Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Minimum size tree-decompositions[☆,☆☆]

Bi Li[c], Fatima Zahra Moataz[b,a], Nicolas Nisse[a,b,*], Karol Suchan[d,e]

[a] Inria, France
[b] Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, Sophia Antipolis, France
[c] School of Mathematics and Statistics, Xidian University, Xi'an, China
[d] Universidad Adolfo Ibáñez, Santiago, Chile
[e] AGH University of Science and Technology, Krakow, Poland

ARTICLE INFO

Article history:
Received 29 November 2015
Received in revised form 28 October 2016
Accepted 30 January 2017
Available online xxxx

Keywords:
Tree-decomposition
Treewidth
NP-hard

ABSTRACT

We study in this paper the problem of computing a tree-decomposition of a graph with width at most k and minimum number of bags. More precisely, we focus on the following problem: given a fixed $k \geq 1$, what is the complexity of computing a tree-decomposition of width at most k with minimum number of bags in the class of graphs with treewidth at most k? We prove that the problem is NP-complete for any fixed $k \geq 4$ and polynomial for $k \leq 2$; for $k = 3$, we show that it is polynomial in the class of trees and 2-connected outerplanar graphs.

1. Introduction

A *tree-decomposition* of a graph [15] G is a way to represent G by a family of subsets of its vertex-set organized in a tree-like manner and satisfying some connectivity property. The *treewidth* of G measures the proximity of G to a tree. More formally, a tree-decomposition of $G = (V, E)$ is a pair $(T, \mathcal{X})$ where $\mathcal{X} = \{X_t | t \in V(T)\}$ is a family of subsets of V, called *bags*, and T is a tree, such that:

- $\bigcup_{t \in V(T)} X_t = V$;
- for any edge $uv \in E$, there is a bag X_t (for some node $t \in V(T)$) containing both u and v;
- for any vertex $v \in V$, the set $\{t \in V(T) | v \in X_t\}$ induces a subtree of T.

The *width* of a tree-decomposition $(T, \mathcal{X})$ is $max_{t \in V(T)} |X_t| - 1$ and its *size* is the order $|V(T)|$ of T. The treewidth of G, denoted by $tw(G)$, is the minimum width over all possible tree-decompositions of G. If T is constrained to be a path, $(T, \mathcal{X})$ is called a *path-decomposition* of G. The pathwidth of G, denoted by $pw(G)$, is the minimum width over all possible path-decompositions of G.

Tree-decompositions are the corner-stone of many dynamic programming algorithms for solving graph problems. For example, the famous Courcelle's Theorem states that any problem expressible in MSOL can be solved in linear-time in the class of bounded treewidth graphs [7]. Another framework based on graph decompositions is the *bi-dimensionality theory* that allowed the design of sub-exponential-time algorithms for many problems in the class of graphs excluding some fixed

☆ This work has been partially supported by European Project FP7 EULER, ANR project Stint (ANR-13-BS02-0007), the associated Inria team AlDyNet, the project ECOS-Sud Chile (C12E03) and a grant from the "Conseil régional Provence Alpes-Côte d' Azur".

☆☆ An extended abstract of this paper has been published in the proceedings of the 8th Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS 2015) (Li et al., 2015).

* Corresponding author at: Inria, France.
E-mail address: nicolas.nisse@inria.fr (N. Nisse).

http://dx.doi.org/10.1016/j.dam.2017.01.030
0166-218X/

graph as a minor (e.g., [8]). Given a tree-decomposition with width w and size n, the time-complexity of most of such dynamic programming algorithms can often be expressed as $O(2^{w}n)$ or $O(2^{w\log w}n)$. These algorithms have mainly theoretical interest because their time-complexity depends exponentially on the treewidth and, on the other hand, no practical algorithms are known to compute a good tree-decomposition for graphs with treewidth at least 5.

Since the computation of tree-decompositions is a challenging problem, we propose in this article to study it from a new point of view. Namely, we aim at minimizing the number of bags of the tree-decomposition when the width is bounded. This new perspective is interesting on its own and we hope it will allow to gain more insight into the difficulty of designing practical algorithms for computing tree-decompositions.

We consider the problem of computing tree-decompositions with minimum size. If the width is not constrained, then a trivial solution is a tree-decomposition of the graph with one bag (the full vertex-set). Hence, given a graph G and an integer $k \geq tw(G)$, we consider the problem of minimizing the size of a tree-decomposition of G with width at most k.

