Journal of Computer and System Sciences 78 (2012) 1606–1622

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Catalan structures and dynamic programming in H-minor-free graphs ☆

Frederic Dorn [b,*], Fedor V. Fomin [a], Dimitrios M. Thilikos [c]

[a] Department of Informatics, University of Bergen, Bergen, Norway
[b] SINTEF Energy Research, Trondheim, Norway
[c] Department of Mathematics, National & Kapodistrian University of Athens, Greece

ARTICLE INFO

Article history:
Received 5 November 2009
Received in revised form 19 January 2012
Accepted 24 February 2012
Available online 1 March 2012

Keywords:
Parameterized complexity
Longest path
Minor-free graphs
Catalan structure

ABSTRACT

We give an algorithm that, for a fixed graph H and integer k, decides whether an n-vertex H-minor-free graph G contains a path of length k in $2^{O(\sqrt{k})} \cdot n^{O(1)}$ steps. Our approach builds on a combination of Demaine–Hajiaghayi's bounds on the size of an excluded grid in such graphs with a novel combinatorial result on certain branch decompositions of H-minor-free graphs. This result is used to bound the number of ways vertex disjoint paths can be routed through the separators of such decompositions. The proof is based on several structural theorems from the Graph Minors series of Robertson and Seymour. With a slight modification, similar combinatorial and algorithmic results can be derived for many other problems. Our approach can be viewed as a general framework for obtaining time $2^{O(\sqrt{k})} \cdot n^{O(1)}$ algorithms on H-minor-free graph classes.

1. Introduction

One of the motivations of this paper was the seminal result of Alon, Yuster, and Zwick in [3] that proved that a path of length $\log n$ can be found in polynomial time, answering to a question by Papadimitriou and Yannakakis in [32]. One of the open questions left in [3] was: "*Is there a polynomial time* (*deterministic or randomized*) *algorithm for deciding if a given graph G contains a path of length, say,* $\log^2 n$?". Of course, a $2^{O(\sqrt{k})} \cdot n^{O(1)}$ step algorithm for checking if a graph contains a path of length k would resolve this question. However, an algorithm of running time $2^{o(k)} \cdot n^{O(1)}$ for this problem, even for sparse graphs, would contradict the widely believed exponential time hypothesis, i.e. would imply that 3-SAT can be solved in subexponential time [25]. In this paper, we devise a $2^{O(\sqrt{k})} \cdot n^{O(1)}$ step algorithm for this problem on H-minor-free graphs, implying a polynomial-time algorithm for a $\log^2 n$-length path. This result is tight, because, according to Deĭneko, Klinz, and Woeginger [8], the existence of a $2^{o(\sqrt{k})} \cdot n^{O(1)}$ step algorithm, even for planar graphs, would again violate the exponential time hypothesis.

Our work is also motivated by the paradigm of parameterized algorithms [21,22,31]. A common technique in parameterized algorithms for problems asking for the existence of vertex/edge subsets of size k with certain properties is based on branchwidth (treewidth) and involves the following two ingredients: The first is a combinatorial proof that if the branchwidth of the input graph is at least $f(k)$ (where f is some function of k) then the answer to the problem is directly implied. The second is a $g(\mathbf{bw}(G)) \cdot n^{O(1)}$ step dynamic programming algorithm for the problem (here $\mathbf{bw}(G)$ is the branchwidth of the input graph G). For obtaining a $2^{O(\sqrt{k})} \cdot n^{O(1)}$ step algorithm out of this, we further require that

☆ An extended abstract of this paper appeared in SODA'08 (Dorn et al., 2008 [18]). The research of the last author has been supported by the project "Kapodistrias" (AΠ 736/24.3.2006 and 02839/28.07.2008) of the National and Kapodistrian University of Athens (project code: 70/4/8757).

* Corresponding author.
E-mail addresses: Frederic.Dorn@sintef.no (F. Dorn), fomin@ii.uib.no (F.V. Fomin), sedthilk@math.uoa.gr (D.M. Thilikos).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2012.02.004

(a) $f(k) = O(\sqrt{k})$, and
(b) $g(k) = 2^{O(\mathbf{bw}(G))}$.

