A Unifying Framework for Characterizing and Computing Width Measures

Eduard Eiben
Department of Computer Science, Royal Holloway, University of London, Egham, UK

Robert Ganian
Algorithms and Complexity Group, TU Wien, Austria

Thekla Hamm
Algorithms and Complexity Group, TU Wien, Austria

Lars Jaffke
Department of Informatics, University of Bergen, Norway

O-joung Kwon
Department of Mathematics, Incheon National University, South Korea
Discrete Mathematics Group, Institute for Basic Science, Daejeon, South Korea

Abstract

Algorithms for computing or approximating optimal decompositions for decompositional parameters such as treewidth or clique-width have so far traditionally been tailored to specific width parameters. Moreover, for mim-width, no efficient algorithms for computing good decompositions were known, even under highly restrictive parameterizations. In this work we identify $\mathcal{F}$-branchwidth as a class of generic decompositional parameters that can capture mim-width, treewidth, clique-width as well as other measures. We show that while there is an infinite number of $\mathcal{F}$-branchwidth parameters, only a handful of these are asymptotically distinct. We then develop fixed-parameter and kernelization algorithms (under several structural parameterizations) that can approximate *every possible* $\mathcal{F}$-branchwidth, providing a unifying parameterized framework that can efficiently obtain near-optimal tree-decompositions, k-expressions, as well as optimal mim-width decompositions.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases branchwidth, parameterized algorithms, mim-width, treewidth, clique-width

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.63

Related Version *Full Version*: https://arxiv.org/abs/2109.14610

Funding *Robert Ganian*: Robert Ganian acknowledges support by the Austrian Science Fund (FWF, projects P31336 and Y1329).
Thekla Hamm: Thekla Hamm acknowledges support by the Austrian Science Fund (FWF, projects P31336, Y1329 and W1255-N23).
Lars Jaffke: Lars Jaffke acknowledges support by the Norwegian Research Council (NFR, project 274526).
O-joung Kwon: O-joung Kwon was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education (No. NRF- 2018R1D1A1B07050294), and by the Institute for Basic Science (IBS-R029-C1).

1 Introduction

Over the past twenty years, the study of decompositional graph parameters and their algorithmic applications has become a focal point of research in theoretical computer science. *Treewidth* [30] is, naturally, the most prominent example of such a parameter, but is far from

13th Innovations in Theoretical Computer Science Conference (ITCS 2022).
Editor: Mark Braverman; Article No. 63; pp. 63:1–63:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the only parameter of interest: graph classes of unbounded treewidth may still have bounded *clique-width* [10], and graph classes of unbounded clique-width may still have bounded *mim-width* [24, 33]. A common feature of such parameters is that they are all defined using some notion of a "decomposition" which is also of importance in various algorithmic applications, although the precise definitions of such decompositions vary from parameter to parameter.

Given their importance, it is not surprising that finding efficient algorithms for computing suitable decompositions has become a prominent research direction in the field. While computing optimal decompositions has proven to be a challenging task, in many scenarios it is sufficient to simply obtain an approximately-optimal decomposition instead – i.e., we ask for an algorithm which, given a graph of "width" at most k, outputs a decomposition witnessing that the "width" is at most $f(k)$ for some function f. Such approximately-optimal decompositions are often obtained by finding an asymptotically equivalent reformulation of the width parameter[1]. Indeed, there has been a series of fixed-parameter algorithms for computing approximately-optimal tree decompositions [4], and the efficient computation of approximately-optimal decompositions for clique-width was a long-standing open problem that has culminated in the introduction of rank-width [28], a parameter which is asymptotically equivalent to clique-width. On the other hand, up to now approximately-optimal mim-width decompositions could only be computed in a few highly restrictive settings [22, 31].

Algorithmic applications of treewidth, clique-width, and mim-width have a number of unifying characteristics. For instance, many well-studied graph classes such as interval graphs and permutation graphs as well as several of their generalizations have constant mim-width, with efficiently computable decompositions [1]. Therefore, efficient algorithms for problems on graphs of constant mim-width often unify a number of ad-hoc results on these graph classes. This applies to, e.g., all locally checkable vertex problems [9] including Independent Set, Dominating Set and many of its variants, H-Homomorphism, and Odd Cycle Transversal; distance versions [23], and connected and acyclic variants of these problems [2], among others Feedback Vertex Set and problems related to finding induced paths. Another unification behaviour is observed in the algorithmic framework [2] which is capable of exploiting multiple of these measures at once. However, to the best of our knowledge no unifying framework for the task of computing the respective decompositions was known. Indeed, while all three measures admit asymptotically-equivalent characterizations via the notion of *branchwidth* [25, 28], which is a generic template for defining width parameters based on a notion of *cut-functions*, the cut-functions employed in these previously known formulations were fundamentally different.

Contribution. We introduce the notion of $\mathcal{F}$-branchwidth, which is a restriction of branchwidth to cut functions defined via an infinite class $\mathcal{F}$ of obstructions; more precisely, $\mathcal{F}$ can be any class of bipartite graphs with a basic property which we call *partner-hereditarity* ($\mathtt{ph}$). Intuitively, a class $\mathcal{F}$ of bipartite graphs is $\mathtt{ph}$ if every graph in $\mathcal{F}$ has its vertex set partitioned into pairs of vertices (one on each side), and each subgraph induced on a subset of the pairs is also in $\mathcal{F}$. As our first result, we show that treewidth, clique-width, and mim-width are all asymptotically equivalent to $\mathcal{F}$-branchwidth for suitable choices of $\mathcal{F}$.

Taken on its own, this result is not surprising: partner-hereditarity is a weak restriction that simply allows graph classes to serve as obstructions in cut-functions, there are infinitely many $\mathtt{ph}$ graph classes, and some happen to characterize these well-studied decompositional parameters. For our second result, we consider an additional useful property of our obstruction classes: $\mathcal{F}$ is *size-identifiable* ($\mathtt{si}$) if there exists a unique graph in $\mathcal{F}$ of each order. We show:

[1] Two width parameters are asymptotically equivalent if on every graph class, they are either both bounded or both unbounded.

▶ **Result 1.** *There exist precisely six* $\texttt{si ph}$ *classes. Moreover, for each* $\texttt{ph}$ *class* $\mathcal{F}$*,* $\mathcal{F}$*-branchwidth is asymptotically equivalent to* $\mathcal{F}^{\downarrow}$*-branchwidth, where* $\mathcal{F}^{\downarrow}$ *is the restriction of* $\mathcal{F}$ *to the* $\texttt{si ph}$ *classes contained in* $\mathcal{F}$*.*

The six $\texttt{si ph}$ classes comprise the classes of *complete bipartite*, *matching* and *chain* graphs along with their bipartite complements. Essentially, Result 1 means that all decompositional parameters which can be defined using $\texttt{ph}$ obstruction classes can be characterized by a combination of these $\texttt{si}$ classes, with treewidth, clique-width and mim-width being three of these. While some other combinations of obstruction classes do not give rise to interesting parameters (e.g., it is easy to verify that the parameter obtained by excluding the *complete bipartite* class and its complement class is only bounded for bounded-size graphs), other combinations of obstruction classes lead to completely new structural graph parameters which might have unique properties and have, to the best of our knowledge, never been studied (e.g., excluding the *anti-matching* class or the *chain* class).

What is more, we show that this unification result can be strengthened even further when the aim is to compute approximately-optimal decompositions for $\mathcal{F}$-branchwidth instead of characterizing the widths themselves. We identify three “$\texttt{primal}$” $\texttt{si ph}$ classes – the classes of matching, chain, and bipartite complements of matchings – and show:

▶ **Result 2.** *For a* $\texttt{ph}$ *class* $\mathcal{F}$*, let* $\mathcal{F}^*$ *be the restriction of* $\mathcal{F}$ *to the* $\texttt{primal ph}$ *classes contained in* $\mathcal{F}$*. Then every optimal* $\mathcal{F}^*$*-branch decomposition is also an approximately-optimal* $\mathcal{F}$*-branch decomposition.*

Result 2 allows us to reduce the computation of approximately-optimal $\mathcal{F}$-branch decompositions for any $\mathcal{F}$ to combinations of just three $\texttt{primal ph}$ classes. For instance, treewidth and clique-width have different characterizations in terms of $\texttt{si ph}$ classes, but computing decompositions for both parameters corresponds to a single case in our framework. Computing decompositions for mim-width then corresponds to another case.

Having reduced the computation of this wide range of decompositional parameters to $\mathcal{F}^*$-Branchwidth, i.e., the problem of computing an optimal $\mathcal{F}^*$-branch decomposition for some union $\mathcal{F}^*$ of $\texttt{primal ph}$ classes, the second part of our paper is dedicated to solving this problem. Obtaining an algorithm for $\mathcal{F}^*$-Branchwidth when parameterized by the target width itself is a problem that not only unifies the approximation of treewidth and clique-width, but also the long-standing open problem of computing mim-width (which is known to be W[1]-hard [31] and for which even XP-tractability is open). But while we cannot hope for fixed-parameter tractability under this parameterization, we make substantial progress by considering other parameters: our contribution includes three novel fixed-parameter algorithms for $\mathcal{F}^*$-Branchwidth that not only give a unified platform for computing multiple width parameters, but also push the boundaries of tractability specifically for the notoriously difficult problem of computing mim-width [22, 31, 34].

▶ **Result 3.** *Let* $\mathcal{F}^*$ *be the union of some* $\texttt{primal ph}$ *families.* $\mathcal{F}^*$*-*Branchwidth*:*

1. *is fixed-parameter tractable when parameterized by treewidth and the maximum degree,*
2. *is fixed-parameter tractable when parameterized by treedepth, and*
3. *admits a linear kernel when parameterized by the feedback edge set number.*

Organization of the Paper. In Section 3 we introduce $\mathcal{F}$-branchwidth. In Section 4, we substantiate $\mathcal{F}$-branchwidth as a unifying parameter by comparing it to previously studied parameters and identifying six *size identifiable* graph classes as a foundation that covers the whole breadth of $\mathcal{F}$-branchwidth. Sections 5, 6 and 7 include an outline of our three unifying algorithms and their correctness proofs.

2 Preliminaries

For an integer i, we let $[i] = \{1, 2, \ldots, i\}$. We refer to the handbook by Diestel [12] for standard graph terminology. We also refer to the standard books for a basic overview of parameterized complexity theory [11, 14]. Two graph parameters a, b are *asymptotically equivalent* if for every graph class $\mathcal{G}$, either both a and b are bounded by some constant, or neither is. $\Delta(G)$ denotes the maximum vertex degree in G. The *length* of a path refers to the number of edges on that path.

Ramsey Theory. Given an edge-colored graph G a subset of vertices $C \subseteq V(G)$ is a *monochromatic clique* of color c in G if the induced subgraph $G[C]$ of G is complete and every edge in $G[C]$ is colored by color c.

▶ **Fact 2.1** (Ramsey's Theorem, see, e.g., [19])**.** *Let $n_1, n_2, \ldots, n_q \in \mathbb{N}$. There exists an integer R such that for every edge-colored complete graph G on R vertices with q colors $1, \ldots, q$ there exists $i \in [q]$ such that G contains a monochromatic clique of color i and size n_i.*

We denote by $R(n_1, n_2, \ldots, n_q)$ the minimum integer that satisfies the above fact and call such a number (multicolor) Ramsey Number. We note the following well known bounds on $R(n_1, n_2, \ldots, n_q)$.

▶ **Fact 2.2** (Erdős and Szekeres [15])**.** $R(n_1, n_2) \leq \binom{n_1+n_2-1}{n_1-1} \leq 2^{n_1+n_2}$.

▶ **Fact 2.3.** $R(n_1, n_2, \ldots, n_q) \leq R(n_1, n_2, \ldots, n_{q-2}, R(n_{q-1}, n_q))$.

Treewidth. A *nice tree-decomposition* $\mathcal{T}$ of a graph $G = (V, E)$ is a pair (T, χ), where T is a tree (whose vertices we call *nodes*) rooted at a node r and χ is a function that assigns each node t a set $\chi(t) \subseteq V$ such that the following holds:

- For every $uv \in E$ there is a node t such that $u, v \in \chi(t)$.
- For every vertex $v \in V$, the set of nodes t satisfying $v \in \chi(t)$ forms a subtree of T.
- $|\chi(\ell)| = 1$ for every leaf ℓ of T and $|\chi(r)| = 0$.
- There are only three kinds of non-leaf nodes in T:

  Introduce node: a node t with exactly one child t' such that $\chi(t) = \chi(t') \cup \{v\}$ for some vertex $v \notin \chi(t')$.

  Forget node: a node t with exactly one child t' such that $\chi(t) = \chi(t') \setminus \{v\}$ for some vertex $v \in \chi(t')$.

