Minimum Size Tree-decompositions

Bi Li, Fatima Zahra Moataz, Nicolas Nisse, Karol Suchan

Minimum Size Tree-decompositions [1]

Bi Li [a,b,c,2], Fatima Zahra Moataz [b,a,3], Nicolas Nisse [a,b,4] and Karol Suchan [d,e,5]

[a] *Inria, France*

[b] *Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, Sophia Antipolis, France*

[c] *Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China*

[d] *FIC, Universidad Adolfo Ibáñez, Santiago, Chile*

[e] *WMS, AGH, University of Science and Technology, Krakow, Poland*

Abstract

Tree-decompositions are the corner-stone of many dynamic programming algorithms for solving graph problems. Since the complexity of such algorithms generally depends exponentially on the *width* (size of the *bags*) of the decomposition, much work has been devoted to compute tree-decompositions with small width. However, practical algorithms computing tree-decompositions only exist for graphs with *treewidth* less than 4. In such graphs, the time-complexity of dynamic programming algorithms is dominated by the *size* (number of bags) of the tree-decompositions. It is then interesting to minimize the size of the tree-decompositions. In this extended abstract, we consider the problem of computing a tree-decomposition of a graph with width at most k and minimum size. 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.

Keywords: Minimum size tree-decomposition, treewidth, NP-hard.

1 Introduction

A *tree-decomposition* of a graph [9] is a way to represent it by a family of subsets of its vertex-set organized in a tree-like manner and satisfying some connectivity property. More formally, a tree decomposition of a graph $G = (V, E)$ 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: (1) $\bigcup_{t \in V(T)} X_t = V$; (2) for any edge $uv \in E$, there is $t \in V(T)$ such that X_t contains both u and v; and (3) for any vertex $v \in V$, the set $\{t \in V(T) | v \in X_t\}$ induces a subtree of T.

The *width* of $(T, \mathcal{X})$ is $max_{t \in V(T)} |X_t| - 1$ and its *size* is the order $|V(T)|$ of T. If T is a path, $(T, \mathcal{X})$ is called a *path-decomposition* of G. The treewidth (resp., pathwidth) of G, denoted by $tw(G)$ (resp., $pw(G)$), is the minimum width over all tree-decompositions (resp., 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 [5]. 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 graph as a minor (e.g., [6]). 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 $O(2^{w \log w} n)$. Therefore, the problem of computing tree-decompositions with small width has drawn much attention in the last decades.

The above-mentioned 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. However, for small (≤ 4) treewidth graphs, practical algorithms exist to compute tree-decompositions with minimum width [11,2,10]. Since the computation of tree-decompositions is a challenging problem, we propose in this paper to study it from a new point of view. Namely, we aim at minimizing the number of bags of the tree-

[1] This work has been partially supported by ANR project Stint (ANR-13-BS02-0007), the associated Inria team AlDyNet, the project ECOS-Sud Chile and a grant from the "Conseil régional Provence Alpes-Côte d'Azur".
[2] Email: `bi.li@inria.fr`
[3] Email: `fatima-zahra.moataz@inria.fr`
[4] Email: `nicolas.nisse@inria.fr`
[5] Email: `karol.suchan@uai.cl`

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. It is NP-hard to compute tree-decompositions with minimum width [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-node graph with treewidth at most k [3]. Recently, a single-exponential (in k) algorithm has been proposed that computes a tree-decomposition with width $\leq 5k$ in the class of graphs of treewidth $\leq k$ [4].

We are not aware of any work dealing with the computation of tree-decompositions with minimum size. In [7], Dereniowski *et al.* study the problem of size-constrained path-decompositions. Given any $k > 0$ 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 $k \geq 4$ (resp. $k \geq 5$), computing l_k is NP-complete in general graphs (resp., in the class of connected graphs) [7]. Moreover, computing l_k can be solved in polynomial-time in the class of graphs with pathwidth at most k for any $k \leq 3$. 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 [7]. Moreover, by the same techniques, the original proof can be extended to an analogous result on treewidth.

