Applied Mathematics and Computation 332 (2018) 42–47

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Computing the numbers of independent sets and matchings of all sizes for graphs with bounded treewidth☆

Pengfei Wan[a], Jianhua Tu[b], Shenggui Zhang[a,*], Binlong Li[a]

[a] Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China
[b] Department of Mathematics, Beijing University of Chemical Technology, Beijing 100029, PR China

ARTICLE INFO

MSC:
05C69
05C80

Keywords:
Independent set
Matching
Treewidth
Dynamic programming

ABSTRACT

In the theory and applications of graphs, it is a basic problem to compute the numbers of independent sets and matchings of given sizes. Since the problem of computing the total number of independent sets and that of matchings of graphs is #P-complete, it is unlikely to give efficient algorithms to find the numbers of independent sets and matchings of given sizes. In this paper, for graphs with order n and treewidth at most p, we present two dynamic algorithms to compute the numbers of independent sets of all sizes with runtime $O(2^p \cdot pn^3)$ and the numbers of matchings of all sizes with runtime $O(2^{2p} \cdot pn^3)$, respectively. By the algorithms presented in this paper, for graphs with small treewidths, the numbers of independent sets and matchings of all possible sizes can be computed efficiently.

1. Introduction

Let G be a graph with n vertices. An *independent set* of G is a set I of vertices of G such that no two vertices in I are adjacent. For $0 \le k \le n$, the number of independent sets with k vertices is denoted by $i_k(G)$. We regard the empty set $\emptyset$ as an independent set of size 0 and set $i_0(G) = 1$. The total number of independent sets, denoted by $i(G)$, is $i(G) = \sum_{k=0}^{n} i_k(G)$. In chemical literature $i(G)$ is referred to as the *Merrifield–Simmons index* [23]. It is also called the *Fibonacci number* [28] of G in some literature. This graph invariant has close relations to the combinatorial group theory [2] and statistical mechanics [18]. The Merrifield–Simmons index is one of the most popular topological indices in chemistry, which was investigated by many chemists and mathematicians. For more details, we refer to Refs. [14,24,36].

Let G be a graph with m edges. A *matching* of G is a set M of edges in G such that no two edges in M share a common vertex. For $0 \le k \le m$, the number of matchings with k edges is denoted by $i'_k(G)$. We set $i'_0(G) = 1$ as the case of independent sets. The total number of matchings, denoted by $i'(G)$, is $i'(G) = \sum_{k=0}^{m} i'_k(G)$. In the chemical literature the parameter $i'(G)$ is referred to as the *Hosoya index* which was introduced by Hosoya in [15]. In a series of subsequent papers, it was shown that the Hosoya index is connected with a variety of physico-chemical properties of alkanes, for example, boiling point, entropy, heat of vaporization. The applicability of the Hosoya index in the theory of conjugated π-electron systems was also revealed [16]. For more details, we refer to Refs. [14,24,36].

☆ Supported by NSFC (Nos. 11571135, 11671320 and 11601429).
* Corresponding author.
E-mail addresses: pfwan@mail.nwpu.edu.cn (P. Wan), tujh81@163.com (J. Tu), sgzhang@nwpu.edu.cn (S. Zhang), binlongli@nwpu.edu.cn (B. Li).

https://doi.org/10.1016/j.amc.2018.03.017
0096-3003/

Studies on the information content of graphs and complex networks have been initiated in the late 1950s [25,30,35], following the seminal work [32] of Shannon. Later, various graph entropies were developed [7,11,20]. Distinct graph entropies have been used successfully in various disciplines, e.g., in pattern recognition, biology, chemistry, and computer science [5,6,11,12,26]. For more details, we refer to Refs. [13,33,34]. Recently, Cao et al. [9] introduced two graph entropies for a graph G, i.e., the entropy based on independent sets, denoted by $I_v(G)$ and the entropy based on matchings, denoted by $I_e(G)$, as follows:

$$I_v(G) = -\sum_{k=0}^{n} \frac{i_k(G)}{i(G)} \log \frac{i_k(G)}{i(G)}, \tag{1.1}$$

$$I_e(G) = -\sum_{k=0}^{m} \frac{i'_k(G)}{i'(G)} \log \frac{i'_k(G)}{i'(G)}. \tag{1.2}$$

The problem of computing the numbers of independent sets and that of matchings of all sizes naturally arises when we study the Merrifield–Simmons index, the Hosoya index, the entropy based on independent sets and the entropy based on matchings, etc. Provan and Ball [29] verified that the problem of computing the total number of independent sets is #P-complete for general graphs and even for bipartite graphs. Jerrum [17] showed that even for planar graphs, computing the total number of matchings is #P-complete. There are no efficient algorithms to compute the numbers of independent sets and matchings of all sizes for general graphs till now.