  Join node: a node t with two children t_1, t_2 such that $\chi(t) = \chi(t_1) = \chi(t_2)$.

The *width* of a nice tree-decomposition (T, χ) is the size of a largest set $\chi(t)$ minus 1, and the *treewidth* of the graph G, denoted $\mathrm{tw}(G)$, is the minimum width of a nice tree-decomposition of G. Efficient fixed-parameter algorithms are known for computing a nice tree-decomposition of near-optimal width [4, 26].

▶ **Proposition 2.4** (Bodlaender et al. [4])**.** *There exists an algorithm which, given an n-vertex graph G and an integer k, in time $2^{\mathcal{O}(k)} \cdot n$ either outputs a tree-decomposition of G of width at most $5k + 4$ and $\mathcal{O}(n)$ nodes, or determines that $\mathrm{tw}(G) > k$.*

Treedepth and Feedback Edge Set. A *rooted forest* is a forest in which every component has a specified node called a *root*. The *closure* of a rooted forest T is the graph on $V(T)$ in which two vertices are adjacent if and only if one is a descendant of the other in T. The *height* of a rooted forest is the number of vertices in a longest root-to-leaf path in T. The *treedepth* of a graph G, denoted by $\operatorname{td}(G)$, is the minimum height of a rooted forest whose closure contains G as a subgraph.

▶ **Fact 2.5** (Nešetřil and Ossona de Mendez [27]). *Given an n-vertex graph G and a constant w, it is possible to decide whether G has treedepth at most w, and if so, to compute an optimal treedepth decomposition of G in time $\mathcal{O}(n)$.*

A set $Q \subseteq E(G)$ of edges in a graph G is called a *feedback edge set* if $G - Q$ is acyclic. The *feedback edge set number* of G is the minimum size of a feedback edge set of G and can be computed efficiently.

▶ **Fact 2.6.** *A minimum feedback edge set of a graph G can be obtained by deleting the edges of minimum spanning trees of all connected components of G, and hence can be computed in time $\mathcal{O}(|E(G)| + |V(G)|)$.*

Both measures have been used to obtain fixed-parameter algorithms and/or polynomial kernels in a number of applications [3, 16, 17, 20].

Branchwidth and Branch Decompositions. While branchwidth and branch decompositions have also been defined over general ground sets, here we introduce the definition that matches the most common usage scenario on graphs. A set function $f : 2^M \to \mathbb{Z}$ over a ground set M is called *symmetric* if $f(X) = f(M \setminus X)$ for all $X \subseteq M$. A tree is *subcubic* if all its nodes have degree at most 3.

A *branch decomposition* of a graph G is a subcubic tree $\mathfrak{B}$ whose leaves are mapped to the vertices of G via a bijective mapping $\mathcal{L}$. For an edge e of $\mathfrak{B}$, the connected components of $\mathfrak{B} - e$ induce a bipartition (X_e, Y_e) of the set of leaves of $\mathfrak{B}$, and through $\mathcal{L}$ also of $V(G)$. For a symmetric function $f : 2^{V(G)} \to \mathbb{Z}$, the *width* of e is $f(X_e)$, the width of the branch decomposition $\mathfrak{B}$ is the maximum width over all edges of $\mathfrak{B}$, and the branchwidth of G is the minimum width of a branch decomposition of G. Here, f is often called the *cut-function*, and we use $G[X, Y]$ to denote the bipartite graph obtained from G by removing all edges that do not have precisely one endpoint in X and precisely one endpoint in Y.

Well-known applications of this definition include mim-width [24, 33], where the cut-function is the size of a maximum induced matching in $G[X, Y]$. Another example is maximum-matching width, which is asymptotically equivalent to treewidth [25] and where the cut-function is the size of a maximum (but not necessarily induced) matching in $G[X, Y]$. Rank-width is another well-known example of branchwidth; there, the cut-function is the rank of the bipartite adjacency matrix induced by (X, Y). Rank-width is asymptotically equivalent to clique-width [28].

In Sections 5 and 6, it will be useful to consider the *restrictions* of (the tree structure of) a branch decomposition $(\mathfrak{B}, \mathcal{L})$. For a tree T and a set $A \subseteq V(T)$ we define the *restricted tree* of T with respect to A as the tree $\operatorname{restr}_T(A)$ that arises from the minimal subtree of T containing all vertices in A by recursively contracting all edges uv where u has degree 2 and is not in A.

Note that for a branch decomposition $(\mathfrak{B}, \mathcal{L})$ the restricted tree with respect to subset of leaves $A \subseteq \mathcal{L}(V(G))$ of a branch decomposition of G can be seen as a branch decomposition of $G[A]$. Conversely a branch decomposition $(\mathfrak{B}_1, \mathcal{L}_1)$ of G *extends* a branch decomposition $(\mathfrak{B}_2, \mathcal{L}_2)$ of $G[A]$, if $\operatorname{restr}_{\mathfrak{B}_1}(A) = \mathfrak{B}_2$ and $\mathcal{L}_1(a) = \mathcal{L}_2(a)$ for $a \in A$.

Note also that every edge $xy \in E(\mathrm{restr}_{\mathfrak{B}}(A))$ can be understood as the contraction of the path P between x and y in $\mathfrak{B}$. We also say that the edge xy of the restricted tree with respect to X of $\mathfrak{B}$ *corresponds* to P.

To streamline the presentation in some sections (such as Section 5), we suppress the function $\mathcal{L}$ of a branch decomposition $\mathfrak{B}$ by assuming that the leaves of branch decompositions are actually the leaves of G.

Typical Sequences. Typical sequences, introduced in [6], will be an important tool in the FPT algorithm for computing $\mathcal{F}^*$-branchwidth parameterized by treewidth and vertex degree. We provide a brief overview over important definitions and results.

▶ **Definition 2.7** (E.g. Bodlaender and Kloks [6]). *Let $s = (s_1, \ldots, s_\ell)$ be a sequence of natural numbers (including 0) of length ℓ. The* typical sequence *$\tau(s)$ of s is obtained from s by an iterative exhaustive application of the following two operations:*

1. *Removing consecutive repetitions: If there is an index $i \in [\ell - 1]$ such that $s_i = s_{i+1}$, we change the sequence s from $(s_1, \ldots, s_\ell)$ to $(s_1, \ldots, s_i, s_{i+2}, \ldots, s\ell)$.*
2. *Typical operation: If there are $i, j \in [\ell]$ such that $j - i \geq 2$ and for all $i \leq k \leq j$, $s_i \leq s_k \leq s_j$, or for all $i \leq k \leq j$, $s_i \geq s_k \geq s_j$ then we change the sequence s from $s_1, \ldots, s_\ell$ to $s_1, \ldots, s_i, s_j, \ldots, s_\ell$, i.e. we remove all entries of s (strictly) between index i and j.*

We say a sequence of natural numbers is typical *whenever it is identical to its typical sequence.*

While it is maybe not immediately obvious from the definition, the typical sequence of a sequence of natural numbers is always unique. As we use typical sequences to index table entries or *records* in a dynamic program in Section 5 and also branch on ways to modify them, the following combinatorial bounds will be useful.

▶ **Fact 2.8** (Bodlaender and Kloks [6]).

- *The length of the typical sequence of a sequence with entries in $\{0\} \cup [k]$ is at most $2k + 1$.*
- *The number of typical sequences with entries in $\{0\} \cup [k]$ is at most $\frac{8}{3}2^{2k}$.*

It is known that typical sequences of a sequence of natural numbers can be computed in linear time.

▶ **Fact 2.9** (Bodlaender et al. [5]). *The typical sequence of a sequence of natural numbers of length ℓ can be computed in time in $\mathcal{O}(\ell)$.*

For the correctness of our dynamic program it is helpful to note some trivial observations: The typical sequence of a concatenation of two sequences s and s' is the typical sequence of the concatenation of the typical sequences of s and s'. Moreover, the typical sequence of a sequence $(s_1 + z, \ldots, s_\ell + z)$ that arises from a typical sequence by adding some constant $z \in \mathbb{N}$ to every entry of a sequence s is equal to the sequence that arises by adding z to each entry of the typical sequence of s. We also write $s + z$ for a sequence s of length ℓ and a natural number z to denote the sequence $(s_1 + z, \ldots, s_\ell + z)$.

A definition which will be crucial at the join nodes in our dynamic program is that of the $\oplus$-operator for typical sequences.

▶ **Definition 2.10** (Bodlaender and Kloks [6]). *Let s be a sequence of ℓ natural numbers. We define the set $E(s)$ of* extensions *of s as the set of sequences that are obtained from s by repeating each of its elements an arbitrary number of times, and at least once. Now we define the* interleaving *of two typical sequences s and t of arbitrary length to be the set $s \oplus t = \{\tau((\tilde{s}_1 + \tilde{t}_1, \ldots, \tilde{s}_\ell + \tilde{t}_\ell)) \mid \tilde{s} \in E(s), \tilde{t} \in E(t), |\tilde{s}| = |\tilde{t}| = \ell\}$, where we use $|r|$ to denote the length of a sequence r.*

▶ **Fact 2.11** (Thilikos et al. [32]). *The interleaving of two typical sequences with entries in $[k]$ can be computed in time in $\mathcal{O}(k^4 2^{4k(2k+1)})$.*

3 $\mathcal{F}$-Branchwidth

In this paper we consider branchwidth and branch decompositions where the value of the cut-function $f(X)$ is defined by the maximum number of vertices of any induced subgraph of $G[X, V(G) \setminus X]$ that is isomorphic to some graph in some fixed infinite family of bipartite graphs. We focus on families of bipartite graphs where each part has the same size satisfying a certain closure property. Let $\mathcal{F}$ be an infinite family of bipartite graphs where each graph in $\mathcal{F}$ has bipartition $(A = (a_1, \dots, a_n), B = (b_1, \dots, b_n))$. We say that $\mathcal{F}$ is partner-hereditary (*ph*) if for each $2n$-vertex graph in $\mathcal{F}$ and each subset L of $[n]$, the graph induced on $\{a_i, b_i | i \in L\}$ is isomorphic to a graph in $\mathcal{F}$. Note that the vertices in both parts of the bipartite graph in $\mathcal{F}$ are ordered. Given a bipartite graph $H = (V, E)$ with bipartition $(A = (a_1, \dots, a_n), B = (b_1, \dots, b_n))$, for each $i \in [n]$ we call a_i and b_i *partners*. Moreover, for simplicity, when referring to a bipartite graph H with bipartition (A, B) and edge set E we will sometimes write $H = (A, B, E)$.

For a *ph* family of bipartite graphs $\mathcal{F}$, we define $\mathcal{F}$*-branchwidth* as the branchwidth where the cut-function $\mathcal{L}$ of a cut (X, Y) is defined as the largest n such that a $2n$-vertex graph in $\mathcal{F}$ is isomorphic to an induced subgraph of $G[X, Y]$. For a graph G, a branch decomposition $\mathfrak{B}$ of G, and an edge e of $\mathfrak{B}$, we denote by $\mathcal{F}\text{-bw}(\mathfrak{B}, e)$ the width of the cut of G *induced* by e, by $\mathcal{F}\text{-bw}(\mathfrak{B})$ the width of $\mathfrak{B}$ and we denote by $\mathcal{F}\text{-bw}(G)$ the $\mathcal{F}$-branchwidth of G. We begin by noting the following simple observations that follow directly from the fact that the width of an edge depends only on the existence of some graph in $\mathcal{F}$ as induced subgraph across the cut.

▶ **Observation 3.1.** *Let $\mathcal{F}$ be a ph family of graphs, G a graph, and H an induced subgraph of G, then $\mathcal{F}\text{-bw}(H) \leq \mathcal{F}\text{-bw}(G)$.*

▶ **Observation 3.2.** *Let $\mathcal{F}, \mathcal{F}'$ be two ph families of graphs such that there are only finitely many graphs in $\mathcal{F}' \setminus \mathcal{F}$ and let G be a graph. Then $\mathcal{F}'\text{-bw}(G) \leq \max\{\mathcal{F}\text{-bw}(G), |F| \mid F \in \mathcal{F}' \setminus \mathcal{F}\}$.*

We say that two families of graphs $\mathcal{F}_1$ and $\mathcal{F}_2$ are *isomorphic* if for every graph H_1 in $\mathcal{F}_1$, the class $\mathcal{F}_2$ contains a graph isomorphic to H_1 and for every graph H_2 in $\mathcal{F}_2$, the class $\mathcal{F}_1$ contains a graph isomorphic to H_2. Since the ordering on the vertices in $G[X, Y]$ for a partition (X, Y) of the vertices of G is not important to determine the width of the cut, we get the following observation.