Related work. The problem of computing "good" tree-decompositions has been extensively studied. Computing optimal tree-decomposition – i.e., with width $tw(G)$ – is NP-complete in the class of general graphs G [1]. For any fixed $k \geq 1$, Bodlaender designed an algorithm that computes, in time $O(k^{k^3}n)$, a tree-decomposition of width k of any n-vertex graph with treewidth at most k [3]. Recently, a single-exponential (in k) algorithm has been proposed that computes a tree-decomposition with width at most $5k$ in the class of graphs with treewidth at most k [4]. As far as we know, the only practical algorithms for computing optimal tree-decompositions hold for graphs with treewidth at most 1 (trivial since $tw(G) = 1$ if and only if G is a tree), 2 (graphs excluding K_4 as a minor) [18], 3 [2,12,14] and 4 [16].

In [9], Dereniowski et al. consider the problem of minimum size path-decompositions. Given any positive integer k and any graph G with pathwidth at most k, let $l_k(G)$ denote the smallest size (length) of a path-decomposition of G with width at most k. For any fixed $k \geq 4$, computing l_k is NP-complete in the class of general graphs and it is NP-complete, for any fixed $k \geq 5$, in the class of connected graphs [9]. Moreover, computing l_k can be solved in polynomial-time in the class of graphs with pathwidth at most k for any $k \leq 3$. Finally, the "dual" problem is also hard: for any fixed $l \geq 2$, it is NP-complete in general graphs to compute the minimum width of a path-decomposition with length l [9].[1] We have generalized the problem of minimum size path-decomposition presented in [9], and introduced the problem of minimum size tree-decomposition in a shorter version of this paper [13]. To the best of our knowledge, no other paper has dealt with the computation of tree-decompositions with minimum size before [13]. However, very recently, following the work in [13] and [9], Bodlaender et al. [6] have proposed exact subexponential time algorithms to solve the problems of minimum size tree-decomposition and minimum size path-decomposition for a fixed width k in $2^{O(n/\log(n))}$ time and showed that the two problems cannot be solved in $2^{o(n/\log(n))}$ time, assuming the Exponential Time Hypothesis.

Contribution. Let k be any positive integer and G be any graph. If $tw(G) > k$, let us set $s_k(G) = \infty$. Otherwise, let $s_k(G)$ denote the minimum size of a tree-decomposition of G with width at most k. See a simple example in Fig. 1. We first prove in Section 2 that, for any (fixed) $k \geq 4$, the problem of computing s_k is NP-hard in the class of graphs with treewidth at most k. Moreover, the computation of s_k for $k \geq 5$ is NP-hard in the class of connected graphs with treewidth at most k. Furthermore, the computation of s_4 is NP-complete in the class of planar graphs with treewidth 3. In Section 3, we present a general approach for computing s_k for any $k \geq 1$. In the rest of the article, we prove that computing s_2 can be solved in polynomial-time, and show that s_3 can be computed in polynomial-time in the class of trees and 2-connected outerplanar graphs.

2. NP-hardness in the class of bounded treewidth graphs

In this section, we prove that:

Theorem 1. *For any fixed integer $k \geq 4$ (resp., $k \geq 5$), the problem of computing s_k is NP-complete in the class of graphs (resp., of connected graphs) with treewidth at most k.*

Note that the corresponding decision problem is clearly in NP. Hence, we only need to prove it is NP-hard.

Our proof mainly follows the one of [9] for minimum size path-decompositions. Hence, we recall here the two steps of the proof in [9]. First, it is proved that, if computing l_k is NP-hard for any $k \geq 1$ in general graphs, then the computation of l_{k+1} is NP-hard in the class of connected graphs. Second, it is shown that computing l_4 is NP-hard in general graphs with pathwidth 4. In particular, this implies that computing l_5 is NP-hard in the class of connected graphs with pathwidth 5. The second step consists of a reduction from the 3-PARTITION problem [10] to the one of computing l_4. Precisely, for any instance $\mathcal{I}$ of 3-PARTITION, a graph $G_{\mathcal{I}}$ is built such that $\mathcal{I}$ is a YES instance if and only if $l_4(G_{\mathcal{I}})$ equals a defined value $\ell_{\mathcal{I}}$.

Our contribution consists first in showing that the first step of [9] directly extends to the case of tree-decompositions. That is, it directly implies that, if computing s_k is NP-hard for some $k \geq 4$ in general graphs, then so is the computation of s_{k+1} in the class of connected graphs. Our main contribution of this section is to show that, for the graphs $G_{\mathcal{I}}$ built in the reduction proposed in [9], any tree-decomposition of $G_{\mathcal{I}}$ with width at most 4 and minimum size is a path-decomposition. Hence, in this class of graphs, $l_4 = s_4$ and, for any instance $\mathcal{I}$ of 3-PARTITION, $\mathcal{I}$ is a YES instance if and only if $s_4(G_{\mathcal{I}})$ equals a defined value $\ell_{\mathcal{I}}$. We describe the details in what follows.

1 This result proved for path-decompositions can be straightforwardly extended to tree-decompositions, i.e. for any fixed $s \geq 2$, it is NP-complete in general graphs to compute the minimum width of a tree-decomposition with size s.