Journal of Computer and System Sciences 78 (2012) 698–706

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Faster algorithms for finding and counting subgraphs

Fedor V. Fomin [a,*,1], Daniel Lokshtanov [a], Venkatesh Raman [b], Saket Saurabh [b], B.V. Raghavendra Rao [c]

[a] Department of Informatics, University of Bergen, Bergen, Norway
[b] The Institute of Mathematical Sciences, Chennai, India
[c] Department of Computer Science, Universitaet des Saarlandes, Germany

ARTICLE INFO

Article history:
Received 23 July 2010
Received in revised form 26 September 2011
Accepted 10 October 2011
Available online 20 October 2011

Keywords:
Parameterized complexity
Subgraph Isomorphism
Homomorphism
Counting
Treewidth

ABSTRACT

In the Subgraph Isomorphism problem we are given two graphs F and G on k and n vertices respectively as an input, and the question is whether there exists a subgraph of G isomorphic to F. We show that if the treewidth of F is at most t, then there is a randomized algorithm for the Subgraph Isomorphism problem running in time $\mathcal{O}^*(2^k n^{2t})$. Our proof is based on a novel construction of an arithmetic circuit of size at most $n^{\mathcal{O}(t)}$ for a new multivariate polynomial, Homomorphism Polynomial, of degree at most k, which in turn is used to solve the Subgraph Isomorphism problem. For the counting version of the Subgraph Isomorphism problem, where the objective is to count the number of distinct subgraphs of G that are isomorphic to F, we give a deterministic algorithm running in time and space $\mathcal{O}^*(\binom{n}{k/2} n^{2p})$ or $\binom{n}{k/2} n^{\mathcal{O}(t \log k)}$. We also give an algorithm running in time $\mathcal{O}^*(2^k \binom{n}{k/2} n^{5p})$ and taking $\mathcal{O}^*(n^p)$ space. Here p and t denote the pathwidth and the treewidth of F, respectively. Our work improves on the previous results on Subgraph Isomorphism, it also extends and unifies most of the known results on sub-path and sub-tree isomorphisms.

1. Introduction

In this paper we consider the classical problem of finding and counting a fixed pattern graph F on k vertices in an n-vertex host graph G, when we restrict the treewidth of the pattern graph F by t. More precisely the problems we consider are the Subgraph Isomorphism problem and the #Subgraph Isomorphism problem. In the Subgraph Isomorphism problem we are given two graphs F and G on k and n vertices respectively as an input, and the question is whether there exists a subgraph in G which is isomorphic to F? In the #Subgraph Isomorphism problem the objective is to count the number of distinct subgraphs of G that are isomorphic to F. Recently #Subgraph Isomorphism, in particular when F has bounded treewidth, has found applications in the study of biomolecular networks. We refer to Alon et al. [1] and references therein for further details.

In a seminal paper Alon et al. [3] introduced the method of Color-Coding for the Subgraph Isomorphism problem, when the treewidth of the pattern graph is bounded by t and obtained randomized as well as deterministic algorithms running in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(t)}$. This algorithm was derandomized using k-perfect hash families. In particular, Alon et al. [3] gave a randomized $\mathcal{O}^*(5.4^k)$[2] time algorithm and a deterministic $\mathcal{O}^*(c^k)$ time algorithm, where c is a large constant, for

* Corresponding author.
E-mail addresses: fomin@ii.uib.no (F.V. Fomin), daniello@ii.uib.no (D. Lokshtanov), vraman@imsc.res.in (V. Raman), saket@imsc.res.in (S. Saurabh), bvrr@cs.uni-sb.de (B.V.R. Rao).

1 Supported by the European Research Council advanced grant.

2 We use $\mathcal{O}^*()$ notation that hides factors polynomial in n and the parameter k.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2011.10.001

the k-Path problem, a special case of Subgraph Isomorphism where F is a path of length k. There have been a lot of efforts in parameterized algorithms to reduce the base of the exponent of both deterministic as well as the randomized algorithms for the k-Path problem. In the first of such attempts, Chen et al. [10] and Kneis et al. [17] independently discovered the method of Divide and Color and gave a randomized algorithm for k-Path running in time $\mathcal{O}^*(4^k)$. Chen et al. [10] also gave a deterministic algorithm running in time $\mathcal{O}^*(4^{k+o(k)})$ using an application of universal sets. While the best known deterministic algorithm for k-Path problem still runs in time $\mathcal{O}^*(4^{k+o(k)})$, the base of the exponent of the randomized algorithm for the k-Path problem has seen a drastic improvement. Koutis [18] used an algebraic approach based on group algebras for k-Path and gave a randomized algorithm running in time $\mathcal{O}^*(2^{3k/2}) = \mathcal{O}^*(2.83^k)$. Williams [21] augmented this approach with more random choices and several other ideas and gave an algorithm for k-Path running in time $\mathcal{O}^*(2^k)$. Currently the fastest randomized algorithm for the problem is due to Björklund et al. [6], which runs in time $\mathcal{O}^*(1.66^k)$.