▶ **Observation 3.3.** *Let $\mathcal{F}_1, \mathcal{F}_2$ be two isomorphic ph families of graphs and G a graph, then $\mathcal{F}_1\text{-bw}(G) = \mathcal{F}_2\text{-bw}(G)$.*

It will be useful to introduce the following six special families of *ph* classes. In the following, we denote a bipartite graph with vertex bipartition (A, B) and edge set E by (A, B, E). Let:

1. $H^n_\emptyset = ((a_1, \dots, a_n), (b_1, \dots, b_n), \emptyset)$ and let $\mathcal{F}_\emptyset = \{H^n_\emptyset \mid n \in \mathbb{N}\}$, i.e., $\mathcal{F}_\emptyset$ contains all edgeless graphs on an even number of vertices.
2. $H^n_\equiv = ((a_1, \dots, a_n), (b_1, \dots, b_n), E_\equiv)$ where $E_\equiv = \{a_i b_i \mid i \in [n]\}$ and let $\mathcal{F}_\equiv = \{H^n_\equiv \mid n \in \mathbb{N}\}$, i.e., $\mathcal{F}_\equiv$ contains all 1-regular graphs (matchings).
3. $H^n_\lesseqqgtr = ((a_1, \dots, a_n), (b_1, \dots, b_n), E_\lesseqqgtr)$ where $E_\lesseqqgtr = \{a_i b_j \mid i \leq j \in [n]\}$ and let $\mathcal{F}_\lesseqqgtr = \{H^n_\lesseqqgtr \mid n \in \mathbb{N}\}$, i.e., $\mathcal{F}_\lesseqqgtr$ contains all bipartite chain graphs without twins.

4. $H^n_{\lessdot} = ((a_1, \ldots, a_n), (b_1, \ldots, b_n), E_{\lessdot})$ where $E_{\lessdot} = \{a_ib_j \mid i < j \in [n]\}$ and let $\mathcal{F}_{\lessdot} = \{H^n_{\lesssim} \mid n \in \mathbb{N}\}$, i.e., $\mathcal{F}_{\lessdot}$ contains each graph in $\mathcal{F}_{\lesssim}$ without the matching.
5. $H^n_{\bowtie} = ((a_1, \ldots, a_n), (b_1, \ldots, b_n), E_{\bowtie})$ where $E_{\bowtie} = \{a_ib_j \mid i \neq j \in [n]\}$ and let $\mathcal{F}_{\bowtie} = \{H^n_{\bowtie} \mid n \in \mathbb{N}\}$, i.e., $\mathcal{F}_{\bowtie}$ contains for every n the $2n$-vertex $(n-1)$-regular bipartite graph (all anti-matchings).
6. $H^n_{\boxtimes} = ((a_1, \ldots, a_n), (b_1, \ldots, b_n), E_{\boxtimes})$ where $E_{\boxtimes} = \{a_ib_j \mid i, j \in [n]\}$ and let $\mathcal{F}_{\boxtimes} = \{H^n_{\boxtimes} \mid n \in \mathbb{N}\}$, i.e., $\mathcal{F}_{\boxtimes}$ contains all complete bipartite graphs $K_{n,n}$.

We will later prove that to compute an approximately-optimal branch decomposition for any ***ph*** family $\mathcal{F}$, it suffices to compute an optimal branch decomposition for $\mathcal{F}^*$, where $\mathcal{F}^*$ is the union of a subset of classes $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\lesssim}$, and $\mathcal{F}_{\bowtie}$ that are fully contained in $\mathcal{F}$ (see Lemma 4.13).

4 Size-Identifiable Classes and Comparison to Existing Measures

Here, we focus on the six families introduced at the end of Section 3. We first show that while there is an infinite number of possible ***ph*** graph classes, to asymptotically characterize $\mathcal{F}$-branchwidth for any ***ph*** family of graphs $\mathcal{F}$ we only need to consider these six families. Afterwards, we turn our attention on several previously studied graph measures and show how they compare to $\mathcal{F}$-branchwidth. Finally we show that to compute an approximate $\mathcal{F}^*$-branch decomposition, it suffices to focus on only three of these classes, namely $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\lesssim}$, and $\mathcal{F}_{\bowtie}$.

4.1 Size-Identifiable Classes

We say that $\mathcal{F}$ is size-identifiable (***si***) if for each $n \in \mathbb{N}$, there is a single $2n$-vertex graph in $\mathcal{F}$ (up to isomorphism). We will show $\mathcal{F}_{\emptyset}$, $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\lesssim}$, $\mathcal{F}_{\lessdot}$, $\mathcal{F}_{\bowtie}$, and $\mathcal{F}_{\boxtimes}$ are the only ***si*** ***ph*** families of graph. To show that these are the only ***si*** ***ph*** families as well as to show importance of these families, we will make use of the following lemma that basically shows that any bipartite graph with ordered partitions contains a large subset of partners that induce a graph isomorphic to a graph in one of the six above families.

▶ **Lemma 4.1.** *Let $q \geq R(2n, 2n, 2n, 2n)$ and let $H^q = ((a_1, \ldots, a_q), (b_1, \ldots, b_q), E)$ be a bipartite graph. Then there exists $L \subseteq [q]$ such that $|L| = n$ and the graph induced on $\{a_i, b_i \mid i \in L\}$ is isomorphic to one of $H^n_{\emptyset}$, $H^n_{\equiv}$, $H^n_{\lesssim}$, $H^n_{\lessdot}$, $H^n_{\bowtie}$, or $H^n_{\boxtimes}$.*

Proof. To prove the lemma, we will construct an auxiliary edge-colored complete graph G on q vertices such that there is bijection between the vertices of G and partners a_i, b_i in H^q. Moreover, a monochromatic clique in G will correspond to one of $H^n_{\emptyset}$, $H^n_{\equiv}$, $H^n_{\lesssim}$, $H^n_{\bowtie}$, or $H^n_{\boxtimes}$. Let G be the edge-colored graph such that $V(G) = (v_1, v_2, \ldots, v_n)$ and the edge v_iv_j, $1 \leq i < j \leq q$, has color

- 1 if both edges a_ib_j, a_jb_i are in E,
- 2 if the edge a_ib_j is in E and the edge a_jb_i is not in E,
- 3 if the edge a_jb_i is in E and the edge a_ib_j is not in E,
- 4 if both edges a_ib_j, a_jb_i are not in E,

Since $q \geq R(2n, 2n, 2n, 2n)$, by Ramsey's Theorem G contains a monochromatic clique on $2n$ vertices. Note that either at least n vertices of the clique correspond to partners that are adjacent to each other, or at least n vertices of the clique correspond to partners that are not adjacent to each other. Let C be a monochromatic clique on n vertices such that either $a_ib_i \in E$ for each $v_i \in C$ (i.e., all partners in C are matched), or $a_ib_i \notin E$ for each $v_i \in C$ (i.e., all partners in C are not matched). Let us now consider each of the following possibilities:

Color 1 and partners are adjacent. It is easy to verify that the graph induced on $\{a_i, b_i \mid i \in L\}$ is isomorphic with $H^n_{\boxtimes}$.

Color 1 and partners are not adjacent. Then the graph induced on $\{a_i, b_i \mid i \in L\}$ contains an edge a_i, b_j if and only if $i \neq j$ and it is isomorphic with $H^n_{\times}$.

Color 2 and partners are adjacent. Then the graph induced on $\{a_i, b_i \mid i \in L\}$ contains an edge a_i, b_j if and only if $i \leq j$ and it is isomorphic with $H^n_{\leqq}$.

Color 2 and partners are not adjacent. Then the graph induced on $\{a_i, b_i \mid i \in L\}$ contains an edge a_i, b_j if and only if $i < j$ and it is isomorphic with $H^n_{\lessdot}$.

Color 3 and partners are adjacent. Then the graph induced on $\{a_i, b_i \mid i \in L\}$ contains an edge a_i, b_j if and only if $i \geq j$ and it is isomorphic with $H^n_{\leqq}$. Note that here we have to change the order of the pairs to obtain $H^n_{\leqq}$.

Color 3 and partners are not adjacent. Then the graph induced on $\{a_i, b_i \mid i \in L\}$ contains an edge a_i, b_j if and only if $i > j$ and it is isomorphic with $H^n_{\lessdot}$.

Color 4 and partners are adjacent. Then the graph induced on $\{a_i, b_i \mid i \in L\}$ contains an edge a_i, b_j if and only if $i = j$ and it is isomorphic with $H^n_{\equiv}$.

Color 4 and partners are not adjacent. Then it is easy to verify that the graph induced on $\{a_i, b_i \mid i \in L\}$ is isomorphic with $H^n_{\emptyset}$. ◀

▶ **Lemma 4.2.** *There are precisely six* **si** *and* **ph** *families: $\mathcal{F}_{\emptyset}$, $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\leqq}$, $\mathcal{F}_{\lessdot}$, $\mathcal{F}_{\times}$, $\mathcal{F}_{\boxtimes}$.*

Proof. It is rather straightforward to verify that each of the families $\mathcal{F}_{\emptyset}$, $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\leqq}$, $\mathcal{F}_{\lessdot}$, $\mathcal{F}_{\times}$, $\mathcal{F}_{\boxtimes}$ is **ph** and **si**. Let $\mathcal{F}$ be an **si** and **ph** graph family and let H^n denote the unique $2n$-vertex graph in $\mathcal{F}$. First note that if H^n is in one of the six families described above (i.e., for example $H^n = H^n_{\equiv}$) then for all $m < n$ the graph H^m is in the same family of the graphs. Indeed for each of the six families the bipartite subgraph of H^n induced on the first m partners is in the same family. Since $\mathcal{F}$ is **ph**, this subgraph is in $\mathcal{F}$ and since $\mathcal{F}$ is also **si**, this subgraph is isomorphic to H^m. To finish the proof, it suffices to show that H^n is always one of the the following six graphs: $H^n_{\emptyset}$, $H^n_{\equiv}$, $H^n_{\leqq}$, $H^n_{\lessdot}$, $H^n_{\times}$, or $H^n_{\boxtimes}$. Indeed, let $q = R(2n, 2n, 2n, 2n)$ and let us consider the graph H^q. By Lemma 4.1 there exists $L \subseteq [q]$ such that $|L| = n$ and the subgraph of H^q induced on $\{a_i, b_i \mid i \in L\}$ is isomorphic to one of $H^n_{\emptyset}$, $H^n_{\equiv}$, $H^n_{\leqq}$, $H^n_{\lessdot}$, $H^n_{\times}$, or $H^n_{\boxtimes}$. Since $\mathcal{F}$ is **ph**, $\mathcal{F}$ contains a graph isomorphic to one of $H^n_{\emptyset}$, $H^n_{\equiv}$, $H^n_{\leqq}$, $H^n_{\lessdot}$, $H^n_{\times}$, or $H^n_{\boxtimes}$ and since $\mathcal{F}$ is **si**, this graph is indeed isomorphic to H^n. ◀

We now apply Ramsey's Theorem to show that these six **si** classes capture the whole hierarchy of width parameters induced by $\mathcal{F}$-branchwidth.

▶ **Lemma 4.3.** *Let $\mathcal{F}$ be a* **ph** *graph class and let $\mathcal{F}'$ be the union of all* **si** *families of which all but finitely many elements are contained in $\mathcal{F}$. Then $\mathcal{F}$-branchwidth is asymptotically equivalent to $\mathcal{F}'$-branchwidth.*

Proof. Since $\mathcal{F}'$ is contained in $\mathcal{F}$ up to a finite number of elements, it follows from Observation 3.2 that for every graph G it holds that $\mathcal{F}'\text{-bw}(G) \leq \mathcal{F}\text{-bw}(G) + q$, where $q = \max\{|F| \mid F \in \mathcal{F}' \setminus \mathcal{F}\}$ is a constant. It hence suffices to show that $\mathcal{F}\text{-bw}(G)$ is upper-bounded by a function of $\mathcal{F}'\text{-bw}(G)$.

Let q be the size of the largest graph H such that $H \in \mathcal{F} \setminus \mathcal{F}'$ and H is isomorphic to one of $H^q_{\emptyset}$, $H^q_{\equiv}$, $H^q_{\leqq}$, $H^q_{\lessdot}$, $H^q_{\times}$, or $H^q_{\boxtimes}$; if no such graph H exists, we set q to 0. Note that, in any case, $\mathcal{F}$ does not contain all but finitely many graphs in the **si** class containing H and q

is a constant. Let us consider a branch decomposition $\mathfrak{B}$ of G such that $\mathcal{F}'\text{-}\operatorname{bw}(\mathfrak{B}) = k$ and let $n = 1 + \max(q, k)$. We will show that $\mathcal{F}\text{-}\operatorname{bw}(\mathfrak{B}) < R(2n, 2n, 2n, 2n)$, which will complete the proof.

