Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

Hardness of computing width parameters based on branch decompositions over the vertex set

Sigve Hortemo Sæther [1,2]

Department of Informatics
University of Bergen
Bergen, Norway

Martin Vatshelle [1,3]

Department of Informatics
University of Bergen
Bergen, Norway

Abstract

Many width parameters of graphs are defined using branch decompositions over the vertex set of the graph and a corresponding cut-function. In this paper, we give a general framework for showing hardness of many width parameters defined in such a way, by reducing from the problem of deciding the exact value of the cut-function. We show that this implies NP-hardness for deciding both boolean-width and mim-width, and that mim-width is W[1]-hard, and not in APX unless NP=ZPP.

Keywords: complexity, boolean-width, branch decomposition, cut-function, mim-width

[1] Funded by the Norwegian Research Council.

[2] Email: `sigve.sether@ii.uib.no`

[3] Email: `martin.vatshelle@ii.uib.no`

1 Introduction

Many width parameters of graphs have been studies in recent years. Some of the most well known are treewidth, clique-width, branch-width and rank-width, all of which are NP-hard to compute [1,4,9,6]. Treewidth, branch-width and rank-width can all be computed in FPT time, whereas it is a long-standing open problem if computing clique-width is FPT or W[1]-hard. In contrast, for the parameters boolean-width and mim-width, no hardness results have been shown (for definitions and an overview of these parameters, see [10]). In this paper, we show that both mim-width and boolean-width are NP-hard to compute and also that mim-width is W[1]-hard. To our knowledge, this is the first width parameter of graphs based on non-linear decompositions that have been shown to be W[1]-hard to compute. Both of these parameters are defined using branch decompositions over the vertex set, and a cut-function. The cut-function of both boolean-width and mim-width are known to be NP-hard to compute. We give a reduction from deciding the value of these cut-functions to the problem of deciding the boolean-width and mim-width, respectively. Our result is a general reduction not only applicable to boolean-width and mim-width, but to all width parameters based on branch decompositions where the cut-function in question satisfies certain constraints. One of these constraints being that the value of the cut function should not increase when adding a twin vertex to the graph.

Our reduction preserves the parameter to within a constant factor of the original decision problem, so many parameterized hardness results will also translate to parameterized hardness of the width parameter in question. For instance we get W[1]-hardness of computing mim-width, and no polynomial time constant factor approximation for mim-width (unless NP = ZPP), because of similar hardness results for computing the mim-width of any particular cut.

Our main result is the following theorem, which follows from Lemma 3.1 and 3.2 described later. Using this theorem, we are able to show hardness results of computing f-width for any cut-function f from a large class of functions we name *$\mathcal{C}$-satisfying cut-functions*, as long as computing f on a single cut is hard. The graph $G_k(A)$ is a specific graph we can construct in polynomial time and is described in detail later.

Theorem 1.1 *Given a graph G, a subset $A \subseteq V(G)$, a $\mathcal{C}$-satisfying cut function f and a non-negative number $k \in \mathbb{R}$, the graph $G_{\lfloor k \rfloor+1}(A)$ has f-width at most $k + \lfloor k \rfloor + 1$ if and only if $f_G(A)$ is at most k.*

So given a graph G and subset $A \subseteq V(G)$, answering the question "is $f_G(A) \leq k$?" can be done by instead answering the question "for $t = \lfloor k \rfloor + 1$, does $G_t(A)$ have f-width at most $k + t$?"

Using Theorem 1.1 in combination with known NP-hardness results for counting the number of Maximal Independent Sets in a bipartite graph $G[A, \overline{A}]$ (equivalent to counting $2^{\text{bool}(A)}$, by Rabinovich et al. [8]) and deciding the size of a maximum induced matching in a bipartite graph $G[A, \overline{A}]$ (equivalent to computing $\text{mim}(A)$), by Provan and Ball [7] and Cameron [2], respectively, we get the following corollary.

Corollary 1.2 *Both deciding the mim-width of a graph, and deciding the boolean-width of a graph is NP-hard.*

By Moser and Sikdar [5] showing that finding a maximum induced matching in a bipartite graph is W[1]-hard.

Corollary 1.3 *Deciding the mim-width of a graph is W[1]-hard.*

By Elbassioni et al. [3], deciding the size of a maximum induced matching in a bipartite graph is not in APX unless NP=ZPP, which gives us the following corollary.

