Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 190 (2007) 1487–1491
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

An algorithm for calculating the independence and vertex-cover polynomials of a graph

Gordon G. Cash

New Chemicals Screening and Assessment Branch, Risk Assessment Division (7403M), US Environmental Protection Agency, 1200 Pennsylvania Avenue, N. W., Washington, DC 20460, United States

Abstract

The independence polynomial, $\omega(G,x) = \sum w_k x^k$, of a graph, G, has coefficients, w_k, that enumerate the ways of selecting k vertices from G so that no two selected vertices share an edge. The independence number of G is the largest value of k for which $w_k \neq 0$. Little is known of less straightforward relationships between graph structure and the properties of $\omega(G,x)$, in part because of the difficulty of calculating values of w_k for specific graphs. This study presents a new algorithm for these calculations which is both faster than existing ones and easily adaptable to high-level computer languages.

Keywords: Independence polynomial; Graph polynomial; Independence number; Vertex-cover polynomial

1. Introduction

The independence polynomial of a graph, G, enumerates the ways of selecting k vertices from the graph is such a way that no two are connected by an edge. The original form of this polynomial is [1,2]

$$\omega(G,x) = \sum_{k=0}^{n} w_k x^k, \tag{1}$$

where the w_k are the number of ways of so choosing k vertices. By analogy with the much-studied matching polynomial, which enumerates independent sets of edges rather that vertices, the form

$$\omega(G,x) = \sum_{k=0}^{n} (-1)^k w_k x^{n-k} \tag{2}$$

is also encountered [3,4]. Further complicating the issue,

$$\omega(G,x) = \sum_{k=0}^{n} (-1)^k w_k x^{2n-2k} \tag{3}$$

E-mail address: cash.gordon@epa.gov

0096-3003/$ - see front matter
doi:10.1016/j.amc.2007.02.028

has been put forward as well [5]. Of course, the w_k are the same in (1)–(3), and the forms may be readily interconverted. An exactly analogous situation for the edge-matching polynomial has already been recognized, since (2) corresponds to the usual form of this polynomial and (1) corresponds to the Hosoya Z-counting polynomial [6]; formulas for interconverting these two polynomials have been published [7]. The present study uses (1) for the independence polynomial.

There is almost nothing in the published literature regarding practical applications of the independence polynomial, probably because of the difficulty of computing its coefficients. Indeed, Makowsky and Mariño [8] show this computation is NP-hard. Their paper evaluates the computational complexity of this and many other graph polynomials, but they provide no algorithms for actually performing computations. Fajtlowicz and Larson [9] recently suggested that the independence number, which is the largest value of k for which $w_k > 0$, of a fullerene graph is a predictor of fullerene stability. This paper, however, mentions the well-known difficulty of calculating even this one coefficient.

There are two algorithms available for calculating $\omega(G,x)$ for a specific graph, one in the published literature [10], and the other available on the internet [11]. As the authors in [10] noted, this method is practically limited to graphs of eight or fewer vertices. Advances in computer hardware and software have doubtless increased this number somewhat, but they have not overcome the inherent limitations of the method. The internet site [11] includes a downloadable Mathematica package that contains a function IndependencePolynomial[g, x]. Besides requiring Mathematica to run, this function operates on a Mathematica data object of type Graph_, and is not readily portable to compiler languages such as Fortran, C, or Pascal. As the documentation on the site points out, Mathematica is slower at computing polynomial coefficients than compiled code. Furthermore, that method is slower than the new algorithm presented here, even though both run in Mathematica.

2. The vertex-cover polynomial

The vertex-cover polynomial of an undirected graph, $\Psi(G,x)$, was discussed recently by Dong et al. [12], and nothing about it seems to have appeared since then, except for one conference presentation [13]. As with the independence polynomial, the lack of attention to practical applications may be due to the difficulty of calculating the coefficients. Dong et al. define $\Psi(G,x)$ as follows: Let $V(G)$ and $E(G)$ be the vertex set and edge set of an undirected graph without loops or multiple edges. Then, $V' \subseteq V$ is a k-vertex cover in G iff $|V'| = k$ and $V' \cap \{x,y\} \neq \emptyset$ for all $x,y \in E(G)$. In other words, a k-vertex cover is a set of vertices selected such that every edge terminates in at least one vertex from the set. The vertex-cover polynomial then simply enumerates the k-covers of various cardinalities:

$$\Psi(G,x) = \sum_{k=0}^{|V(G)|} p_k x^k.$$

It has long been known that $p_k = w_{n-k}$ [14], although Dong et al. [12] do not mention the complementarity of these sets. This is curious, since their paper gives formulas equivalent to (4) and (5) below. They also provide formulas for the vertex polynomials of paths and cycles, using binomial coefficients instead of Chebyshev polynomials.

3. The algorithm

The algorithm described here relies on four formulas presented in [2,10]. First, where G and H are disjoint graphs

$$\omega(G \cup H, x) = \omega(G,x) \times \omega(H,x). \tag{4}$$

Second, where v is an arbitrary vertex in G, N_v is v plus all the vertices directly joined to v by edges, and $G \setminus s$ is G with vertex set s and all edges incident to s deleted

$$\omega(G,x) = \omega(G \setminus v, x) + x\omega(G \setminus N_v, x). \tag{5}$$