For the sake of contradiction, let us assume that this is not the case and that there is an edge $e \in \mathfrak{B}$ such that $\mathcal{F}\text{-}\operatorname{bw}(\mathfrak{B}, e) \geq R(2n, 2n, 2n, 2n)$. Let X, Y be the partition of $V(G)$ induced by e. Then $G[X, Y]$ has an induced subgraph on $2m$ vertices, for some $m \geq R(2n, 2n, 2n, 2n)$, that is isomorphic to some graph H^m in $\mathcal{F}$. However, $m \geq R(2n, 2n, 2n, 2n)$ and by Lemma 4.1, there is a subset $L \subseteq [m]$ such that $|L| = n$ and the graph H^n induced on $\{a_i, b_i \mid i \in L\}$ is isomorphic to one of $H^n_{\emptyset}$, $H^n_{\equiv}$, $H^n_{\lesseqgtr}$, $H^n_{\lessdot}$, $H^n_{\bowtie}$, or $H^n_{\divideontimes}$. Since $n > q$, H^n is isomorphic to a graph in an ***si*** family fully contained in $\mathcal{F}$. But then $\mathcal{F}'\text{-}\operatorname{bw}(\mathfrak{B}, e) \geq n > k$, contradicting $\mathcal{F}'\text{-}\operatorname{bw}(\mathfrak{B}) = k$. ◀

4.2 Comparison to Existing Measures

Next, we discuss how specific ***si ph*** classes capture previously studied parameters. First, mim-width is precisely $\mathcal{F}_{\equiv}$-branchwidth.

▶ **Observation 4.4.** *For every graph G, mim-width of G is equal to $\mathcal{F}_{\equiv}$-branchwidth of G.*

The next decomposition parameter that we consider is treewidth. To show the equivalence, we actually show that *maximum-matching width*, which is always linearly upper- and lower-bounded by treewidth [25, 33], is also $\mathcal{F}$-branchwidth for the ***ph*** family $\mathcal{F}$ of all ***ph*** graphs that contain a matching on the partners.

▶ **Lemma 4.5.** *There exists a **ph** family $\mathcal{F}$ such that for all graphs G it holds $\mathcal{F}\text{-bw}(G) \leq \operatorname{tw}(G) + 1 \leq 3 \cdot \mathcal{F}\text{-bw}(G)$.*

By Lemma 4.3, treewidth is then asymptotically equivalent to $\mathcal{F}^*$-branchwidth for $\mathcal{F}^*$ being the union of some ***si ph*** classes, notably $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\lesseqgtr}$, $\mathcal{F}_{\bowtie}$, and $\mathcal{F}_{\divideontimes}$. By applying the easy observation that $\mathcal{F}' \subseteq \mathcal{F}$ implies $\mathcal{F}'\text{-}\operatorname{bw}(G) \leq \mathcal{F}\text{-bw}(G)$ for any ***ph*** families $\mathcal{F}$ and $\mathcal{F}'$ and any graph G, we can also establish an explicit upper bound when considering $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\lesseqgtr}$ and $\mathcal{F}_{\bowtie}$. The reason we are interested in specifically this set of families will become clear when we define `primal` families.

▶ **Corollary 4.6.** *Let $\mathcal{F}^*$ be a union of any combination of families among $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\lesseqgtr}$, and $\mathcal{F}_{\bowtie}$ and let G be a graph. Then $\mathcal{F}^*\text{-}\operatorname{bw}(G) \leq \operatorname{tw}(G) + 1$.*

Throughout Section 5 and in Proposition 6.1 in Section 6, the fact that $\mathcal{F}^*\text{-bw}(G)$ is bounded in the respective parameters is essential, and Corollary 4.6 gives this bound.

The next parameter that we will consider is clique-width. Similarly as for treewidth, it will be easier to show equivalence with another parameter, *module-width*, which is known to be asymptotically equivalent to cliquewidth [7, 29]. An important tool to establish this equivalence is Corollary 2.4 in [13], which intuitively states that any sufficiently large twin-free bipartition of vertices in a graph must contain a large chain, anti-matching or matching.

▶ **Lemma 4.7.** *Let $\mathcal{F} = \mathcal{F}_{\equiv} \cup \mathcal{F}_{\bowtie} \cup \mathcal{F}_{\lesseqgtr}$. Then there exists a computable function f such that $\mathcal{F}\text{-bw}(G) \leq \operatorname{modw}(G) \leq f(\mathcal{F}\text{-bw}(G))$ for every graph G; in particular, module-width and $\mathcal{F}$-branchwidth with $\mathcal{F} = \mathcal{F}_{\equiv} \cup \mathcal{F}_{\bowtie} \cup \mathcal{F}_{\lesseqgtr}$ are asymptotically equivalent.*

Finally, we compare $\mathcal{F}$-branchwidth to the so-called H-join decompositions introduced by Bui-Xuan, Telle, and Vatshelle [8]. We note that unlike other parameters, H-join decompositions do not have a specific “width”; each graph either admits a decomposition or not, and the complexity measure is captured by the order of the graph H. Keeping this in mind, we can show that $\mathcal{F}$-branchwidth captures H-joins as well.

▶ **Lemma 4.8.** *Let $\mathcal{F} = \mathcal{F}_{\equiv} \cup \mathcal{F}_{\bowtie} \cup \mathcal{F}_{\lesseqgtr}$ and let H be a fixed graph. Then for all H-join decomposable graphs G it holds that $\mathcal{F}\text{-bw}(G) \leq |V(H)|$.*

4.3 Primal Families and Computing $\mathcal{F}$-Branchwidth

Let us call the *ph* families $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\lesseqgtr}$, and $\mathcal{F}_{\bowtie}$ *primal*. As the last result of this section, we can build on Lemma 4.3 and show that it suffices to focus our efforts to compute good $\mathcal{F}$-branch decompositions merely on unions of *primal* classes. To this end, one can prove the following simple claim.

▶ **Lemma 4.9.** *Let $\mathcal{F}$ be a union of si ph classes. If $\mathcal{F}_{\lessdot} \subseteq \mathcal{F}$, then for every graph G it holds that $|\mathcal{F}\text{-bw}(G) - \mathcal{F}'\text{-bw}(G)| \leq 1$, where $\mathcal{F}' = (\mathcal{F} \setminus \mathcal{F}_{\lessdot}) \cup \mathcal{F}_{\lesseqgtr}$.*

Proof. Let X, Y be a partition of $V(G)$ and let $n \in \mathbb{N}$ be the largest integer such that $G[X,Y]$ contains an induced subgraph isomorphic to $H^n_{\lesseqgtr}$. If $H^n_{\lesseqgtr} = ((a_1, \dots, a_n), (b_1, \dots, b_n), E_{\lesseqgtr})$, then it is easy to see that $((a_2, \dots, a_n), (b_1, \dots, b_{n-1}), E_{\lesseqgtr})$ is isomorphic to $H^{n-1}_{\lessdot}$. On the other hand if $m \in \mathbb{N}$ is the largest integer such that $G[X,Y]$ contains an induced subgraph isomorphic to $H^m_{\lessdot} = ((a_1, \dots, a_m), (b_1, \dots, b_m), E_{\lessdot})$, then $((a_1, \dots, a_{m-1}), (b_2, \dots, b_m), E_{\lessdot})$ is isomorphic to $H^{m-1}_{\lesseqgtr}$. ◀

To prove that $\mathcal{F}_{\emptyset}$ and $\mathcal{F}_{\bowtie}$ are redundant we will use the following well-known fact that for any subset of leaves of branch decomposition $\mathfrak{B}$ there is an edge of $\mathfrak{B}$ that separates the subset in balanced way. Let $\mathfrak{B} = (V, E)$ be a subcubic tree and consider a nonnegative weight function $\mathbf{w} : V \to \mathbb{R}_{\geq 0}$. Let e be an edge in $\mathfrak{B}$ and $\mathfrak{B}_1, \mathfrak{B}_2$ be the two connected components of $\mathfrak{B} - e$. We say that e is α-balanced, for $0 < \alpha \leq \frac{1}{2}$, if $\alpha \cdot \mathbf{w}(\mathfrak{B}) \leq \mathbf{w}(\mathfrak{B}_1) \leq (1 - \alpha) \cdot \mathbf{w}(\mathfrak{B})$, where $\mathbf{w}(\mathfrak{B}) = \sum_{v \in V(\mathfrak{B})} \mathbf{w}(v)$ (note that $\mathbf{w}(\mathfrak{B}) = \mathbf{w}(\mathfrak{B}_1) + \mathbf{w}(\mathfrak{B}_2)$ so also $\alpha \cdot \mathbf{w}(\mathfrak{B}) \leq \mathbf{w}(\mathfrak{B}_2) \leq (1 - \alpha) \cdot \mathbf{w}(\mathfrak{B})$). The following lemma is folklore and we include the proof only for completeness.

▶ **Lemma 4.10.** *Let $\mathfrak{B} = (V, E)$ be a subcubic tree and consider a nonnegative weight function $\mathbf{w} : V \to \mathbb{R}_{\geq 0}$ on vertices of $\mathfrak{B}$. Then there exists a $\frac{1}{3}$-balanced edge in $\mathfrak{B}$.*

Proof. Let us orient the edges of $\mathfrak{B}$ as follows. Let e be an edge in $\mathfrak{B}$ and $\mathfrak{B}_1, \mathfrak{B}_2$ be the two connected components of $\mathfrak{B} - e$. We orient the edge from $\mathfrak{B}_1$ to $\mathfrak{B}_2$ if $\mathbf{w}(\mathfrak{B}_1) < \frac{1}{2} \cdot \mathbf{w}(\mathfrak{B})$ and from $\mathfrak{B}_2$ to $\mathfrak{B}_1$ otherwise. The directed graph obtained in this way is acyclic and it has a sink, i.e., vertex of out-degree zero. It is straightforward to verify that there is an edge incident to the sink such that $\frac{1}{3} \cdot \mathbf{w}(\mathfrak{B}) \leq \mathbf{w}(\mathfrak{B}_1) \leq \frac{1}{2} \cdot \mathbf{w}(\mathfrak{B})$. ◀

Using the above lemma we get the following corollaries.

▶ **Corollary 4.11.** *Let $n \in \mathbb{N}$ and let G be a graph such that there exists a partition (A, B) of $V(G)$ with $H^{3n}_{\emptyset}$ isomorphic to an induced subgraph H of $G[A, B]$. Then each branch decomposition $\mathfrak{B}$ of G has an edge e that induces a partition (X, Y) of G such that $H^n_{\emptyset}$ is isomorphic to an induced subgraph of $G[X, Y]$.*

Proof. Let $H = (A_H, B_H, E_H)$ such that $A_H = A \cap V(H)$ and $B_H = B \cap V(H)$. Let $\mathbf{w} : V(\mathfrak{B}) \to \mathbb{R}_{\geq 0}$ such that $\mathbf{w}(v) = 1$ if $v = \mathcal{L}(u)$ for some vertex $u \in V(H)$ and $\mathbf{w}(v) = 0$, otherwise. Let e be a $\frac{1}{3}$-balanced edge of $\mathfrak{B}$ and let (X, Y) be the partition of G induced by e such that $\mathbf{w}(X) \leq \mathbf{w}(Y)$. Then $\frac{|V(H)|}{3} \leq |V(H) \cap X| \leq \frac{|V(H)|}{2}$. Furthermore, let us assume, without loss of generality, that $|A_H \cap X| \geq |B_H \cap X|$. Clearly, $|A_H \cap X| \geq n$ and $|B_H \cap X| \leq 2n$, hence $|B_H \cap Y| \geq n$. It is easy to verify that $(A_H \cap X) \cup (B_H \cap Y)$ induces a graph isomorphic to $H^n_{\emptyset}$. ◀

Following an analogous argument for $\mathcal{F}_{\bowtie}$ we get.

▶ **Corollary 4.12.** *Let $n \in \mathbb{N}$ and let G be a graph such that there exists a partition (A, B) of $V(G)$ with $H^{3n}_{\bowtie}$ isomorphic to an induced subgraph H of $G[A, B]$. Then each branch decomposition $\mathfrak{B}$ of G has an edge e that induces a partition (X, Y) of G such that $H^{n}_{\bowtie}$ is isomorphic to an induced subgraph of $G[X, Y]$.*

Given the above two corollaries it is straightforward to show the following lemma.

