Information Processing Letters 111 (2011) 508–511

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A simple algorithm to generate the minimal separators and the maximal cliques of a chordal graph ☆

Anne Berry *, Romain Pogorelcnik

LIMOS UMR CNRS 6158, Ensemble Scientifique des Cézeaux, F-63 173 Aubière, France

ARTICLE INFO

Article history:
Received 15 February 2010
Received in revised form 13 October 2010
Accepted 20 February 2011
Available online 3 March 2011
Communicated by W.-L. Hsu

Keywords:
Graph algorithms
LexBFS
MCS
Minimal separator
Maximal clique
Moplex ordering

ABSTRACT

We present a simple unified algorithmic process which uses either LexBFS or MCS on a chordal graph to generate the minimal separators and the maximal cliques in linear time in a single pass.

1. Introduction

Chordal graphs (which are the graphs with no chordless cycle on four or more vertices), were characterized by Fulkerson and Gross [7] as the graphs for which one can repeatedly find a simplicial vertex (a vertex whose neighborhood is a clique) and remove it from the graph, until no vertex is left; this process, called *simplicial elimination*, defines an ordering α on the vertices called a *perfect elimination ordering* (*peo*). ($\alpha(i)$ denotes the vertex bearing number i, and $\alpha^{-1}(x)$ the number of vertex x.) At each step of the elimination process, a new *transitory graph* is defined.

Rose [9] showed that for any given peo α of a chordal graph, any minimal separator of the graph is defined in the course of the simplicial elimination process as the transitory neighborhood of some vertex. Moreover, it is easy to see that in any maximal clique K, the vertex y of smallest number by α defines K as its transitory closed neighborhood. In the rest of the paper, we will call such vertices *generators*, and our goal will be to detect these generators efficiently. We will use graph search algorithms Lexicographic Breadth-First Search (LexBFS) [10] and Maximum Cardinality Search (MCS) [11], both tailored to generate a peo.

The idea behind these algorithms is simple. Both algorithms number the vertices from n to 1 (n is the number of vertices of the graph). Each vertex x bears a label, which is the list of numbers of the neighbors of x with a higher number (LexBFS) or the cardinality of this list (MCS). At each step, a new vertex of maximum label is chosen to be numbered. If the label of this new vertex x_i is not larger than the label of the previous vertex x_{i+1}, then we show that x_i generates a minimal separator of the graph. The vertices of this minimal separator are the already numbered neighbors of x_i. We also show that the previously numbered vertex, x_{i+1}, generates a maximal clique: x_{i+1}, together with its already numbered neighbors (excluding x_i), define a maximal clique of the graph.

Thus the labels of algorithms LexBFS and MCS enable the user to detect the generators as soon as they are numbered. The minimal separators and maximal cliques can be

☆ Research supported by the French Agency for Research under the DEFIS program TODO, ANR-09-EMER-010.

* Corresponding author.
E-mail addresses: berry@isima.fr (A. Berry), romain.pogorelcnik@isima.fr (R. Pogorelcnik).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2011.02.013

found with a single pass, without requiring a preliminary pass of the algorithm to number all the vertices.

These results have been proved for MCS in two separate papers. Blair and Peyton [5], while studying how MCS defines a clique tree of a chordal graph, showed how to generate the maximal cliques (which are the nodes of the clique tree), using the MCS labels. Kumar and Madhavan [8] showed how to generate the minimal separators of a chordal graph, given an MCS ordering.

However, although a minimal separator and a maximal clique generator are actually detected at the same step of the algorithm, these results have not been unified. Moreover, Kumar and Madhavan [8] use a peo as input, but the minimal separators as well as the maximal cliques can be computed during the execution, which may be an important feature when handling very large graphs, since a global pre-numbering of the vertices is not necessary or even useful.

LexBFS, as we will show, exhibits the same property as MCS regarding these generators. Our aim is to present a simple unified single-pass algorithm which generates the minimal separators and the maximal cliques of a chordal graph.