However, the problem of computing the total number (or the number of given size) of independent sets and matchings can be solved efficiently for some graph classes. Okamoto et al. [27] provided a linear-time algorithm to compute the total number of independent sets for chordal graphs. Lin and Chen provided two algorithms to compute the total number of independent sets for some subclasses of the class of bipartite graphs [21,22]. Zhang et al. [37] presented a linear-time algorithm to compute the total number of matchings for trees. Ahmadi and Daskhezr [1] calculated the numbers of independent sets of all sizes of some graphs by using their adjacency matrices.

In this paper, we present two dynamic programming algorithms to compute the numbers of independent sets and the numbers of matchings of all sizes for graphs with bounded treewidth. The treewidth is a graph parameter which captures how similar a graph to a tree roughly. This concept was first introduced by Robertson and Seymour [31] in their fundamental work on graph minors. In this paper, given an n-vertex graph together with a nice tree decomposition of width at most p, we describe two dynamic programming algorithms to compute the numbers of independent sets of all sizes and the numbers of matchings of all sizes, with the runtime $O(2^p \cdot pn^3)$ and $O(2^{2p} \cdot pn^3)$, respectively. Many graph families, such as cactus graphs, pseudoforests, series-parallel graphs, outerplanar graphs, Halin graphs, Apollonian networks and so on, have bounded treewidth. By using the algorithms presented in this paper, for these graphs, the numbers of independent sets and matchings can be computed in polynomial time.

The rest of this paper is organized as follows. In Section 2, we give some notations and introduce the concepts of tree decomposition and treewidth formally. In Sections 3 and 4, we present two dynamic programming algorithms to compute the numbers of independent sets and the numbers of matchings of all sizes for graphs with bounded treewidth, respectively.

2. Preliminaries

In this paper, we only consider graphs without loops and multiple edges. For terminology and notations not defined here, we refer the reader to Bondy and Murty [8]. Let G be a graph. For a vertex $v \in V(G)$, denote by $N_G(v)$ the set of neighbors of v in G. A subgraph G' of G is an *induced subgraph* of G (or a *subgraph induced by* $V(G')$) if G' contains all edges $uv \in E(G)$ with $u, v \in V(G')$. For a subset V' of $V(G)$, we use $G[V']$ to denote the subgraph induced by V'. We set $N_{V'}(v) = N_G(v) \cap V'$.

Definition 1. A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \{X_t : t \in V(T)\})$, where T is a tree, $X_t \subseteq V(G)$ for each $t \in V(T)$, and the following three conditions hold:

- $\bigcup_{t \in V(T)} X_t = V(G)$,
- for all edges $uv \in E(G)$, there exists $t \in V(T)$ with $u \in X_t$ and $v \in X_t$,
- for all $r, s, t \in V(T)$: if s is on the path from r to t in T, then $X_r \cap X_t \subseteq X_s$.

We call a vertex of T a *node* of T and X_t the *bag* of the node t. The *width* of a tree decomposition $\mathcal{T}$ is defined as $\max_{t \in V(T)} |X_t| - 1$. The *treewidth* of a graph G, denoted by $tw(G)$, is defined as the minimum width over all tree decompositions of G.

Although computing the exact treewidth of a graph G is NP-hard [3], for a fixed constant k, the problem of determining whether $tw(G) \leq k$ can be solved in polynomial time and a corresponding tree decomposition can be constructed in linear time [4].

A commonly used tree decomposition is the nice tree decomposition which was introduced in [19]. It considerably eases the design of algorithms, and one can also expect in several cases to have better constant factors in the running times of algorithms that use nice instead of normal tree decompositions.