▶ **Lemma 4.13.** *Let $\mathcal{F}$ be a union of a subset of* **`si ph`** *classes other than $\mathcal{F}_{\lesssim}$. Let $\mathcal{F}^*$ be the union of all* **`primal`** *families in $\mathcal{F}$. Then an optimal $\mathcal{F}^*$-branch decomposition of a graph G is a 3-approximate $\mathcal{F}$-branch decomposition of G.*

Proof. Let $\mathfrak{B}$ be an optimal $\mathcal{F}^*$-branch decomposition and let e be an edge in $\mathfrak{B}$ that induces partition (X, Y) of G. If $\mathcal{F}^*\text{-bw}(\mathfrak{B}, e) < \mathcal{F}\text{-bw}(\mathfrak{B}, e) = n$, then either

- $G[X, Y]$ contains an induced subgraph isomorphic to $H^n_{\emptyset}$ and $\mathcal{F}_{\emptyset} \subseteq \mathcal{F} \setminus \mathcal{F}^*$, or
- it contains an induced subgraph isomorphic to $H^n_{\bowtie}$ and $\mathcal{F}_{\bowtie} \subseteq \mathcal{F} \setminus \mathcal{F}^*$.

In both cases, we have $\mathcal{F}\text{-bw}(G) \geq \frac{n}{3}$ by Corollaries 4.11 and 4.12. ◀

We use $\mathcal{F}^*$ to refer to the union of any combination of **`primal`** families, and $\mathcal{F}^*$ -bw(G) to refer to the $\mathcal{F}^*$-branchwidth of G. If follows from Lemmas 4.9 and 4.13 that computing $\mathcal{F}^*$-branchwidth is sufficient to obtain approximately optimal decompositions for all possible $\mathcal{F}$-branchwidths. We conclude this section by showing that for $\mathcal{F}^*$-Branchwidth, i.e., the problem of computing an optimal $\mathcal{F}^*$-branch decomposition for a fixed choice of $\mathcal{F}^*$, it is sufficient to deal with connected graphs only.

▶ **Lemma 4.14.** *Let $\mathcal{F}^*$ be a union of a non-empty set of* **`primal`** *families of graphs. For a graph G, the $\mathcal{F}^*$-branchwidth of G is the maximum $\mathcal{F}^*$-branchwidth over all connected components of G.*

Proof. We will prove the lemma by induction on the number of connected components. Clearly, if G is connected, then the statement hold. Now, if G be a disjoint union of graphs G_1 and G_2. Given a $\mathcal{F}^*$-branch decomposition $\mathfrak{B}$ of G of width k, it is easy to see that the restriction of $\mathfrak{B}$ to G_1 (resp. G_2), that is the branch decomposition obtained from $\mathfrak{B}$ by deleting the leaves that do not correspond to vertices of G_1 (resp. G_2), is a decomposition $\mathfrak{B}$ of G_1 (resp. G_2) of width k. Note that this shows that the $\mathcal{F}^*$-branchwidth of G is at least maximum of $\mathcal{F}^*$-branchwidth of G_1 and $\mathcal{F}^*$-branchwidth of G_2 which is in turn, by induction hypothesis, at least the maximum $\mathcal{F}^*$-branchwidth over all connected components of G.

On other hand, let $\mathfrak{B}_1$ and $\mathfrak{B}_2$ be $\mathcal{F}^*$-branch decompositions of G_1 and G_2 respectively. We construct a $\mathcal{F}^*$-branch decomposition of G of width $\max(\mathcal{F}^*\text{- bw}(\mathfrak{B}_1), \mathcal{F}^*\text{- bw}(\mathfrak{B}_2))$. This shows that $\mathcal{F}^*$-branchwidth of G is at most maximum of $\mathcal{F}^*$-branchwidth of G_1 and $\mathcal{F}^*$-branchwidth of G_2 which is by induction hypothesis the maximum $\mathcal{F}^*$-branchwidth over all connected components of G, finishing the proof.

Let e_1 be an arbitrary edge of $\mathfrak{B}_1$ and e_2 an arbitrary edge of $\mathfrak{B}_2$. Let us subdivide e_1 and e_2 and let the resulting vertices be w_1 and w_2, respectively. Note that subdivision of an edge in a branch decomposition does not change the width of the decomposition. Now let $\mathfrak{B}$ be the $\mathcal{F}^*$-branchwidth decomposition obtained by taking the disjoint union of $\mathfrak{B}_1$ and $\mathfrak{B}_2$ and adding the edge w_1w_2. Let us show that the width of every edge is bounded by $\max(\mathcal{F}^*\text{- bw}(\mathfrak{B}_1), \mathcal{F}^*\text{- bw}(\mathfrak{B}_2))$. First, consider the edge w_1w_2. The partition of $V(G)$ induced by this edge is $(V(G_1), V(G_2))$. There is no edge between $V(G_1)$ and $V(G_2)$. But $\mathcal{F}^*$ is a union of a non-empty set of **`primal`** families of graphs and every vertex in a bipartite graph in $\mathcal{F}^*$ has degree at least one. So no induced subgraph of $G[(V(G_1), V(G_2))]$ is isomorphic to a graph in $\mathcal{F}^*$ and the width of the edge w_1w_2 is 0. Now let e be an edge in $\mathfrak{B}_1$ and let (X, Y) be the partition of $V(G)$ induced by e. It is easy to see that, depending of which connected component of $\mathfrak{B}_1 - e$ contains w_1, we have either $V(G_2) \subseteq X$ or $V(G_2) \subseteq Y$. Let H be an induced subgraph of $G[X, Y]$ isomorphic to a graph in $\mathcal{F}^*$. It follows that every vertex in $V(H) \cap X$ has a neighbor in Y and every vertex in $V(H) \cap Y$ has a neighbor in X. Since either $V(G_2) \subseteq X$ or $V(G_2) \subseteq Y$ and there is no edge between G_1 and G_2, we have $V(H) \subseteq V(G_1)$ and $\mathcal{F}^*\text{-}\operatorname{bw}(\mathfrak{B}, e) = \mathcal{F}^*\text{-}\operatorname{bw}(\mathfrak{B}_1, e)$. An analogous argument for an edge in $\mathfrak{B}_2$ finishes the proof. ◀

5 Treewidth and Maximum Vertex Degree

Our aim in this section is to prove the following theorem.

▶ **Theorem 5.1.** *Let $\mathcal{F}^*$ be the union of some `primal ph` families. $\mathcal{F}^*$-Branchwidth is* FPT *parameterized by the treewidth and the maximum degree of the input graph.*

As an immediate corollary, by considering $\mathcal{F}^* = \mathcal{F}_{\equiv}$ we obtain:

▶ **Corollary 5.2.** *Mim-Width is* FPT *parameterized by the treewidth and the maximum degree of the input graph.*

On a high level, our proof of Theorem 5.1 relies on the usual dynamic programming approach which computes a set of records for each node $t \in V(T)$ of a minimum-width nice tree decomposition (T, χ) of G in a leaves-to-root manner. Here each record captures a set of partial $\mathcal{F}^*$-branch decompositions not exceeding $\mathcal{F}^*$-branchwidth at most $\operatorname{tw}(G)$. It is worth noting that the most involved and difficult parts of our algorithm are necessary to handle the case that $\mathcal{F}_{\equiv} \subseteq \mathcal{F}^*$.

For $t \in V(T)$, denote by T_t the subtree of T rooted at t, and by $G_t = G[\bigcup_{s \in V(T_t)} \chi(s)]$ the subgraph of G induced by the vertices in the bags of T_t. Moreover, for $v \in V(G)$ we denote its *closed distance-i neighbourhood* in G by $N^i[v]$, its *open distance-i neighbourhood in G by* $N^i(v)$, and also extend these notations to vertex sets $V' \subseteq V(G)$ by denoting $N^i[V'] = \bigcup_{v \in V'} N^i[v]$ and $N^i(V') = N^i[V'] \setminus V'$.

We define a *record* $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\ggg})$ of $t \in V(T)$ to consist of the following.

- A binary tree D on at most $2^{|N^3[\chi(t)]|}$ vertices, all leaves of which are identified with distinct vertices in $N^3[\chi(t)]$. Intuitively, D will describe the restricted tree with respect to $N^3[\chi(t)]$ of all branch decompositions of G_t that are captured by the record.
- A set $\flat = \{\flat_e\}_{e \in E(D)}$ indexed by the edges of D, where each $\flat_e$ is a sequence of subsets of $N^{3\operatorname{tw}(G)}(N^3[\chi(t)])$, such that $\bigcup_{e \in E(D)} \flat_e$ is a partition of $N^{3\operatorname{tw}(G)}(N^3[\chi(t)])$. Each $\flat_e$ describes the order in which subtrees containing vertices in $N^{3\operatorname{tw}(G)}(N^3[\chi(t)])$ are attached to the path which corresponds to e in any branch decomposition captured by $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\ggg})$. The entry $\flat$ can be interpreted to express a "refinement" of each edge of D. Similarly to edges of D corresponding to paths in a branch decomposition captured by the record we will define refinements of these paths in such branch decompositions into *blocks* which correspond to pairs (e, i) where $e \in E(D)$ and $i \in [|\flat_e| + 1]$.
- $\Lambda = (\Lambda_e)_{e \in E(D)}$ indexed by the edges of D, where for each $e \in E(D)$, Λ_e is a sequence $\Lambda_e = (\lambda_1, \dots, \lambda_{|\flat_e|+1})$ of subsets of $N^3(\chi(t))$. $\lambda_i \in \Lambda_e$ should be such that the $\mathcal{F}_{\equiv}$-bw value at every edge in the block corresponding to (e, i) in any branch decomposition captured by $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\ggg})$ can be achieved by a matching such that its intersection with $G[N^3[\chi(t)]]$ is incident to precisely the vertices in λ_i. In other words, $\lambda_i \in \Lambda_e$ describes

the interaction we assume any matching to have with $G[N^3[\chi(t)]]$ when computing the $\mathcal{F}_{\equiv}$-bw value at any edge in the block corresponding to (e,i). We can argue (Lemma 5.5) that such a set λ_i exists.

- A collection $\sigma = (\sigma_{e,i})_{e\in E(D), i\in[|\flat_e|+1]}$ of typical sequences $\sigma_{e,i}$ with entries in $\{0\} \cup [\mathrm{tw}(G)+1]$. $\sigma_{e,i}$ should describe a further refinement of each path corresponding to (e,i) in a branch decomposition captured by $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqqgtr})$ according to the $\mathcal{F}_{\equiv}$-bw values that can be achieved by the restriction of the branch decomposition to G_t under the condition imposed by the choice of λ_i. The importance of these typical sequences is that we can argue that it is safe to assume that a partial branch decomposition at node t can be extended to a minimum-$\mathcal{F}^*$-bw decomposition of G if and only if this is possible by attaching vertices only at positions in the typical sequences (Lemma 5.7).
- $\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqqgtr} \in [\mathrm{tw}(G)+1]$ should describe the $\mathcal{F}_{\equiv}$-bw, $\mathcal{F}_{\divideontimes}$-bw and $\mathcal{F}_{\lesseqqgtr}$-bw of any branch decomposition of G_t captured by $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqqgtr})$. For $\alpha^{\equiv}$ there is a slight caveat – to compute this entry in the record we will rely on the correctness of Λ which we cannot ensure at the time of computation, but rather retroactively after the root of T is processed. This is done by constructing a *canonical decomposition* concurrently with each record, which is representative of all branch decompositions captured by $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqqgtr})$, and then after processing the root of T verifying that its $\mathcal{F}_{\equiv}$-bw is at most $\alpha^{\equiv}$.

Note that the number of records is easily seen to be FPT parameterized by $\mathrm{tw}(G)$ and $\Delta(G)$ using the fact that the number of considered typical sequences is at most $\frac{8}{3}2^{2(\mathrm{tw}(G)+1)}$ [6]. We remark that the information stored in D is sufficient to keep track of the $\mathcal{F}_{\divideontimes}$-bw and the $\mathcal{F}_{\lesseqqgtr}$-bw of the constructed partial decomposition (this means to update $\alpha^{\divideontimes}$ and $\alpha^{\lesseqqgtr}$) appropriately. For $\mathcal{F}_{\equiv}$-bw the "forgotten" vertices, i.e. vertices in $V(G_t) \setminus \chi(t)$ have a more far-reaching impact. More explicitly, any graphs in $\mathcal{F}_{\divideontimes}$ and $\mathcal{F}_{\lesseqqgtr}$ induced in a cut of a partial branch decomposition which contains a vertex in $\chi(t)$, can only contain vertices in $N^3[\chi(t)]$.

Obviously, this is not true for graphs in $\mathcal{F}_{\equiv}$ which can have infinite diameter.

Hence, for $\mathcal{F}_{\equiv}$-bw instead of keeping track of vertices which can occur in graphs in $\mathcal{F}^*$ together with vertices in $\chi(t)$, we store the rough location in the constructed partial branch decomposition of a sufficiently large neighbourhood $\tilde{N}$ of $\chi(t)$ in G to separate the interaction (in terms of independence) of edges of $G_t[V(G_t) \setminus \tilde{N}]$ and edges of $G[V(G) \setminus (V(G_t) \cup \tilde{N})]$ that occur in independent induced matchings at cuts in the constructed partial branch decomposition. After this separation we still need to be able to actually derive the $\mathcal{F}_{\equiv}$-bw. To do this we use typical sequences in a similar way as they are also used to compute e.g. treewidth, pathwidth and cutwidth [5, 6, 21, 32], as indicated in the description of σ above.