Our results. For any $tw > 0$ and any graph G of treewidth at most tw, let $s_k(G)$ denote the minimum size of a tree-decomposition of G of width at most k, $k \geq tw$. We consider the complexity of computing s_k for $k \geq 1$. Table 1 summarizes our results as well as the remaining open questions.

	s_1	s_2	s_3	s_4	$s_k, k = \max\{tw+1, 5\}$
Graphs of treewidth at most $tw = 1$	P (trivial)	P	P	?	?
Graphs of treewidth at most $tw = 2$	-	P	?	?	?
Graphs of treewidth at most $tw = 3$	-	-	?	NP-hard	?
Graphs of treewidth at most $tw \geq 4$	-	-	-	NP-hard	NP-hard

Table 1

2 Complexity results on the computation of s_k

In this section, we present some NP-hardness results and some polynomial algorithms for the computation of s_k for $k \geq 1$. Due to lack of space, all proofs have been omitted and can be found in [8]. First we prove that:

Theorem 2.1 *For any fixed integer $k \geq 5$, the problem of computing s_k is NP-complete in the class of connected graphs with treewidth at most $k-1$; furthermore, the problem of computing s_4 is NP-complete in the class of planar graphs with treewidth at most 3.*

Our proof extends the one of [7] for size-constrained path-decompositions.

In the rest of this extended abstract, we consider the computation of s_k for $k \leq 3$. We give polynomial-time algorithms in several graph classes. Let us first present the general approach that is used in what follows.

General Approach. Let $k \geq 1$ and G be a graph with $tw(G) \leq k$. A subset $B \subseteq V(G)$ is a *k-potential-leaf* if there is a tree-decomposition $(T, \mathcal{X})$ with width at most k and size $s_k(G)$ such that B is a leaf bag of $(T, \mathcal{X})$ (i.e., the node corresponding to bag B in T is a leaf). Abusing the notations, we will identify the subset B with the subgraph it induces in G. Given a class of graphs $\mathcal{C}$ and a positive integer k, a set of graphs $\mathcal{P}$ is called a *complete set of k-potential-leaves* of $\mathcal{C}$, if for any graph $G \in \mathcal{C}$, there exists a graph $H \in \mathcal{P}$ such that G contains a k-potential leaf isomorphic to H.

The key idea of our algorithms is to identify a complete set of potential-leaves. Then we proceed recursively: we find in G a k-potential-leaf H from the complete set, put it in a bag and combine in with the minimum size tree-decomposition of $G \setminus H$. The next lemmas formalize this idea.

Given a graph $G = (V, E)$ and a subset $S \subseteq V$, we denote by G_S the graph obtained from G by adding the minimum number of edges to G such that S induces a clique in G_S.

Lemma 2.2 *Let $k \geq 1$ and $G = (V, E)$ be a graph with $tw(G) \leq k$. Let $B \subseteq V$ be a k-potential-leaf of G. Let $S \subset B$ be the set of vertices of B that have a neighbor in $V \setminus B$. Then $s_k(G) = s_k(G_S \setminus (B \setminus S)) + 1$.*

This lemma implies the following corollary:

Corollary 2.3 *Let $k \in \mathbb{N}^*$ and $\mathcal{C}$ be the class of graphs with treewidth at most k. If there is a polynomial-time algorithm that, for any graph $G \in \mathcal{C}$, computes a k-potential-leaf of G, then s_k and a minimum size tree decomposition of width at most k can be computed in polynomial-time in $\mathcal{C}$.*

From Corollary 2.3, solving the Minimum Size Tree-Decomposition problem in a class of graphs $\mathcal{C}$ reduces to the problem of finding a k-potential-leaf for any graph $G \in \mathcal{C}$. This can be done in polynomial time if we characterize a finite complete set of k-potential-leaves of $\mathcal{C}$. In the rest of this section, we present our results for $k \in \{2, 3\}$.

