ARTICLE IN PRESS

Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Eigenvalue location in cographs[☆]

David P. Jacobs [a], Vilmar Trevisan [b,*], Fernando Colman Tura [c]

[a] *School of Computing, Clemson University Clemson, SC 29634, USA*
[b] *Instituto de Matemática, UFRGS, 91509–900 Porto Alegre, RS, Brazil*
[c] *Departamento de Matemática, UFSM, 97105–900 Santa Maria, RS, Brazil*

ARTICLE INFO

Article history:
Received 22 October 2015
Received in revised form 8 February 2017
Accepted 13 February 2017
Available online xxxx

Keywords:
Cograph
Adjacency matrix
Eigenvalue

ABSTRACT

We give an $O(n)$ time and space algorithm for constructing a diagonal matrix congruent to $A + xI$, where A is the adjacency matrix of a cograph and $x \in \mathbb{R}$. Applications include determining the number of eigenvalues of a cograph's adjacency matrix that lie in any interval, obtaining a formula for the inertia of a cograph, and exhibiting infinitely many pairs of equienergetic cographs with integer energy.

1. Introduction

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. For $v \in V$, $N(v)$ denotes the *open neighborhood* of v, that is, $\{w|\{v, w\} \in E\}$. The *closed neighborhood* $N[v] = N(v) \cup \{v\}$. If $|V| = n$, the *adjacency matrix* $A = [a_{ij}]$ is the $n \times n$ matrix of zeros and ones such that $a_{ij} = 1$ if and only if v_i is adjacent to v_j (that is, there is an edge between v_i and v_j). A value λ is an *eigenvalue* if $\det(A - \lambda I) = 0$, and since A is real symmetric its eigenvalues are real. In this paper, a graph's *eigenvalues* are the eigenvalues of its adjacency matrix.

This paper is concerned with *cographs*. The notion of a cograph under the name decomposable graphs was introduced by Kelmans in the 1960's [14,15] and this class of graphs has been discovered independently by several authors in many equivalent ways since then. Corneil, Lerchs and Burlingham [6] define cographs recursively:

(1) A graph on a single vertex is a cograph;
(2) A finite union of cographs is a cograph;
(3) The complement of a cograph is a cograph.

A graph is a cograph if and only if it has no induced path of length four [6]. They are often simply called P_4 *free* graphs in the literature. Linear time algorithms for recognizing cographs are given in [7] and more recently in [10].

While recognition algorithms for cographs are an interesting problem, our motivation for considering cographs comes from *spectral graph theory* [4,8]. Spectral properties of cographs were studied by Royle in [18] where the surprising result was obtained that the rank of a cograph is the number of non-zero rows in the adjacency matrix. An elementary proof of this property was later given in [5]. More recently, in [2] Bıyıkoğlu, Simić and Stanić obtained the multiplicity of −1 and 0 for cographs.

The purpose of this paper is to extend to cographs eigenvalue location algorithms that exist for trees [11], threshold graphs [12] and generalized lollipop graphs [9]. Recall that two real symmetric matrices R and S are *congruent* if there exists

☆ Work supported by Science without Borders CNPq — Grant 400122/2014-6, Brazil.
* Corresponding author.
E-mail addresses: dpj@clemson.edu (D.P. Jacobs), trevisan@mat.ufrgs.br (V. Trevisan), ftura@smail.ufsm.br (F.C. Tura).

http://dx.doi.org/10.1016/j.dam.2017.02.007
0166-218X/

Please cite this article in press as: D.P. Jacobs, et al., Eigenvalue location in cographs, Discrete Applied Mathematics (2017), http://dx.doi.org/10.1016/j.dam.2017.02.007

a nonsingular matrix P for which $R = P^TSP$. Our main focus is an algorithm that uses $O(n)$ time and space for constructing a diagonal matrix congruent to $A + xI$, where A is adjacency matrix of a cograph, and $x \in \mathbb{R}$. Our paper is similar in spirit to the papers [11,12] which describe $O(n)$ diagonalization algorithms for trees and for threshold graphs. Threshold graphs are P_4, C_4, and $2K_2$ free, and therefore are a subclass of cographs. Hence our algorithm is an extension of the algorithm in [12].

Several points are worth noting. First, while one might expect linear time algorithms for graphs with *sparse* adjacency matrices such as trees, the adjacency matrix of a cograph can be *dense*. Next, while our algorithm's correctness is based on elementary matrix operations, its implementation operates directly on the cotree and uses only $O(n)$ space. Finally, the *analysis* of algorithms for trees and threshold graphs has led to interesting theoretical results. For example, in [17] conditions were determined for the index (largest eigenvalue) in trees to be integer. In [12] the authors showed that all eigenvalues of threshold graphs, except -1 and 0, are simple. In [13] the algorithm was used to show that no threshold graphs have eigenvalues in $(-1, 0)$.

