph of an interval graph in $O(m+n)$ time. This improves the running time of $O(n^2)$ given in [11]. The algorithm is simple and uses no complicated data structures.

Published by Elsevier B.V.

1. Introduction

Interval graphs model many problems involving linear arrangements and consequently they have applications in a number of areas, including archeology, computational biology, file organization, partially ordered sets, and psychology [2,7,16]. Many well-known NP-complete graph problems can be solved on interval graphs in polynomial time [13]. Interval graphs can be recognized in $\Theta(m+n)$ time using PQ-trees [1], simplified versions of PQ-trees [14, 10], modular decomposition [9], or a vertex ordering computed with multiple passes of Lex-BFS [5]. Interval graphs are a subclass of chordal graphs.

A dynamic graph algorithm maintains a solution to a graph problem as the graph undergoes a series of small changes, such as single edge deletions or insertions. For each change, the algorithm updates the solution faster than recomputing the solution from scratch, i.e., with no previously computed information. Ibarra [12] presented the first dynamic graph algorithm for recognizing interval graphs, which runs in $O(n \log n)$ time per edge deletion or edge insertion, where n is the number of vertices in the graph. (All the running times in this paper are worst-case.)

The algorithm in [12] is based on the clique-separator graph for chordal graphs [11]. Given a chordal graph G, its clique-separator graph $\mathcal{G}$ is a graph whose nodes are the maximal cliques and minimal vertex separators of G and whose (directed) arcs and (undirected) edges represent the containment relations between the sets corresponding to the nodes. The clique-separator graph has various structural properties when G is a chordal graph, interval graph, proper interval graph, or split graph [11]. The clique-separator graph is different from other structures with similar names; see [11] for a discussion.

The clique-separator graph can be constructed in $O(n^3)$ time if G is a chordal graph, in $O(n^2)$ time if G is an interval graph, and in $O(m+n)$ time if G is a proper interval graph [11]. In this paper, we show that the clique-separator graph of an interval graph can be constructed in $O(m+n)$ time. This reduces the preprocessing time for the algorithm in [12] from $O(n^2)$ to $O(m+n)$. The algorithm is simple and uses no complicated data structures.

In Section 2 we review definitions and previous results. In Section 3 we present the algorithm. In Section 4 we present conclusions.

E-mail address: lwibarr@tycho.ncsc.mil.

http://dx.doi.org/10.1016/j.ipl.2017.04.007
0020-0190/Published by Elsevier B.V.

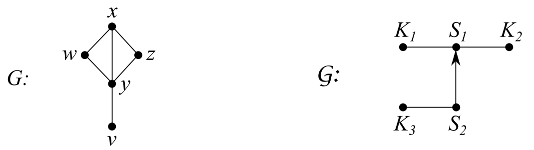

Fig. 1. Interval graph G and its clique-separator graph $\mathcal{G}$.

2. Preliminaries

2.1. Chordal graphs and interval graphs

Let $G = (V, E)$ be a simple, undirected graph and let $n = |V|$ and $m = |E|$. A *clique* of G is a set of pairwise adjacent vertices of G. A *maximal clique* of G is a clique of G that is not properly contained in any clique of G. Fig. 1 shows a graph G with maximal cliques $\{w, x, y\}, \{x, y, z\}$, and $\{y, v\}$.

For a set $U \subseteq V$, the subgraph of G *induced by* U is $G[U] = (U, E(U))$, where $E(U) = \{\{u, v\} \in E \mid u, v \in U\}$. Let $S \subset V$. $G - S$ denotes $G[V - S]$. For $u, v \in V$, S is a uv*-separator* of G if u and v are connected in G and not connected in $G - S$. S is a *minimal vertex separator* of G if for some $u, v \in V$, S is a uv-separator of G that does not properly contain any uv-separator of G. Fig. 1 shows a graph G with minimal vertex separators $\{x, y\}$ and $\{y\}$.

A graph is *chordal* (or *triangulated*) if every cycle of length greater than 3 has a *chord*, which is an edge joining two nonconsecutive vertices of the cycle. A graph G is chordal if and only if G has a *clique tree*, which is a tree T on the maximal cliques of G with the *clique intersection property*: for any two maximal cliques K and K', the set $K \cap K'$ is contained in every maximal clique on the K-K' path in T [3,7]. The clique tree is not necessarily unique. We refer to the *vertices* of G and the *nodes* of T.