In the remainder of this subsection we give an overview of the technical results and formal definitions that justify our approach. For this fix a node $t \in V(T)$ of a tree decomposition (T, χ) of G, let $\mathfrak{B}$ be a branch decomposition of G of minimum $\mathcal{F}^*$-branchwidth, $D = \mathrm{restr}_{\mathfrak{B}}(N^3[\chi(t)])$, for an edge $e \in E(D)$, P_e denote the path in $\mathfrak{B}$ that corresponds to e, and $\flat$ be such that for every $e \in E(D)$, (1) the subtree of $\mathfrak{B} - P_e$ containing $v \in N^{3\,\mathrm{tw}(G)}(N^3[\chi(t)])$ is attached to an internal node of P_e if and only if $v \in \bigcup_{d\in\flat_e} d$; and (2) for $v, w \in \bigcup_{d\in\flat_e} d$, v is contained in a subtree of $\mathfrak{B} - P_e$ attached at a node *before* (in an arbitrary but fixed traversal of P_e) the node at which the subtree of $\mathfrak{B} - P_e$ containing w is attached if and only if the set in $\flat_e$ that contains v is before the set in $\flat_e$ that contains w in the sequence $\flat_e$.

▶ **Definition 5.3.** *For $e \in E(D)$ which corresponds to the path P_e in $\mathfrak{B}$, each maximal subpath of P_e which does not have internal nodes at which subtrees of $\mathfrak{B} - P_e$ containing vertices in $N^{3\,\mathrm{tw}(G)}(N^3[\chi(t)])$ are attached is called a* block *of e.*

▶ **Observation 5.4.** *There are $|\flat_e| + 1$ blocks of an edge $e \in E(D)$.*

In this way we can immediately relate each block to a pair (e, i) where $e \in E(D)$ and $i \in [|\flat_e| + 1]$, where we enumerate the blocks of e according to the fixed traversal of P_e. For the remainder of this subsection we fix an edge $e \in E(D)$ and its corresponding path P_e in $\mathfrak{B}$.

Next we show that blocks have the desirable property that we can consider induced matchings with uniformly restricted interaction with $N^3[\chi(t)]$ when computing the $\mathcal{F}_{\equiv}$-bw values at edges of a fixed block. Even stronger this restricted interaction does not depend on the internal structure of a block which is important for the application of typical sequences.

▶ **Lemma 5.5.** *Let P' be a block of P_e. Then there is some $\lambda \subseteq N^3[\chi(t)]$ such that the following holds. Consider an arbitrary branch decomposition $\tilde{\mathfrak{B}}$ of G that arises from $\mathfrak{B}$ by deleting all subtrees of $\mathfrak{B} - P_e$ that are attached at internal vertices of P' and reattaching binary trees with the same cumulative set of leaves at internal nodes of P' or subdivisions of edges of P'. Denote by $\tilde{P}$ the path in $\tilde{\mathfrak{B}}$ which corresponds to the path of all (possibly subdivided) edges of P'. Then for every edge $\tilde{e} \in E(\tilde{P})$ there is an induced matching $\tilde{M}$ in the bipartite subgraph of G induced by $\tilde{e}$ with $2\mathcal{F}_{\equiv}\text{-bw}(\mathfrak{B}, \tilde{e})$ vertices such that the edges in $E(\tilde{M}[N^3[\chi(t)]])$ are incident to exactly λ.*

Lemma 5.5 justifies the following definition.

▶ **Definition 5.6.** *We say that a block P' of P_e is a* λ-block *where $\lambda \subseteq N^3[\chi(t)]$ satisfies the properties described in Lemma 5.5.*

Note that the entry Λ in our dynamic programming records acts as a guess for which sets $\lambda_{e,i}$ each block associated to (e, i) is a $\lambda_{e,i}$-block for.

Now consider a λ-block P_λ of P. We show that we can apply typical sequences in the usual way within P_λ, i.e. assume that whenever some vertices in $V(G) \setminus V(G_t)$ are attached in $\mathfrak{B}$ at P_λ, that they are attached at certain points of P_λ which can be distinguished even after using typical sequences for the $\mathcal{F}_{\equiv}$-bw to compress P_λ.

For a branch decomposition $\mathfrak{B}'$ of G_t and an edge $f \in E(\mathfrak{B}')$ we use $\mathcal{F}_{\equiv}\text{-bw}_\lambda(\mathfrak{B}', f)$ to denote the maximum size of an induced matching M in the cut of $\mathfrak{B}'$ at f in which the set of vertices in $N^3[\chi(t)]$ adjacent to edges in $M[N^3[\chi(t)]]$ is equal to λ. In the pathological case that there is no such matching in which the set of vertices in $N^3[\chi(t)]$ adjacent to edges in $M[N^3[\chi(t)]]$ is equal to λ, we set $\mathcal{F}_{\equiv}\text{-bw}_\lambda(\mathfrak{B}', f) = \perp$. Finally for two edges $p, q \in E(P_\lambda)$, we write $P_\lambda(p, q)$ to denote the subpath of P_λ between p and q and excluding p and q. We do not make any distinction between $P_\lambda(p, q)$ and $P_\lambda(q, p)$.

▶ **Lemma 5.7.** *Let $\mathfrak{B}' = \text{restr}_{\mathfrak{B}}(N^3[V(G_t)])$ be the branch decomposition of $G[N^3[V(G_t)]]$ given by $\mathfrak{B}$, and let P' be the subpath of $\mathfrak{B}'$ that corresponds to P_λ. Assume that there are $p, q \in E(P')$ such that all of the following hold.*

1. $\mathcal{F}_{\equiv}\text{-bw}_\lambda(\mathfrak{B}', p) = \min\{\mathcal{F}_{\equiv}\text{-bw}_\lambda(\mathfrak{B}', r) \mid r \in E(P'(p, q))\}$.
2. $\mathcal{F}_{\equiv}\text{-bw}_\lambda(\mathfrak{B}', q) = \max\{\mathcal{F}_{\equiv}\text{-bw}_\lambda(\mathfrak{B}', r) \mid r \in E(P'(p, q))\}$.
3. *For every subtree $\mathfrak{B}_v$ of $\mathfrak{B}$ which is attached at an internal node v of P_λ, it holds that $V(\mathfrak{B}'_v) \cap N^3[\chi(t)] = \emptyset$.*

Obtain $\mathfrak{B}^{**}$ from $\mathfrak{B}$ by removing for each internal node v of $P_\lambda(p, q)$ $V(G) \setminus V(G_t)$ from $\mathfrak{B}$ and attaching $F_v = \text{restr}_{\mathfrak{B}_v}(V(G) \setminus V(G_t))$ in the order of the considered vertices v along P_λ at the iterative subdivision of p. See Figure 1 for an illustration.

Then $\mathcal{F}_{\equiv}\text{-bw}(\mathfrak{B}^{**}) \leq \mathcal{F}_{\equiv}\text{-bw}(\mathfrak{B})$, $\mathcal{F}_{\bowtie}\text{-bw}(\mathfrak{B}^{**}) \leq \mathcal{F}_{\bowtie}\text{-bw}(\mathfrak{B})$, and $\mathcal{F}_{\lesseqgtr}\text{-bw}(\mathfrak{B}^{**}) \leq \mathcal{F}_{\lesseqgtr}\text{-bw}(\mathfrak{B})$.

In the later application of Lemma 5.7, $P'(p, q)$ takes the role of a path that is contracted in for obtaining any of the typical sequences in a record, and iteratively applying this lemma ensures the “safeness” (w.r.t. $\mathcal{F}^*$-branchwidth and the properties of blocks) of using them.

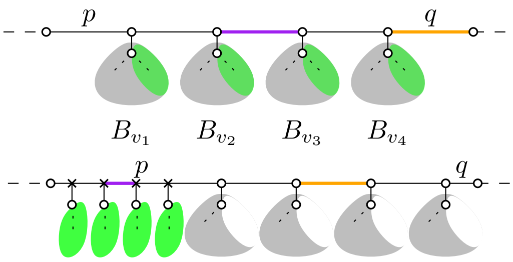

Figure 1 Illustration of modification of $\mathfrak{B}$ (top) to $\mathfrak{B}^{**}$ (bottom) as described in Lemma 5.7. The cross-vertices in the bottom figure are the subdivision vertices used to attach the $\mathrm{restr}_{\mathfrak{B}_v}(V(G) \setminus V(G_t))$ to p. The cutfunction values at the orange and purple edge in $\mathfrak{B}^{**}$ are upper-bounded by the cutfunction values at the orange and purple edge in $\mathfrak{B}^{*}$ respectively.

Dynamic Programming Procedure

With the machinery now in place we are able to describe the dynamic programming procedure which we use to prove Theorem 5.1.

We traverse T in leaves-to-root order and compute a set $\mathcal{R}(t)$ of records together with corresponding *canonical* branch decompositions $\mathfrak{B}$ of G_t for each node $t \in V(T)$. Intuitively the information in the records allows us to iteratively extend the associated branch decompositions while bounding $\mathcal{F}_{\divideontimes}$-bw and $\mathcal{F}_{\lessgtr}$-bw by $\alpha^{\divideontimes}$ and $\alpha^{\lessgtr}$ respectively. For $\mathcal{F}_{\equiv}$-bw we face a slight caveat. During the dynamic programming we are not able to keep track of the actual $\mathcal{F}_{\equiv}$-bw without keeping a significant amount of information about forgotten vertices, which we cannot allow ourselves without exceeding any bound in our parameters. Instead we store a bound $\alpha^{\equiv}$ on the $\mathcal{F}_{\equiv}$-bw of the constructed branch decompositions *assuming that in the records by which a certain record is reached each* $\sigma_{e,i}$ *corresponds to a* $\lambda_i \in \Lambda_e$*-block which is consistent with the choices made for blocks at lower levels of* T. Checking whether this assumption is actually consistent with the construction in the dynamic programming procedure is the reason we include the canonical branch decompositions, some edges of which have "*pointers*" to elements of sequences in σ for the respective record.

Keeping this in mind, we can formulate four procedures by which we compute $\mathcal{R}(t)$ depending on the type of t assuming $\mathcal{F}$ contains all of $\mathcal{F}_{\equiv}$, $\mathcal{F}_{\divideontimes}$ and $\mathcal{F}_{\lessgtr}$. It can easily be seen that whenever $\mathcal{F}_{\equiv}$ is not contained in $\mathcal{F}$, the entries $\flat$, Λ and $\alpha^{\equiv}$ as well as the canonical branch decompositions for records can be safely omitted in each step, as the computation of the other entries do not depend on them. Analogously whenever $\mathcal{F}_{\divideontimes}$ is not contained in $\mathcal{F}$, $\alpha^{\divideontimes}$ can safely be omitted from all computation steps, and whenever $\mathcal{F}_{\lessgtr}$ is not contained in $\mathcal{F}$, $\alpha^{\lessgtr}$ can safely be omitted from all computation steps. All of the following constructions are easily seen to be executable in FPT time parameterized by $\mathrm{tw}(G)$ and $\Delta(G)$ using Facts 2.8, 2.9 and 2.11.

At the root r of T, assume we have correctly computed $\mathcal{R}(r)$. Now go through the elements of $\mathcal{R}(r)$ in ascending order of $\max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lessgtr}\}$ for the respective records. For each element, consider its branch decomposition and compute its $\mathcal{F}_{\equiv}$-bw. If it is equal to $\alpha^{\equiv}$ then we output $\max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lessgtr}\}$ and the associated branch decomposition witnesses that $\mathcal{F}^{*}$-bw$(G) = \max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lessgtr}\}$.