Corollary 1.4 *There is no polynomial time algorithm for approximating the mim-width of a graph to within a constant factor of the optimal, unless NP = ZPP.*

2 Preliminaries and terminology

For a graph G and vertex v, we denote its set of neighbours as $N_G(v)$, and denote by $N_G[v]$ the set $N_G(v) \cup \{v\}$. For a set $S \subseteq V(G)$, we denote by $N_G[S]$ and $N_G(S)$ the sets $\bigcup_{s \in S} N_G[s]$ and $N_G[S] \setminus S$, respectively. For a subset $S \subseteq V(G)$, the graph $G[S]$ is an *induced* subgraph G, with vertex set S and edge set $\{uv \in E(G) : u, v \in S\}$. For disjoint subsets $S_1, S_2 \subseteq V(G)$, by $G[S_1, S_2]$ we denote the induced bipartite subgraph of G with vertex set $S_1 \cup S_2$ and edge set $\{uv \in E(G) : u \in S_1, v \in S_2\}$. Two vertices $u, v \in V(G)$ are said to be *twins* in G if $N_G(v) = N_G(u)$, and u is a *twin vertex* of v.

For a grid graph G, we denote by C_i and R_i its i-th column and row, respectively. A subdivided grid graph is a graph resulting from replacing each edge uv in a grid by a vertex v_{uv} with neighbourhood $N_G(v_{uv}) = \{u, v\}$. In this paper, we refer to the vertices added by this operation as *sub-vertices*. The non sub-vertices we refer to as *cell-vertices*. For a subdivided grid, we denote by C_i and R_i the same set of vertices as C_i and R_i denote in the original grid graph (i.e., cell-vertices). For a set of cell-vertices X, we denote by $\mathtt{sub}(X)$ the set of sub-vertices adjacent to exactly two vertices of X. For two sets X_1, X_2 of cell-vertices, we denote by $\mathtt{sub}(X_1, X_2)$ the set of sub-vertices with one neighbour in X_1 and one neighbour in X_2.

Given a graph G, a *cut function* is a function f_G of the form $f_G : 2^{V(G)} \to \mathbb{R}$. The value $f_G(A)$ is the *f-value* of A with respect to G. For disjoint sets $A, B \subseteq V(G)$, we might write $f_G(A, B)$ to mean $f_{G[A \cup B]}(A)$. If there is no ambiguity, we omit the subscript. A *cut* is a bipartition (A, B) of $V(G)$. We might abuse notation slightly and refer to a cut simply by a set $A \subseteq V(G)$. In that case we mean the cut $(A, V(G) \setminus A)$.

A branch decomposition over the vertices of a graph is a pair (T, δ) where T is a sub-cubic (maximum degree three) tree and δ is a bijection from the leaves of T to the vertices in $V(G)$. Each edge $e = uv$ in T can be seen to separate T into two subtrees T_u and T_v by the operation $T - e$. We say that an edge $e = uv$ in T *induces* a cut of $V(G)$; namely the bipartition with one part consisting of all the vertices of $V(G)$ mapped (by σ) from the leaves of T_u and one part consisting of the vertices mapped from the leaves of T_v. So a decomposition (T, δ) induces $|E(T)|$ cuts.

The f-width of a branch decomposition $\mathcal{D} = (T, \delta)$ (denoted $f(\mathcal{D})$) is the maximum f-value over all the cuts induced by edges in $E(T)$. The f-width of a graph G (denoted $f(G)$) is the minimum f-width over all branch decompositions of $V(G)$.

The width-parameter *Boolean-width*, is an f-width where f is defined by $f = \text{bool}$, where

$$\text{bool}_G(A) = \log_2 \begin{pmatrix} \text{number of inclusion-wise maximal independent} \\ \text{sets in } G[A, \overline{A}] \end{pmatrix}.$$

The width-parameter *mim-width* (Minimum Induced Matching-width), is an f-width where f is defined by $f = \text{mim}$, where

$$\text{mim}_G(A) = \text{size of a maximum induced matching in } G[A, \overline{A}].$$

An induced matching is in induced subgraph of only degree one vertices. If there is no ambiguity, we might omit subscripts.

3 Deciding cut value through graph width

In this section we will show that we can reduce the problem of deciding the value of a cut-function f on a cut to the problem of deciding the f-width of a graph.

The idea of how to achieve such a reduction is that we construct, based on the input graph G and cut (A, B), a new graph consisting of a subdivided grid of known f-width, and attach copies of A to the left-hand side of the grid, and copies of B to the right-hand side of the grid. The grid will enforce the existence of a cut separating A from B in any optimal decomposition, making us able to deduce the value of $f_G(A, B)$.

In order to enforce a cut such as mentioned above, we cannot allow all kinds of cut-functions. In fact, we need our cut-functions to satisfy the following constraints in order to work. However, these constraints are upheld by cut functions of many known width-parameters defined through branch decompositions. If a cut function satisfies the below constraints $\mathcal{C}$, we say that it is a $\mathcal{C}$-*satisfying* cut-function. The constraints $\mathcal{C}$ are as follows, and must hold for any graph G and any set $S \subseteq V(G)$:

(i) $f_G(S) = f_G(\overline{S})$ and $f_G(S)$ depends only on the unlabeled graph $G[S, \overline{S}]$.