Minimum Size Tree-Decompositions for $k \in \{2, 3\}$. We first characterize a complete set of 2-potential-leaves in the class of non-empty graphs (containing at least one edge) of treewidth at most 2 (see Fig. 1). Then, we identify a complete set of 3-potential-leaves in the class of trees with at least four vertices and in the class of 2-connected outerplanar graphs (see Fig. 2). Note that the complete sets of potential-leaves in the figures are finite and each potential-leaf in them can be found in polynomial time. This together with Corollary 2.3 imply the following result:

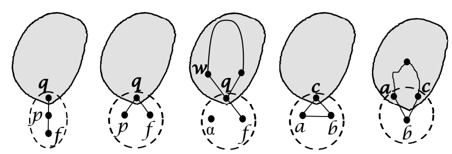
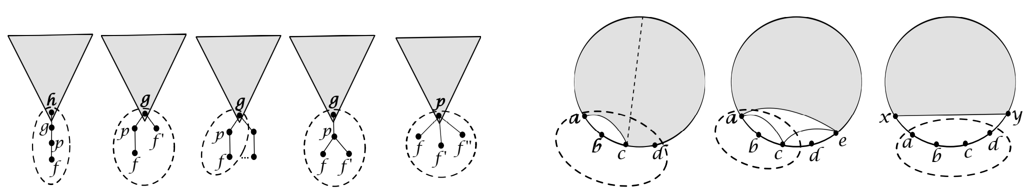


Fig. 1. Complete set of 2-potential-leaves of graphs of treewidth at most 2



(a) Trees (b) 2-connected outerplanar graphs

Fig. 2. Complete sets of 3-potential-leaves of trees and 2-connected outerplanars

Theorem 2.4 *s_2 and a minimum size tree decomposition can be computed in polynomial-time in general graphs; s_3 and a minimum size tree decomposition of width at most 3 can be computed in polynomial-time in the class of trees and 2-connected outerplanar graphs.*

3 Conclusion

In this extended abstract, we give preliminary results on the complexity of minimizing the size of tree-decompositions with given width. Table 1 summarizes our results as well as the remaining open questions. We currently investigate the computation of s_3 in the class of graphs with treewidth 2 and s_k for $k \geq 3$ in the class of trees. These cases are more intricate than the polynomial cases we have considered. It seems that a global view of the graph needs to be considered to decide whether a subgraph is a 3-potential leaf of the graph; in addition, a k-potential leaf of a tree for $k \geq 5$ can be disconnected (illustrating examples can be found in [8]).

References

[1] Arnborg, S., D. G. Corneil and A. Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Methods **8** (1987), pp. 277–284. URL http://dx.doi.org/10.1137/0608024

[2] Arnborg, S. and A. Proskurowski, *Characterization and recognition of partial 3-trees*, SIAM J. Algebraic Discrete Methods **7** (1986), pp. 305–314. URL http://dx.doi.org/10.1137/0607033

[3] Bodlaender, H. L., *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput. **25** (1996), pp. 1305–1317.

[4] Bodlaender, H. L., P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov and M. Pilipczuk, *A $o(c^k n)$ 5-approximation algorithm for treewidth*, CoRR **abs/1304.6321** (2013).

[5] Courcelle, B., *The monadic second-order logic of graphs. i. recognizable sets of finite graphs*, Information and Computation **85** (1990), pp. 12 – 75. URL http://www.sciencedirect.com/science/article/pii/089054019090043H

[6] Demaine, E. D. and M. Hajiaghayi, *The bidimensionality theory and its algorithmic applications*, Comput. J. **51** (2008), pp. 292–302.

[7] Dereniowski, D., W. Kubiak and Y. Zwols, *Minimum length path decompositions*, CoRR **abs/1302.2788** (2013).

[8] Li, B., F. Z. Moataz, N. Nisse and K. Suchan, *Size-Constrained Tree Decompositions*, Research report, INRIA Sophia-Antipolis (2014). URL https://hal.inria.fr/hal-01074177

[9] Robertson, N. and P. D. Seymour, *Graph minors. ii. algorithmic aspects of tree-width*, J. Algorithms **7** (1986), pp. 309–322.

[10] Sanders, D. P., *On linear recognition of tree-width at most four*, SIAM J. Discret. Math. **9** (1996), pp. 101–117.

[11] Wald, J. A. and C. J. Colbourn, *Steiner trees, partial 2-trees, and minimum ifi networks*, Networks **13** (1983), pp. 159–167.