Correctness

To argue correctness, we first we show that the claimed $\mathcal{F}^{*}$-branchwidth given as the output of the algorithm actually can be achieved by a branch decomposition of G, in particular this is true for the canonical branch decomposition $\mathfrak{B}$ associated to the record $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lessgtr})$

that leads to the output. Because we ensure that $\mathcal{F}_{\equiv}$-bw$(\mathfrak{B}) = \alpha^{\equiv}$, we obtain immediately that $\mathcal{F}^*$-bw$(\mathfrak{B}) \leq \max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr}\}$. Assume for contradiction that $\mathcal{F}^*$-bw$(\mathfrak{B}) > \max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr}\}$. Then $\mathcal{F}_{\divideontimes}$-bw$(\mathfrak{B}) > \max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr}\}$ or $\mathcal{F}_{\lesseqgtr}$-bw$(\mathfrak{B}) > \max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr}\}$. Let $v \in V(G)$ be in a subgraph H of G which witnesses $\mathcal{F}_{\divideontimes}$-bw$(\mathfrak{B}) > \max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr}\}$ or $\mathcal{F}_{\lesseqgtr}$-bw$(\mathfrak{B}) > \max\{\alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr}\}$. Then because of the small diameter of antimatchings and chain graphs we have that $V(H) \subseteq N^3[v]$. Consider $t \in V(T)$ in which v is introduced, or the leaf node with $\chi(t) = \{v\}$. By construction there must be a record $(D', \flat', \Lambda', \sigma', \alpha^{\equiv'}, \alpha^{\divideontimes'}, \alpha^{\lesseqgtr'})$ together with an associated branch decomposition $\mathfrak{B}'$ such that $\mathrm{restr}_{\mathfrak{B}}(V(G_t)) = \mathfrak{B}'$, and in particular $\mathrm{restr}_{\mathfrak{B}}(N^3[v])$ is a subtree of D'. This means that $\alpha^{\divideontimes'} \geq \mathcal{F}_{\divideontimes}$-bw$(D') > \alpha^{\divideontimes}$ or $\alpha^{\lesseqgtr'} \geq \mathcal{F}_{\lesseqgtr}$-bw$(D') > \alpha^{\lesseqgtr}$. However, the iterative construction ensures that the entries for $\alpha^{\divideontimes''}$ and $\alpha^{\lesseqgtr''}$ can not decrease for records $(D'', \Lambda'', \sigma'', \alpha^{\equiv''}, \alpha^{\divideontimes''}, \alpha^{\lesseqgtr''})$ encountered when constructing $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr})$ from $(D', \Lambda', \sigma', \alpha^{\equiv'}, \alpha^{\divideontimes'}, \alpha^{\lesseqgtr'})$. Thus $\alpha^{\divideontimes} \geq \alpha^{\divideontimes'}$ and $\alpha^{\lesseqgtr} \geq \alpha^{\lesseqgtr'}$, yielding a contradiction.

Conversely, now we show that the $\mathcal{F}^*$-branchwidth returned by the algorithm is at most as large as the $\mathcal{F}^*$-branchwidth of any branch decomposition of G, and in particular this implies that the algorithm always returns a solution (which is not immediately clear from the description of the root step). Assume for contradiction that there is a node $t \in V(T)$ such that there is no record $(D, \flat, \Lambda, \sigma, \alpha^{\equiv}, \alpha^{\divideontimes}, \alpha^{\lesseqgtr})$ with canonical branch decomposition $\mathfrak{B}$ of G_t in $\mathcal{R}(t)$ such that the following two conditions hold.

- $\mathfrak{B}$ can be extended to a branch decomposition $\mathfrak{B}^*$ of G realizing the $\mathcal{F}^*$-branchwidth of G.
- Every path in $\mathfrak{B}^*$ that corresponds to an edge $e \in E(D)$ consists of a sequence of blocks $P_1, \ldots, P_{|\flat_e|+1}$, specifically P_i is a $\lambda_{e,i}$-block. Further each typical sequence of $(\mathcal{F}_{\equiv}\text{-bw}_{\lambda_{e,i}}(\mathfrak{B}, e))_{e \in E(P_i')}$ where P_i' is the path in $\mathfrak{B}$ which P_i corresponds to is equal to $\sigma_{e,i}$.

Let t be such a node such that for all nodes $t' \in V(T_t)$ there is a record $(D', \Lambda', \sigma', \alpha^{\equiv'}, \alpha^{\divideontimes'}, \alpha^{\lesseqgtr'})$ with corresponding branch decomposition $\mathfrak{B}'$ of $G_{t'}$ in $\mathcal{R}(t')$ such that the following two conditions hold:

- $\mathfrak{B}'$ can be extended to the same branch decomposition $\mathfrak{B}^*$ of G realizing the $\mathcal{F}^*$-branchwidth of G.
- Every path in $\mathfrak{B}^*$ that corresponds to an edge $e \in E(D')$ consists of a sequence of blocks $P_1, \ldots, P_{|\flat_e'|+1}$, specifically P_i is a $\lambda_{e,i}'$-block. Further each typical sequence of $(\mathcal{F}_{\equiv}\text{-bw}_{\lambda_{e,i}'}(\mathfrak{B}', e))_{e \in E(P_i')}$ where P_i' is the path in $\mathfrak{B}'$ which P_i corresponds to is equal to $\sigma_{e,i}'$.

We can distinguish the type of t and arrive at a contradiction in every case.

6 Treedepth

In this section we give a fixed parameter tractable algorithm for $\mathcal{F}^*$-Branchwidth parameterized by the treedepth of the input graph, when $\mathcal{F}^*$ is the union of a non-empty set of some ***primal ph*** families. We begin by outlining the high-level idea.

If a graph has small treedepth but sufficiently many vertices, then one can always find a small vertex set R such that $G - R$ has many components where each of them has bounded size. The main step is to argue that in this case, we can safely remove ("prune") one of the components without changing its $\mathcal{F}^*$-branchwidth. Proposition 6.1 captures the idea of this procedure, and we inductively use this result from bottom to top in the treedepth decomposition. At the end, we reduce the given graph to a graph whose number of vertices is bounded by a function of the treedepth of the given graph, i.e., a (non-polynomial) kernel. The problem can then be solved on such a kernel using an arbitrary brute-force algorithm.

▶ **Proposition 6.1.** *There is a function $g\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ satisfying the following. Let t, p, and m be positive integers, and let $\mathcal{F}^*$ be the union of a non-empty set of `primal ph` classes. Let G be a graph, let R be a vertex set of size at most t, let H be a graph on p vertices, and let $\{G_i \mid i\in[m]\}$ be a set of components of $G-R$ such that*

- *for each $i\in[m]$, there is a graph isomorphism ϕ_i from H to G_i,*
- *for all $i,j\in[m]$ and $h\in V(H)$ and $v\in R$, $\phi_i(h)$ is adjacent to v if and only if $\phi_j(h)$ is adjacent to v.*

If $m\geq g(t,p)$ and $\mathcal{F}^*\text{-bw}(G)\leq t$, then $\mathcal{F}^*\text{-bw}(G)=\mathcal{F}^*\text{-bw}(G-V(G_m))$.

We explain the idea to prove Proposition 6.1. Suppose we have G, R, H, and the family $\{G_i \mid i\in[m]\}$ as in the statement. Let $(\mathfrak{B},\mathcal{L})$ be a branch decomposition of $G-V(G_m)$ of optimal width, and we want to find a branch decomposition of G having same width. Let $V(H)=\{h_1,h_2,\dots,h_p\}$. First we take nodes corresponding to R and its least common ancestors. Let S be the union of all least common ancestors in the tree. We take a large subset I_1 of $[m]$ satisfying that for all $a\in[p]$ and $i_1,i_2\in I_1$, $\phi_{i_1}(h_a)$ and $\phi_{i_2}(h_a)$ are contained in the same component of $\mathfrak{B}-S$.

As a second step, we show that there are a large subset $I_2\subseteq I_1$, a set F of at most $\ell-1$ edges in $\mathfrak{B}$ for some $1\leq\ell\leq p$, a partition $(A_1,\dots,A_\ell)$ of $[p]$, and a bijection μ from $[\ell]$ to the set of connected components of $\mathfrak{B}-F$ such that for all $j\in[\ell]$ and $i_1,i_2\in I_2$, the minimal subtree of $\mathfrak{B}$ containing all nodes in $\{\mathcal{L}^{-1}(\phi_{i_1}(h_b)) \mid b\in A_j\}$ and the minimal subtree of $\mathfrak{B}$ containing all nodes in $\{\mathcal{L}^{-1}(\phi_{i_2}(h_b)) \mid b\in A_j\}$ are vertex-disjoint. In fact, by the choice of S and I_2, there should be a component of $\mathfrak{B}-S-F$ containing all nodes corresponding to $\{h_b \mid b\in A_j\}$. Then in each component of $\mathfrak{B}-S-F$ corresponding to A_j, we can find an edge separating the family of minimal subtrees corresponding to A_j in a balanced way, and expand that edge to some subtree corresponding to vertices of $\{\phi_m(h_b) \mid b\in A_j\}$. When we expand the edge, we take the restricted tree of $\mathfrak{B}$ with respect to some $\{\mathcal{L}^{-1}(\phi_\alpha(h_b)) \mid b\in A_j\}$ which is closest to the edge, and make a copy and identify with the edge in a natural way. Then we can prove that every new cut has small width as well.

To prove the main theorem of this section, the following definition is useful. Let T be a rooted tree with height k, and let r be the root of T. For every node t of T, we define the *rank* of t as $k-\text{dist}_T(t,r)$. Note that the rank of a leaf t with maximum $\text{dist}_T(t,r)$ has rank 1.

▶ **Theorem 6.2.** *Let $\mathcal{F}^*$ be the union of a non-empty set of `primal ph` classes. Then $\mathcal{F}^*$-Branchwidth is fixed-parameter tractable when parameterized by treedepth.*

Proof. Let G be a graph of treedepth k. By Lemma 4.14, we may assume that G is connected. Let T be a rooted forest with height $k=\text{td}(G)$ whose closure contains G as a subgraph. As G is connected, we may assume that T is a tree. Let r denote the root of T.

Let g be the function defined in Proposition 6.1. We define $h(1):=1$ and for $1<j\leq k$, we recursively define $h(j):=2^{\binom{h(j-1)}{2}}\cdot 2^{(k+j-1)h(j-1)}\cdot g(k+1,h(j-1))$.

For each $j\in[k]$, we recursively obtain a graph G^j and a rooted tree T^j whose closure contains G^j as a subgraph such that

- $\mathcal{F}^*\text{-bw}(G)=\mathcal{F}^*\text{-bw}(G^j)$,
- T^j has height k, and
- for every $i\in[j]$ and every node t in T^j of rank i, $G^j[V(G^j)\cap V((T^j)_t)]$ has at most $h(i)$ vertices.

We define $G^1:=G$ and $T^1:=T$. Clearly, G^1 and T^1 satisfy the above properties. If this holds, then G^k will have the size bounded by a function of k, and the problem can be solved on G^k using an arbitrary brute-force algorithm.

We assume that $1 < j \leq k$ and that G^{j-1} and T^{j-1} have been constructed, and we now want to construct G^j and T^j in polynomial time. For each node t of T^{j-1}, we denote $(G^{j-1})_t := G^{j-1}[V(G^{j-1}) \cap V((T^{j-1})_t)]$. We check whether $(G^{j-1})_t$ has at most $h(j)$ vertices for every node t of rank j. If this is true, then we can set $G^j := G^{j-1}$, $T^j := T^{j-1}$ and we are done. Thus, we may assume that there is a node t of rank j such that $(G^{j-1})_t$ has more than $h(j)$ vertices. Assume that $\{t_i \mid i \in [x]\}$ is the set of all children of t in T^{j-1}, and $R = \{r_i \mid i \in [k+j-1]\}$ is the vertex set of the path from the root r to t in T^{j-1}.

We consider a pair (H, γ) of a graph H on at most $h(j-1)$ vertices and a function $\gamma : V(H) \to 2^{V(R)}$. Up to isomorphism, there are at most $2^{\binom{h(j-1)}{2}} \cdot 2^{(k+j-1)h(j-1)}$ such pairs. Intuitively, we use this pair to reflect the information how the corresponding vertex in $(G^{j-1})_{t_i}$ is adjacent to a vertex in $V(R)$.

As each $(G^{j-1})_{t_i}$ has at most $h(j-1)$ vertices and $(G^{j-1})_t$ has more than $h(j)$ vertices, there is a subset $\{p_i \mid i \in [y]\}$ of $\{t_i \mid i \in [x]\}$ and a pair (H, γ) with $y = g(k+1, h(j-1))$ such that

- for every $i \in [y]$, there is a graph isomorphism ϕ_i from H to $(G^{j-1})_{p_i}$,
- for every $i \in [y]$ and $z \in [k-j+1]$ and $v \in V(H)$, $\phi_i(v)$ is adjacent to r_z if and only if $r_z \in \gamma(v)$.

Note that by Corollary 4.6, we have $\mathcal{F}^*\text{-bw}(G^{j-1}) = \mathcal{F}^*\text{-bw}(G) \leq \text{tw}(G) + 1 \leq \text{td}(G) + 1 \leq k + 1$. As $y = g(k+1, h(j-1))$, by Proposition 6.1, we know that $\mathcal{F}^*\text{-bw}(G^{j-1}) = \mathcal{F}^*\text{-bw}(G^{j-1} - V((G^{j-1})_{p_y}))$. Thus, we can safely reduce the graph by removing $V((G^{j-1})_{p_y})$. After applying this procedure recursively, we will obtain a graph G^j and a rooted tree T^j such that for every node t in T^j of rank j, $G^j[V(G^j) \cap V((T^j)_t)]$ has at most $h(j)$ vertices.