(ii) $f_G(S)$ is zero if $G[S, \overline{S}]$ has no edges and at least one otherwise.

(iii) Removing a vertex $x \in \overline{S}$ from G does not increase $f_G(S)$, and reduces $f_G(S)$ by at most one.

(iv) If $G[S, \overline{S}]$ is the disjoint union of $G_1 = G[A_1, B_1]$ and $G_2 = G[A_2, B_2]$, then $f_G(S) = f_{G_1}(A_1) + f_{G_2}(A_2)$.

(v) If $v \in \overline{S}$ has a twin vertex in $G[S, \overline{S}]$, then $f_G(S) = f_{G-v}(S)$.

On most known width parameters defined using branch decompositions over $V(G)$, all but the last constraint is upheld, as they are natural properties that come as a result of wanting to measure how many objects of a certain kind lies between the two parts of a cut. The last constraint is the only real limitation of the cut-parameters we investigate.

As a result of the four first constraints, any $\mathcal{C}$-satisfying cut function f on $A \subseteq V(G)$ will always have value at least as large as a maximum induced matching M in $G[A, \overline{A}]$ (i.e. $f(A) \geq mim(A)$), since removing all vertices other than those in M does not decrease the f-value, and we then have $|M|$ disjoint graphs of at least one edge, implying an f-value of at least $|M|$.

Some examples of cut functions that are $\mathcal{C}$-satisfying are the cut functions of the width-parameters mim-width, boolean-width, and rank-width.

To prove Theorem 1.1 we show that given a graph G, a cut A and non-negative integer k, we can in polynomial time construct a graph $G_k(A)$ for which the f-width of $G_k(A)$ is no more than $k + f_G(A)$ and no less than $\min\{2k, k + f_G(A)\}$. This upper and lower bound is proved by Lemma 3.1 and Lemma 3.2, respectively. However, first we must define our graph $G_k(A)$.

The graph $G_k(A)$.

Given a graph G, a cut $(A, B = \overline{A})$ of G and an integer k, we construct $G_k(A)$ as follows. We first start with a subdivided grid G' of height k and width $6k$. Then, for each vertex $a \in A$, we add to $G_k(A)$ a set S_a of k vertices, and let $S_A = \bigcup_{a \in A} S_a$. Similarly, for each vertex $b \in B$, we add to $G_k(A)$ a k-vertex set S_b and let S_B denote the union $\bigcup_{b \in B} S_b$. Then, for each $a \in A$ we add edges making up a matching between the vertices of S_a and the set C_1 of G' and for each $b \in B$ a matching between S_b and C_{6k}. Now we add edges between the vertices of S_A and S_B in such a manner that the induced subgraph on $S_A \cup S_B$ will be the graph $G[A, B]$ with the addition that each vertex has $k-1$ extra twins. That is, we add the edges $E' = \{uv : u \in S_a, v \in S_b, a \in A, b \in B\}$. So, the vertices of $G_k(A)$ are $V(G_k(A)) = V(G') \cup S_A \cup S_B$ and the edges are $E(G') \cup E'$ plus a matching from S_a to C_1 for each $a \in A$, and a matching from S_b to C_{6k} for each $b \in B$.

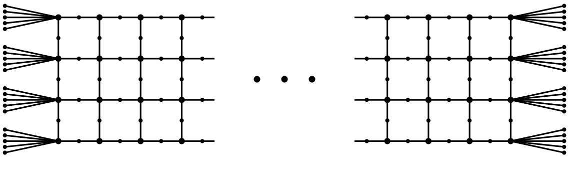

Fig. 1. The graph $G_4(A)$ for some set $A \subset V(G)$ so that $|A| = |\overline{A}| = 5$. Edges between S_A and $S_{\overline{A}}$ are omitted in this figure.

We now show the first part of proving Theorem 1.1, namely upper bounding the f-width of $G_k(A)$.