2. Preliminaries

In the rest of the paper, we will consider all graphs to be connected. Symbol $+$ denotes disjoint union. The *neighborhood* $N_G(x)$ of vertex x in graph G is $N_G(x) = \{y \neq x \mid xy \in E\}$ The *neighborhood* $N_G(X)$ of a vertex set $X \subset V$ is $N_G(X) = \bigcup_{x \in X} N_G(x) - X$. A *clique* is a set of pairwise adjacent vertices. A clique $X \subseteq V$ is *maximal* if $\forall y \in V - X$, $X + \{y\}$ fails to be a clique. A subset X of vertices is called a *module* if the vertices of X share the same external neighborhood: $\forall x \in X, N(x) - X = N(X)$.

A subset S of vertices of a connected graph G is called a *separator* if $G(V - S)$ is not connected. A separator S is called an *xy-separator* if x and y lie in different connected components of $G(V - S)$, a *minimal xy-separator* if S is an xy-separator and no proper subset of S is an xy-separator. A separator S is a *minimal separator* if there is some pair $\{x, y\}$ such that S is a minimal xy-separator. Alternately, S is a minimal separator if and only if $G(V - S)$ has at least two connected components C_1 and C_2 such that $N(C_1) = N(C_2) = S$; such components are called *full components* of S.

3. Main theorem

Definition 3.1. Given a chordal graph G and a peo α of G, for any vertex x, *Madj*(x) is the set of neighbors of x with a number higher than that of x: *Madj*$(x) = \{y \in N(x) \mid \alpha^{-1}(y) > \alpha^{-1}(x)\}$.

Definition 3.2. Given a chordal graph G and a peo α of G, we will call:

- *minimal separator generator* any vertex x_i with number i by α, such that *Madj*(x_i) is a minimal separator of G and *label*$(x_i) \leqslant$ *label*(x_{i+1}), where $x_{i+1} = \alpha(i + 1)$;
- *maximal clique generator* a vertex y such that *Madj*$(y) \cup \{y\}$ is a maximal clique of G.

Theorem 3.3. *Let α be a peo defined by either LexBFS or MCS, let x_i be the vertex with number i, let x_{i+1} be the vertex with number $i + 1$.*

a) *x_i is a minimal separator generator if and only if label$(x_i) \leqslant$ label(x_{i+1}).*
b) *x_{i+1} is a maximal clique generator if and only if label$(x_i) \leqslant$ label(x_{i+1}) or $i + 1 = 1$.*

We will now discuss the proof of Theorem 3.3. We will first present moplex elimination, which is a process that explains how both MCS and LexBFS scan the minimal separators and the maximal cliques of a chordal graph, then prove Theorem 3.3 for LexBFS.

3.1. Moplex elimination

Definition 3.4. (See [1].) A *moplex* is a clique X such that X is a module and $N(X)$ is a minimal separator. We extend this definition to a clique whose neighborhood is empty. A *simplicial moplex* is a moplex whose vertices are all simplicial.

Property 3.5. *(See [1].) Any chordal graph which is not a clique has at least two non-adjacent simplicial moplexes.*

From this was derived a variant of the characterization of Fulkerson and Gross for chordal graphs by simplicial elimination of vertices:

Characterization 3.6. (See [1].) A graph is chordal if and only if one can repeatedly delete a simplicial moplex until the graph is a clique (which we will call the terminal moplex). We call this process *moplex elimination*.

Note that moplex elimination on a chordal graph is a special case of simplicial elimination, since at each step a set of simplicial vertices is eliminated. Note also that for a connected graph G, the transitory elimination graph obtained at the end of each step remains connected.

Moplex elimination defines an ordered partition $(X_1, X_2, \ldots, X_k)$ of the vertices of the graph into the successive moplexes which are defined in the successive transitory elimination graphs. We will call this partition a *moplex ordering*.

Theorem 3.7. [1] *Let G be a chordal graph, let $(X_1, X_2, \ldots, X_k)$ be a moplex ordering of G. At each step $i < k$ of the elimination process finding moplex X_i in transitory graph G_i,*

- *$N_{G_i}(X_i)$ is a minimal separator of G.*
- *$X_i \cup N_{G_i}(X_i)$ is a maximal clique of G.*

The terminal moplex X_k is a maximal clique.

There are no other minimal separators or maximal cliques in G.

[1] The results proved in this subsection were briefly presented in [2].