For planar graphs (and also for H-minor-free graphs or apex-minor-free graphs – see [13] and [9]) **(a)** can be proved systematically using the idea of Bidimensionality [12,19]. However, not an equally general theory exists for **(b)**. On the positive side, **(b)** holds for several combinatorial problems. Typical problems in NP that fall in this category are Vertex Cover, Dominating Set or Edge Dominating Set, where no global conditions are imposed on their certificates [1,2,10,23]. This implies that the existence of such a set of size $\log^2 n$ can be decided in polynomial time and this answers positively the analogue of the question in [3] for these problems on H-minor-free graphs. The bad news is that, for many combinatorial problems, a general algorithm for proving **(b)** is missing. Longest path is a typical example of such a problem. Here the certificate of a solution should satisfy a global connectivity requirement. For this reason, the dynamic programming algorithm must keep track of all the ways the required path may traverse the corresponding separator of the decomposition, that is $\Omega(\ell^\ell)$ on the size ℓ of the separator and therefore of treewidth/branchwidth. The same problem in designing dynamic programming algorithms appears for many other combinatorial problems in NP whose solution certificates are restricted by global properties such as connectivity. Other examples of such problems are Longest Cycle, Connected Dominating Set, Feedback Vertex Set, Hamiltonian Cycle and Graph Metric Traveling Salesman Problem (TSP).

Recently, [20] overcame the above deadlock for the class of planar graphs. Later, a similar result was given in [17] for graphs of bounded genus. The proofs in [20,17] are heavily based on arguments about non-crossing paths in graphs embedded in topological surfaces. This makes it possible to construct special types of graph decompositions of the input graph where the number of ways a path (or a cycle) traverses a separator of the decomposition is linearly bounded by the Catalan number of the separator size (which yields the desired single exponential dependance on treewidth or branchwidth). We refer to Stanley's book [40] for more information and different applications of Catalan numbers. It is not clear, a priori, whether this type of arguments can be extended to graphs excluding a minor. Another example of a technique, where the extension from planar and bounded genus to H-minor-free graphs is not clear, is given in [27,14].

In this paper, we provide a general framework for the design of dynamic programming algorithms on H-minor-free graphs. For this, it is necessary to go through the entire characterization of H-minor-free graphs given by Robertson and Seymour in their Graph Minors project (in particular, in [37]) to prove counting lemmata that can suitably bound the amount of information required in each step of the dynamic programming algorithm.

The main combinatorial result of this paper is Theorem 2, concerning the existence of suitably structured branch decompositions of H-minor-free graphs. While the grid excluding part follows directly from [13], the construction of the branch decomposition of Theorem 2 is quite involved. Indeed it uses the fact, proven by Robertson and Seymour in [37], that any H-minor-free graph can roughly be obtained by identifying in a tree-like way small cliques of a collection of components that are almost embeddable on bounded genus surfaces. The main proof idea is based on a procedure of "almost"-planarizing the components of this collection. However, we require a planarizing set with certain topological properties, able to reduce the high genus embeddings to planar ones where the planarizing vertices are cyclically arranged in the plane. This makes it possible to use a special type of planar branch decomposition, invented in [38], that permits to view collections of paths that may pass through a separator as non-crossing pairings of the vertices of a cycle. This provides the so-called *Catalan structure* of the decomposition and permits us to suitably bound the ways a path may cross its separators. Let us remark that similar ideas were also used in parameterized and approximation algorithms on planar graphs [8,11,27]. This decomposition is used to build a decomposition on the initial almost embeddable graph. Then using the tree-like way these components are linked together, we build a branch decomposition of the entire graph. The most technical part of the proof is to show that each step of this construction, from the almost planar case to the entire graph, maintains the Catalan structure, yielding the claimed upper bound.

Almost immediately, Theorem 2 implies the main algorithmic result of this paper. If a graph G on n vertices contains a $(\sqrt{k} \times \sqrt{k})$-grid, then G has a path of length k. Otherwise, by Theorem 2, it has a branch decomposition of width $O(\sqrt{k})$ with the Catalan structure. By standard dynamic programming on this branch decomposition (e.g., see [6]) we find the longest path in G. We stress that the dynamic programming algorithm is not different from the standard one. It is the special branch decomposition of Theorem 2 that accelerates its running time because the number of states at each step of the dynamic programming is bounded by $2^{O(\sqrt{k})}$. Thus the running time of the algorithm is $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

Organization of the paper. First, we give some preliminaries in Section 2, where we restate the structural theorem on H-minor-free graphs for our purposes. In Section 3, we state our main theorem and give the algorithm for computing branch decompositions with Catalan structure, which we employ in Section 5 for solving the problem of finding a path of length k in H-minor-free graphs. In Section 4 we provide technical details of the correctness proof of the algorithm. We end (Section 6) with concluding remarks and open problems.

2. Preliminaries

Surface embeddable graphs. We refer to the book of Diestel [16] for the basic Graph Theory terminology and to the book of Mohar and Thomassen [29] on basics of Topological Graph Theory. We use the notation $V(G)$ and $E(G)$, for the set of the vertices and edges of G. A *surface* Σ is a compact 2-manifold without boundary (we always consider connected surfaces).