This concludes the theorem. ◀

▶ **Corollary 6.3.** *Mim-Width is fixed-parameter tractable parameterized by the treedepth of the input graph.*

7 Feedback Edge Set

In this section we describe a linear kernel for $\mathcal{F}^*$-Branchwidth parameterized by the size k of a feedback edge set of the input graph, when $\mathcal{F}^*$ is the union of some *primal ph* families.

If a graph G has many vertices but a small feedback edge set, then either it has many bridges (in form of "dangling trees") and isolated vertices, or it has long paths of degree two vertices in G, which we refer to as "unimportant paths". This simple observation paves the road to the kernel: For the former, we can observe that bridges and isolated vertices are not important when it comes to computing $\mathcal{F}^*$-branchwidth. For the latter, we show that we can always shrink any unimportant path to a constant-length subpath without changing the $\mathcal{F}^*$-branchwidth of G. It is not surprising that this gives a safe reduction rule when $\mathcal{F}^*$ does not contain induced matchings. For induced matchings however, proving safeness requires quite an amount of detail. For this we show Lemma 7.3 which states that if G has a long enough unimportant path P, then we can modify each branch decomposition of G such that the vertices of a long enough subpath of P appear "consecutively" in it, without increasing the $\mathcal{F}^*$-branchwidth. In addition to that, we only have to overcome one minor hurdle that concerns induced chains of value 2. From there on it is not difficult to argue that contracting an edge on P does not change the $\mathcal{F}^*$-branchwidth either.

Let us give some details. We first define unimportant paths, and what it means for the vertices of a path to appear "consecutively" in a branch decomposition via the notion of preservation.

▶ **Definition 7.1.** *Let G be a graph. A path $P \subseteq G$ is called* unimportant *if all vertices on P have degree two in G.*

▶ **Definition 7.2.** *Let G be a graph, let $(\mathfrak{B}, \mathcal{L})$ a branch decomposition of G, and let $P = a_1 \dots a_\ell \subseteq G$ be a path. We say that $(\mathfrak{B}, \mathcal{L})$* preserves *$P$ if there is a caterpillar $\mathfrak{B}''$ in $\mathfrak{B}$ such that the linear order induced by $\mathfrak{B}''$ (up to reversal) is $a_1 \dots a_\ell$.*

We now turn to the main technical lemma needed in the correctness proof of the reduction rule alluded to above. Note that it only concerns induced matchings.

▶ **Lemma 7.3.** *Let G be a graph with branch decomposition $(\mathfrak{B}, \mathcal{L})$. Suppose that G contains an unimportant path P of length 7. Then there is a branch decomposition $(\mathfrak{B}', \mathcal{L}')$ of G such that*

1. $\mathcal{F}_{\equiv}\text{-bw}(\mathfrak{B}', \mathcal{L}') \le \mathcal{F}_{\equiv}\text{-bw}(\mathfrak{B}, \mathcal{L})$*,*
2. *$(\mathfrak{B}', \mathcal{L}')$ preserves a subpath $P^\star$ of P on at least five vertices, and*
3. *the set of cuts induced by $(\mathfrak{B}, \mathcal{L})$ on $G - V(P^\star)$ is equal to the set of cuts induced by $(\mathfrak{B}', \mathcal{L}')$ on $G - V(P^\star)$.*

The construction in the previous lemma does not increase the $\mathcal{F}_{\equiv}$-branchwidth of a branch decomposition. It might however introduce induced chain graphs of value 2 in some cut, which is a technical complication that can, luckily, be dealt with by considering the case of $\mathcal{F}^*$-branchwidth 1 separately. Next, we can develop separate arguments that are designed to handle the case of $\mathcal{F}_{\lesseqgtr}$-branchwidth when $\mathcal{F}^*$-branchwidth is at least 2, and show that these two approaches can be combined together to deal with $\mathcal{F}^*$-branchwidth for any union $\mathcal{F}^*$ of ***primal ph*** classes. Recall that a graph is *bridgeless* if there is no edge $e \in E(G)$ such that $G - e$ has more connected components than G; we obtain:

▶ **Lemma 7.4.** *Let $\mathcal{F}^*$ be a union of **primal ph** classes. Let G be a bridgeless graph without isolated vertices with feedback edge set number k and an unimportant path P of length at least 8. Let $e \in E(P)$. Either $|V(G)| \le 8k - 3$, or $\mathcal{F}^*\text{-bw}(G) = \mathcal{F}^*\text{-bw}(G/e)$, or both.*

As the previous lemma requires that our input graph does not have any bridges, we need one more simple lemma to show that we can safely remove bridges from the input graph without changing its $\mathcal{F}^*$-branchwidth.

▶ **Reduction Rule 1.** *Let (G, k) be an instance of $\mathcal{F}^*$-Branchwidth, where G has a feedback edge set of size k. Then, reduce (G, k) to (G', k) where G' is obtained from G by*

- *removing all bridges of G, and then*
- *removing all isolated vertices of G.*

▶ **Reduction Rule 2.** *Let (G, k) be an instance of $\mathcal{F}^*$-Branchwidth, where G has a feedback edge set of size k. Let $P \subseteq G$ be any unimportant path in G of length at least 8 and let $e \in E(P)$. Then, reduce (G, k) to $(G/e, k)$.*

We now have all the tools necessary to describe the kernelization algorithm: We first apply Reduction Rule 1 to clean up the graph. Next, while the input graph has more than $18k - 8$ vertices we find a long enough unimportant path and apply Reduction Rule 2 to it. Together this yields:

▶ **Theorem 7.5.** *Let $\mathcal{F}^*$ be the union of some **primal ph** families. $\mathcal{F}^*$-Branchwidth admits a linear kernel when parameterized by the feedback edge set number of the input graph.*

▶ **Corollary 7.6.** *Mim-Width parameterized by the feedback edge set number of the input graph admits a linear kernel.*

8 Concluding Remarks

While our introduction focused predominantly on $\mathcal{F}$-branchwidth acting as a unifying framework for treewidth, clique-width and mim-width, the concept is in fact significantly more powerful than that; for instance, it is easy to see that the treewidth of the complement of the graph [18] is also captured by $\mathcal{F}$-branchwidth (one can simply take the complements of the classes used in Corollary 4.6). On the other hand, one immediate question arising from this work concerns the newly identified combinations of *si ph* classes: could, e.g., $\mathcal{F}_{\leqq}$- and $\mathcal{F}_{\divideontimes}$-branchwidth be natural counterparts to mim-width?

On the algorithmic side, one fundamental limitation one needs to keep in mind is that computing $\mathcal{F}$-branchwidth parameterized by the measure itself is W[1]-hard [31]. Hence we cannot hope for a fixed-parameter framework that would first compute an optimal $\mathcal{F}$-branch decomposition, and then proceed it to solve a problem of interest. In this sense, the question tackled by our algorithmic contribution is primarily a conceptual one: when can we compute *all* of the captured width parameters via a unified framework? The most obvious question that remains open in this regard is whether Theorem 5.1 can be generalized to the parameterization by treewidth alone.

A separate question that is no less important is whether we can compute approximately-optimal decompositions for $\mathcal{F}$-branchwidth in polynomial time when the width is bounded by a constant; solving this problem already for mim-width alone would be considered a major breakthrough in the field.

References

1 Rémy Belmonte and Martin Vatshelle. Graph classes with structured neighborhoods and algorithmic applications. *Theor. Comput. Sci.*, 511:54–65, 2013. doi:10.1016/j.tcs.2013.01.011.

2 Benjamin Bergougnoux and Mamadou Moustapha Kanté. More applications of the d-neighbor equivalence: Acyclicity and connectivity constraints. *SIAM J. Discret. Math.*, 35(3):1881–1926, 2021. doi:10.1137/20M1350571.

3 Nadja Betzler, Robert Bredereck, Rolf Niedermeier, and Johannes Uhlmann. On bounded-degree vertex deletion parameterized by treewidth. *Discret. Appl. Math.*, 160(1-2):53–60, 2012.

4 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A c^k n 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.

5 Hans L. Bodlaender, Lars Jaffke, and Jan Arne Telle. Typical sequences revisited - computing width parameters of graphs. In Christophe Paul and Markus Bläser, editors, *Proceedings of the 37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020*, volume 154 of *LIPIcs*, pages 57:1–57:16. Schloss Dagstuhl, 2020. doi:10.4230/LIPIcs.STACS.2020.57.

6 Hans L. Bodlaender and Ton Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms*, 21(2):358–402, 1996.

7 Binh-Minh Bui-Xuan, Ondrej Suchý, Jan Arne Telle, and Martin Vatshelle. Feedback vertex set on graphs of low clique-width. *Eur. J. Comb.*, 34(3):666–679, 2013.

8 Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. *H*-join decomposable graphs and algorithms with runtime single exponential in rankwidth. *Discret. Appl. Math.*, 158(7):809–819, 2010.

9 Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theor. Comput. Sci.*, 511:66–76, 2013. doi:10.1016/j.tcs.2013.01.009.

10 B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015. doi:10.1007/978-3-319-21275-3.

12 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate texts in mathematics.* Springer, 4th edition, 2012.

13 Guoli Ding, Bogdan Oporowski, James G. Oxley, and Dirk Vertigan. Unavoidable minors of large 3-connected binary matroids. *J. Comb. Theory, Ser. B*, 66(2):334–360, 1996.

14 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

15 Paul Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio mathematica*, 2:463–470, 1935.

16 Robert Ganian and Sebastian Ordyniak. The complexity landscape of decompositional parameters for ILP. *Artif. Intell.*, 257:61–71, 2018.

17 Robert Ganian and Sebastian Ordyniak. The power of cut-based parameters for computing edge-disjoint paths. *Algorithmica*, 2021. To appear.

18 Guilherme de C. M. Gomes, Carlos V. G. C. Lima, and Vinícius Fernandes dos Santos. Parameterized complexity of equitable coloring. *Discret. Math. Theor. Comput. Sci.*, 21(1), 2019.

19 Ronald L. Graham, Bruce L. Rothschild, and Joel H. Spencer. *Ramsey theory*, volume 20. John Wiley & Sons, 1990.

20 Gregory Z. Gutin, Mark Jones, and Magnus Wahlström. The mixed chinese postman problem parameterized by pathwidth and treedepth. *SIAM J. Discret. Math.*, 30(4):2177–2205, 2016.

21 Thekla Hamm. Finding linear arrangements of hypergraphs with bounded cutwidth in linear time. In Bart M. P. Jansen and Jan Arne Telle, editors, *Proceedings of the 14th International Symposium on Parameterized and Exact Computation, IPEC 2019*, volume 148 of *LIPIcs*, pages 20:1–20:14. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.IPEC.2019.20.

22 Svein Høgemo, Jan Arne Telle, and Erlend Raa Vågset. Linear mim-width of trees. In Ignasi Sau and Dimitrios M. Thilikos, editors, *Proceedings of the 45th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2019*, volume 11789 of *LNCS*, pages 218–231. Springer, 2019.

23 Lars Jaffke, O-joung Kwon, Torstein J. F. Strømme, and Jan Arne Telle. Mim-width III. Graph powers and generalized distance domination problems. *Theor. Comput. Sci.*, 796:216–236, 2019. doi:10.1016/j.tcs.2019.09.012.

24 Lars Jaffke, O-joung Kwon, and Jan Arne Telle. Mim-width I. Induced path problems. *Discret. Appl. Math.*, 278:153–168, 2020.

25 Jisu Jeong, Sigve Hortemo Sæther, and Jan Arne Telle. Maximum matching width: New characterizations and a fast algorithm for dominating set. *Discret. Appl. Math.*, 248:114–124, 2018.

26 Ton Kloks. *Treewidth: Computations and Approximations.* Springer, Berlin, 1994.

27 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and Combinatorics.* Springer, 2012.

28 Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B*, 96(4):514–528, 2006.

29 Michaël Rao. Clique-width of graphs defined by one-vertex extensions. *Discret. Math.*, 308(24):6157–6165, 2008.

30 Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

31 Sigve Hortemo Sæther and Martin Vatshelle. Hardness of computing width parameters based on branch decompositions over the vertex set. *Theor. Comput. Sci.*, 615:120–125, 2016.

32 Dimitrios M. Thilikos, Maria J. Serna, and Hans L. Bodlaender. Cutwidth II: algorithms for partial w-trees of bounded degree. *J. Algorithms*, 56(1):25–49, 2005. doi:10.1016/j.jalgor.2004.12.003.

33 Martin Vatshelle. *New width parameters of graphs*. PhD thesis, University of Bergen, 2012.

34 Koichi Yamazaki. Inapproximability of rank, clique, boolean, and maximum induced matching-widths under small set expansion hypothesis. *Algorithms*, 11(11):173, 2018.