Lemma 3.1 *Given a graph G and subset $A \subseteq V(G)$, the f-width of $G_k(A)$ for a $\mathcal{C}$-satisfying cut function f is at most $f_G(A) + k$, for any non-negative integer k.*

Proof (Sketch) We prove this lemma by construction. The idea is to decompose $G_k(A)$ from left to right, starting in S_A, going through the subdivided grid column by column, and ending in $S_{\overline{A}}$, as shown in Figure 2. For each cut of the decomposition, we can show that removing at most k vertices leaves a

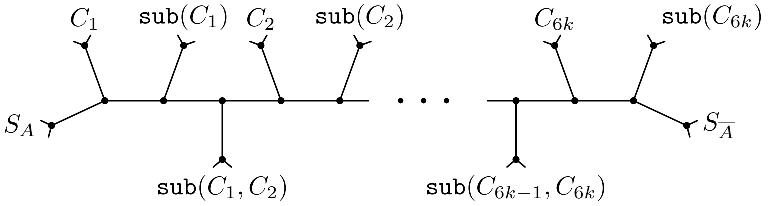

Fig. 2. A high-level view of the decomposition of width at most $f_G(A)+k$ described in Lemma 3.1.

cut of width $f_G(A)$, which by the constraints of $\mathcal{C}$ means the entire decomposition has width at most $f_G(A) + k$. □

We now do the last part for proving Theorem 1.1, namely giving a lower bound on the f-width of $G_k(A)$.

Lemma 3.2 *Given a graph G, a non-negative integer k and subset $A \subseteq V(G)$, for any $\mathcal{C}$-satisfying cut function f we have $f(G_k(A)) \geq \min\{2k,\ f_G(A)+k\}$.*

Proof (Sketch/idea) Let $G' = G_k(A)$. We show that in any decomposition of f-width less than $2k$, there must be a cut (X_1, X_2) so that two columns are entirely contained in X_1, and two columns are entirely contained in X_2. Removing vertices between the two leftmost columns and the two rightmost columns out of these four columns, we disconnect the parts Q_1 and Q_2 depicted in figure 3. We show that the cut (X_1, X_2) restricted only to the vertices of Q_1 will have width at least k. We further show that the graph $G'[X_1 \cap Q_2, X_2 \cap Q_2]$ contains as an induced subgraph a graph isomorphic to $G[A, \overline{A}]$, implying $f(X_1, X_2)$ restricted to Q_2 is at least $f_G(A)$, and thus $f(X_1, X_2) \geq k + f_G(A)$. □

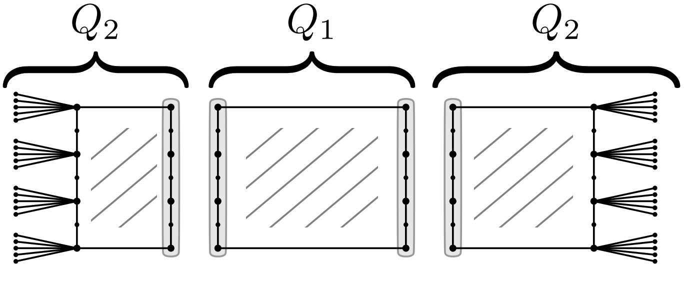

Fig. 3. The four columns (marked in grey) mentioned in the proof sketch of Lemma 3.2 act as a separator between the parts Q_1 and Q_2 depicted. (Again, edges from S_A to $S_{\overline{A}}$ are omitted from the drawing.)

This completes the part of proving Theorem 1.1, as we now have a strict (enough) bound on the f-width of $G_k(A)$ to be able to tie its value to the value of $f_G(A)$.

References

[1] Arnborg, S., D. G. Corneil and A. Proskurowski, *Complexity of finding embeddings in ak-tree*, SIAM Journal on Algebraic Discrete Methods **8** (1987), pp. 277–284.

[2] Cameron, K., *Induced matchings*, Discrete Applied Mathematics **24** (1989), pp. 97–102.

[3] Elbassioni, K. M., R. Raman, S. Ray and R. Sitters, *On the approximability of the maximum feasible subsystem problem with 0/1-coefficients.*, in: *SODA*, SIAM, 2009, pp. 1210–1219.

[4] Fellows, M. R., F. A. Rosamond, U. Rotics and S. Szeider, *Clique-width is np-complete*, SIAM Journal on Discrete Mathematics **23** (2009), pp. 909–939.

[5] Moser, H. and S. Sikdar, *The parameterized complexity of the induced matching problem*, Discrete Applied Mathematics **157** (2009), pp. 715–727.

[6] Oum, S.-I., *Approximating rank-width and clique-width quickly*, TALG **5** (2008), p. 10.

[7] Provan, J. S. and M. O. Ball, *The complexity of counting cuts and of computing the probability that a graph is connected*, SIAM Journal on Computing **12** (1983), pp. 777–788.

[8] Rabinovich, Y., J. A. Telle and M. Vatshelle, *Upper bounds on boolean-width with applications to exact algorithms*, in: *Parameterized and Exact Computation*, Springer, 2013 pp. 308–320.

[9] Seymour, P. D. and R. Thomas, *Call routing and the ratcatcher*, Combinatorica **14** (1994), pp. 217–241.

[10] Vatshelle, M., “New width parameters of graphs,” Ph.D. thesis, The University of Bergen (2012).