For a connected chordal graph G with clique tree T, S is a minimal vertex separator of G if and only if $S = K \cap K'$ for some edge $\{K, K'\}$ of T [8,15]. Then the nodes and edges of T correspond to the maximal cliques and minimal vertex separators of G, respectively. A maximal clique corresponds to exactly one node of T, whereas a minimal vertex separator may correspond to multiple edges of T. A chordal graph has at most n maximal cliques [6] and thus at most $n - 1$ minimal vertex separators.

Blair and Peyton [3] discuss various properties of clique trees, including the following result from Ho and Lee [8] and Lundquist [15].

Theorem 1 *([3]). Let G be a connected chordal graph with clique tree T. For an edge $\{K_i, K_j\}$ of T, let $S = K_i \cap K_j$. Let $\mathcal{K}_i$ and $\mathcal{K}_j$ be the node sets of the trees obtained by deleting edge $\{K_i, K_j\}$ from T. Let $V_i = (\cup_{K \in \mathcal{K}_i} K) - S$ and $V_j = (\cup_{K \in \mathcal{K}_j} K) - S$. Then S is a uv-separator for every pair of vertices $u \in V_i$ and $v \in V_j$.*

An *interval graph* is a graph where each vertex can be assigned an interval on the real line such that two vertices are adjacent if and only if their assigned intervals intersect; such an assignment is an *interval representation*. A graph is an interval graph if and only if it has a *clique path*, which is a clique tree that is a path [6]. A clique path P has the clique intersection property: for any two maximal cliques K and K', the set $K \cap K'$ is contained in every maximal clique on the K-K' subpath of P.

Interval graphs are a proper subclass of chordal graphs. See [2,7,16] for more information about these graph classes.

2.2. Clique-separator graph

Let G be a chordal graph. The *clique-separator graph* $\mathcal{G}$ of G has the following nodes and edges.

- $\mathcal{G}$ has a *clique node* K for each maximal clique K of G and a *separator node* S for each minimal vertex separator S of G.
- Each arc is from a separator node to a separator node. $\mathcal{G}$ has arc (S, S'') if $S \subset S''$ and there is no separator node S' such that $S \subset S' \subset S''$.
- Each edge is between a separator node and a clique node. $\mathcal{G}$ has edge $\{S, K\}$ if $S \subset K$ and there is no separator node S' such that $S \subset S' \subset K$.

We refer to the *vertices* of G and the *nodes* of $\mathcal{G}$. We identify "maximal clique" and "clique node" and identify "minimal vertex separator" and "separator node". Fig. 1 shows an interval graph G with clique-separator graph $\mathcal{G}$, which has clique nodes $K_1 = \{w, x, y\}, K_2 = \{x, y, z\}$, and $K_3 = \{y, v\}$ and separator nodes $S_1 = \{x, y\}$ and $S_2 = \{y\}$.

We will use the following property of an interval graph, which lets us test whether a node contains another node (viewing the nodes as sets of vertices) in $\Theta(1)$ time. Given an interval representation of G and a set $U \subseteq V$, *Intersect*(U) is the intersection of the intervals assigned to the vertices in U.

Theorem 2 *([11, Theorem 12]). Let G be an interval graph with clique-separator graph $\mathcal{G}$. For any separator node S and any node N of $\mathcal{G}$, $S \subset N$ if and only if Intersect$(N) \subset$ Intersect(S).*

3. Algorithm

Given a connected interval graph G, compute a clique path $P = (K_1, K_2, \ldots, K_l)$ of G with an algorithm such as [9]. (If G is not connected, we apply the algorithm to each connected component.) Compute an interval representation of G by assigning to each vertex v the interval $[i, j]$, where i and j are the smallest and largest indices of the clique nodes containing v. Compute the *augmented clique path* $\hat{P}$ from P by inserting the separator node $K_i \cap K_{i+1}$ between each consecutive pair of clique nodes K_i and K_{i+1}. For each node N of $\hat{P}$, compute *Intersect*(N). All of these computations require $\Theta(m + n)$ time.

The path $\hat{P}$ may have duplicate separator nodes and we will use the following result to detect them.

Theorem 3. *Let G be a connected interval graph with augmented clique path $\hat{P}$. For any two separator nodes S and S' of $\hat{P}$, Intersect$(S) =$ Intersect(S') if and only if $S = S'$.*

Proof. ($\Leftarrow$) Immediate. ($\Rightarrow$) We prove the contrapositive. Suppose $S \neq S'$. If $S' \subset S$, then by Theorem 2,