While there has been a lot of work on the k-Path problem, there has been almost no progress on other cases of the Subgraph Isomorphism problem until last year. Cohen et al. gave a randomized algorithm that for an input digraph D decides in time $\mathcal{O}^*(5.704^k)$ if D contains a given out-tree with k vertices [11]. They also showed how to derandomize the algorithm in time $\mathcal{O}^*(6.14^k)$. Amini et al. [4] introduced an inclusion–exclusion based approach in the classical Color-Coding and using it gave a randomized $5.4^k n^{\mathcal{O}(t)}$ time algorithm and a deterministic $5.4^{k+o(k)} n^{\mathcal{O}(t)}$ time algorithm for the Subgraph Isomorphism problem, when F has treewidth at most t. Koutis and Williams [19] generalized their algebraic approach for k-Path to k-Tree, a special case of Subgraph Isomorphism problem where F is a tree on k-vertices, and obtained a randomized algorithm running in time $\mathcal{O}^*(2^k)$ for k-Tree. In this work we generalize the results of Koutis and Williams by extending the algebraic approach to much more general classes of graphs, namely, graphs of bounded treewidth. More precisely, we give a randomized algorithm for the Subgraph Isomorphism problem running in time $\mathcal{O}^*(2^k(nt)^t)$, when the treewidth of F is at most t. The road map suggested by Koutis and Williams [19] and Williams [21] is to reduce the problem to checking a multilinear term in a specific polynomial of degree at most k. However, the construction of such polynomial is non-trivial and requires new ideas. Our first contribution is the introduction of a new polynomial of degree at most k, namely Homomorphism Polynomial, using a relation between graph homomorphisms and injective graph homomorphisms for testing whether a graph contains a subgraph which is isomorphic to a fixed graph F. We show that if the treewidth of the pattern graph F is at most t, then it is possible to construct an arithmetic circuit of size $\mathcal{O}^*((nt)^t)$ for Homomorphism Polynomial which combined with a result of Williams [21] yields our first theorem.

In the second part of the paper we consider the problem of counting the number of pattern subgraphs, that is, the #Subgraph Isomorphism problem. A natural question here is whether we can solve the #Subgraph Isomorphism problem in $\mathcal{O}^*(c^k)$ time, when the k-vertex graph F is of bounded treewidth or whether we can even solve the #k-Path problem in $\mathcal{O}^*(c^k)$ time? Flum and Grohe [13] showed that the #k-Path problem is #W[1]-hard and hence it is very unlikely that the #k-Path problem can be solved in time $f(k)n^{\mathcal{O}(1)}$ where f is any arbitrary function of k. In another negative result, Alon and Gutner [2] have shown that one cannot hope to solve #k-Path better than $\mathcal{O}(n^{k/2})$ using the method of Color-Coding. They show this by proving that any family $\mathcal{F}$ of "balanced hash functions" from $\{1, \ldots, n\}$ to $\{1, \ldots, k\}$, must have size $\Omega(n^{k/2})$. On the positive side, very recently Vassilevska and Williams [20] studied various counting problems and among various other results gave an algorithm for the #k-Path problem running in time $\mathcal{O}^*(2^k (k/2)! \binom{n}{k/2})$ and space polynomial in n. Björklund et al. [5] introduced the method of "meet-in-the-middle" and gave an algorithm for the #k-Path problem running in time and space $\mathcal{O}^*(\binom{n}{k/2})$. They also gave an algorithm for #k-Path problem running in time $\mathcal{O}^*(3^{k/2}\binom{n}{k/2})$ and polynomial space, improving on the polynomial space algorithm given in [20]. We extend these results to the #Subgraph Isomorphism problem, when the pattern graph F is of bounded treewidth or pathwidth. And here also graph homomorphisms come into play! By making use of graph homomorphisms we succeed to extend the applicability of the meet-in-the-middle method to much more general structures than paths. Combined with other tools—inclusion–exclusion, the Disjoint Sum problem, separation property of graph of bounded treewidth or pathwidth and the trimmed variant of Yate's algorithm presented in [7]—we obtain the following results. Let F be a k-vertex graph and G be an n-vertex graph of pathwidth p and treewidth t. Then #Subgraph Isomorphism is solvable in time $\mathcal{O}^*(\binom{n}{k/2} n^{2p})$ and $\binom{n}{k/2} n^{\mathcal{O}(t \log k)}$ and space $\mathcal{O}^*(\binom{n}{k/2})$. We also give an algorithm for #Subgraph Isomorphism that runs in time $\mathcal{O}^*(2^k \binom{n}{k/2} n^{3p} t^{2t})$ (respectively $2^k \binom{n}{k/2} n^{\mathcal{O}(t \log k)}$) and takes $\mathcal{O}^*(n^p)$ space (respectively $\mathcal{O}^*(n^t)$ space). Thus our work not only improves on known results on Subgraph Isomorphism of Alon et al. [3] and Amini et al. [4] but it also extends and generalize most of the known results on k-Path and k-Tree of Björklund et al. [5], Koutis and Williams [19] and Williams [21].

The main theme of both algorithms, for finding and for counting a fixed pattern graph F, is to use graph homomorphisms as the main tool. Counting homomorphisms between graphs has found applications in variety of areas, including extremal graph theory, properties of graph products, partition functions in statistical physics and property testing of large graphs. We refer to the excellent survey of Borgs et al. [8] for more references on counting homomorphisms. One of the main advantages of using graph homomorphisms is that in spite of their expressive power, graph homomorphisms between many structures can be counted efficiently. Secondly, it allows us to generalize various algorithm for counting subgraphs with an ease. We combine counting homomorphisms with the recent advancements on computing different transformations efficiently on subset lattice.