If G is a graph having eigenvalues $\lambda_1, \ldots, \lambda_n$, its *energy*, denoted $E(G)$ is defined to be $\sum_{i=1}^{n} |\lambda_i|$. Two non-cospectral graphs with the same energy are called *equienergetic*. Finding non-cospectral equienergetic graphs is a relevant problem. In [13] the authors presented infinite sequences of connected, equienergetic pairs of non-cospectral threshold graphs with integer energy. In this paper, we continue this investigation.

Here is an outline of the remainder of this paper. In Section 2 we describe cotrees, and present some known facts. In Section 3 we give the elementary matrix operations used in our algorithm. In Section 4 we give the complete diagonalization algorithm. In Section 5, using Sylvester's Law of Inertia, we show how to efficiently determine how many eigenvalues of a cograph lie in a given interval. The *inertia* of a graph G is the triple (n_+, n_0, n_-) giving the number of eigenvalues of G that are positive, zero, and negative, and in Section 6 we give a formula for cograph inertia. Finally in Section 7 we exhibit infinitely many non-threshold cographs equienergetic to a complete graph.

2. Cotrees and adjacency matrix

Cographs have been represented in various ways, and it is useful to recall the representation given in [6]. The unique *normalized form* of a cograph G is defined recursively: If G is connected, then it is in normalized form if it is expressed as a single vertex, or the *complemented union* of $k \geq 2$

$$G = \overline{G_1 \cup G_2 \cup \cdots \cup G_k}$$

connected cographs G_i in normalized form. If G is disconnected its normalized form is the complement of a connected cograph in normalized form. The unique rooted tree T_G representing the parse structure of the cograph's normalized form is called the *cotree*. The leaves or terminal vertices of T_G correspond to vertices in the cograph. The interior nodes represent $\overline{\cup}$ operations.

It is not difficult to show that the class of cographs is also the smallest class of graphs containing K_1, and closed under the union $\cup$ and join $\otimes$ operators. In fact one can transform the cotree of Corneil, Lerchs and Burlingham into an equivalent tree T_G using $\cup$ and $\otimes$. In the connected case, we simply place a $\otimes$ at the tree's root, placing $\cup$ on interior nodes with odd depth, and placing $\otimes$ on interior nodes with even depth. To build a cotree for a disconnected cograph, we place $\cup$ at the root, and place $\otimes$'s at odd depths, and $\cup$'s at even depths. It will be convenient for us to use this unique alternating representation. In [2] this structure is called a *minimal cotree*, but throughout this paper we call it simply a *cotree*. All interior nodes of cotrees have at least two children. Fig. 1 shows a cograph and cotree. The following is well known.

Lemma 1. *If G is a cograph with cotree T_G, vertices u and v are adjacent in G if and only if their least common ancestor in T_G is $\otimes$.*

Two vertices u and v are *duplicates* if $N(u) = N(v)$ and *coduplicates* if $N[u] = N[v]$. We call u and v *siblings* if they are either duplicates or coduplicates. Siblings play an important role in the structure of cographs, as well as in this paper.

Lemma 2. *Two vertices v and u in a cograph are siblings if and only if they share the same parent w node in the cotree. Moreover, if $w = \cup$, they are duplicates. If $w = \otimes$ they are coduplicates.*

Lemma 3. *A cograph G of order $n \geq 2$ has a pair of siblings.*

Proof. The cotree of G must have an interior vertex adjacent to two leaves. □

Let G be a cograph with cotree T_G. Let $G - v$ denote the subgraph obtained by removing v. It is known that $G - v$ is a cograph, so we shall use $T - v$ to denote the cotree of $G - v$. There is a general method for constructing $T - v$ [6, Lem. 1]. However, it somewhat simplifies the process if v has maximum depth. The following lemma can be proved with Lemma 1.

Lemma 4. *Let T_G be a cotree, and let $\{v, u\}$ be siblings of greatest depth, whose parent w has k children. If $k > 2$ we obtain $T - v$ by removing v. If $k = 2$ and w is not the root, we obtain $T - v$ by moving u to the parent of w, and removing v and w. If $k = 2$ and w is the root, the cotree is u.*

We end this section by making an important observation.

Please cite this article in press as: D.P. Jacobs, et al., Eigenvalue location in cographs, Discrete Applied Mathematics (2017), http://dx.doi.org/10.1016/j.dam.2017.02.007