Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Efficient computation of the characteristic polynomial of a threshold graph

Martin Fürer

Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 26 October 2015
Received in revised form 11 July 2016
Accepted 13 July 2016
Available online xxxx

Keywords:
Efficient algorithms
Threshold graphs
Characteristic polynomial

ABSTRACT

An efficient algorithm is presented to compute the characteristic polynomial of a threshold graph. Threshold graphs were introduced by Chvátal and Hammer, as well as by Henderson and Zalcstein in 1977. A threshold graph is obtained from a one vertex graph by repeatedly adding either an isolated vertex or a dominating vertex, which is a vertex adjacent to all the other vertices. Threshold graphs are special kinds of cographs, which themselves are special kinds of graphs of clique-width 2. We obtain a running time of $O(n \log^2 n)$ for computing the characteristic polynomial, while the previously fastest algorithm ran in quadratic time. We improve the running time drastically in the case where there is a small number of alternations between 0's and 1's in the sequence defining a threshold graph.

1. Introduction

The characteristic polynomial of a graph $G = (V, E)$ is defined as the characteristic polynomial of its adjacency matrix A, i.e., $\chi(G, \lambda) = \det(\lambda I - A)$. The characteristic polynomial is a graph invariant, i.e., it does not depend on the enumeration of the vertices of G. The complexity of computing the characteristic polynomial of a matrix is the same as that of matrix multiplication [16,19] (see [3, Chap. 16]), which has recently improved from $O(n^{2.37548})$ [6] to $O(n^{2.37369})$ [7,29] and $O(n^{2.37287})$ [11]. For special classes of graphs, we expect to find faster algorithms for the characteristic polynomial. Indeed, for trees, a chain of improvements [27,18] resulted in an $O(n \log^2 n)$ time algorithm [10]. The determinant and rank of the adjacency matrix of a tree can even be computed in linear time [8]. For threshold graphs (defined below), Jacobs et al. [14] have recently designed an $O(n^2)$ time algorithm to compute the characteristic polynomial. Here, we improve the running time to $O(n \log^2 n)$. As usual, we use the algebraic complexity measure, where every arithmetic operation counts as one step. Throughout this paper, $n = |V|$ is the number of vertices of G.

Threshold graphs [5,12] are defined as follows. Given n and a sequence $b = (b_1, \ldots, b_{n-1}) \in \{0, 1\}^{n-1}$, the threshold graph $G_b = (V, E)$ is defined by $V = [n] = \{1, \ldots, n\}$, and for all $i < j$, $\{i, j\} \in E$ if and only if $b_i = 1$. Thus G_b is constructed by an iterative process starting with the initially isolated vertex n. In step $j > 1$, vertex $n - j + 1$ is added. At this time, vertex j is isolated if b_j is 0, and vertex j is adjacent to all other (already constructed) vertices $\{j + 1, \ldots, n\}$ if $b_j = 1$. It follows immediately that G_b is isomorphic to $G_{b'}$ if and only if $b = b'$. G_b is connected if $b_1 = 1$, otherwise vertex 1 is isolated. An example of a threshold graph is given in Fig. 1. Usually, the order of the vertices being added is $1, 2, \ldots, n$ instead of $n, n - 1, \ldots, 1$. We choose this unconventional order to simplify our main algorithm.

Threshold graphs have been widely studied and have several applications from combinatorics to computer science and psychology [17]. The index of a graph is its largest eigenvalue. Threshold graphs play an important role in the search for

E-mail address: furer@cse.psu.edu.

http://dx.doi.org/10.1016/j.tcs.2016.07.013
0304-3975/© 2016 Published by Elsevier B.V.

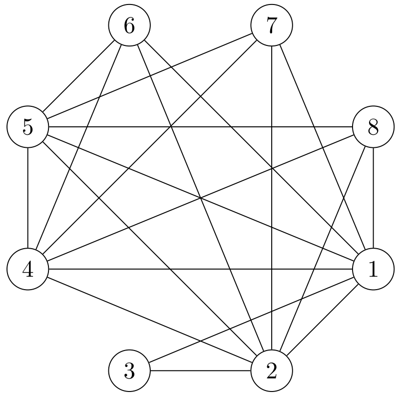

Fig. 1. The threshold graph defined by the sequence $b = (b_1, \ldots, b_7) = (1, 1, 0, 1, 1, 0, 0)$ in our notation. In standard notation, vertex j would be numbered $9 - j$.

graphs of maximal index among all connected graphs with a given number of vertices and edges [24,25]. Threshold graphs have been called *nested split graphs* by some authors [24–26].

In Section 2, we study determinants of weighted threshold graph matrices, a class of matrices containing adjacency matrices of threshold graphs. We also produce recurrence equations for the determinant of a weighted threshold graph matrix. In Section 3, we design our efficient algorithm to compute the characteristic polynomial of threshold graphs and determine its complexity. A refined complexity analysis is given in Section 4 in dependence of the number of alternations. We also look at its bit complexity in Section 5, and finish with open problems.

2. The determinant of a weighted threshold graph matrix

We are concerned with adjacency matrices of threshold graphs, but we consider a slightly more general class of matrices. We call them weighted threshold graph matrices. Let $M^{d_1 d_2 \ldots d_n}_{b_1 b_2 \ldots b_{n-1}}$ be the matrix with the following entries form $\mathbb{C}$.

$$\left(M^{d_1 d_2 \ldots d_n}_{b_1 b_2 \ldots b_{n-1}}\right)_{ij} = \begin{cases} b_i & \text{if } i < j \\ b_j & \text{if } j < i \\ d_i & \text{if } i = j \end{cases}$$

Thus, the weighted threshold graph matrix for $(b_1 b_2 \ldots b_{n-1}; d_1 d_2 \ldots d_n)$ looks like this.

$$M^{d_1 d_2 \ldots d_n}_{b_1 b_2 \ldots b_{n-1}} = \begin{pmatrix} d_1 & b_1 & b_1 & \ldots & b_1 & b_1 \\ b_1 & d_2 & b_2 & \ldots & b_2 & b_2 \\ b_1 & b_2 & d_3 & \ldots & b_3 & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_1 & b_2 & b_3 & \ldots & d_{n-1} & b_{n-1} \\ b_1 & b_2 & b_3 & \ldots & b_{n-1} & d_n \end{pmatrix}$$

We want to compute the determinant D_n of $M^{d_1 d_2 \ldots d_n}_{b_1 b_2 \ldots b_{n-1}}$ for $n \geq 0$. For $n = 0$, it is convenient to define the determinant of the 0×0 matrix to be 1.

Theorem 1. $D_n = \det\left(M^{d_1 d_2 \ldots d_n}_{b_1 b_2 \ldots b_{n-1}}\right)$ *is determined by the recurrence equation*

$$D_n = \begin{cases} 1 & \text{if } n = 0 \\ d_1 & \text{if } n = 1 \\ (d_n + d_{n-1} - 2b_{n-1})D_{n-1} - (d_{n-1} - b_{n-1})^2 D_{n-2} & \text{if } n \geq 2 \end{cases}$$

Proof. The claim is clearly correct for $n \leq 1$. In order to compute the determinant of $M^{d_1 d_2 \ldots d_n}_{b_1 b_2 \ldots b_{n-1}}$ for $n \geq 2$, we subtract the penultimate row from the last row and the penultimate column from the last column. In other words, we do a similarity transformation with the following regular matrix