Twin-width VI: the lens of contraction sequences

Édouard Bonnet
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Eun Jung Kim
Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Amadeus Reinald
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Stéphan Thomassé
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Abstract

A contraction sequence of a graph consists of iteratively merging two of its vertices until only one vertex remains. The recently introduced twin-width graph invariant is based on contraction sequences. More precisely, if one puts error edges, henceforth red edges, between two vertices representing non-homogeneous subsets, the twin-width is the minimum integer d such that a contraction sequence exists that keeps the red degree at value at most d. By changing the condition imposed on the trigraphs (i.e., graphs with some edges being red) and possibly slightly tweaking the notion of contractions, we show how to characterize the well-established bounded rank-width, tree-width, linear rank-width, path-width, and monotone proper minor-closed classes by means of contraction sequences.

Thus going from parameters based on branch-decompositions to parameters based on contraction sequences has a crucial advantage: while we can still express classical width notions, we can go through the planar barrier which is captured by bounded twin-width. As another application we give a transparent alternative proof of the celebrated Courcelle's theorem (actually of its generalization by Courcelle, Makowsky, and Rotics), that MSO_2 (resp. MSO_1) model checking on graphs with bounded tree-width (resp. bounded rank-width) is fixed-parameter tractable in the size of the input sentence. We are hopeful that our characterizations can help in other contexts.

We then explore new avenues along the general theme of contraction sequences both in order to refine the landscape between bounded tree-width and bounded twin-width (via spanning twin-width) and to capture more general classes than bounded twin-width. To this end, we define an oriented version of twin-width, where appearing red edges are oriented away from the newly contracted vertex, and the mere red out-degree should remain bounded. Surprisingly, classes of bounded oriented twin-width coincide with those of bounded twin-width. This greatly simplifies the task of showing that a class has bounded twin-width. As an example, using a lemma by Norine, Seymour, Thomas, and Wollan, we give a 5-line proof that K_t-minor free graphs have bounded twin-width. Without oriented twin-width, this fact was shown by a somewhat intricate 4-page proof in the first paper of the series. Finally we explore the concept of partial contraction sequences, instead of terminating on a single-vertex graph, the sequence ends when reaching a particular target class. We show that FO model checking (resp. ∃FO model checking) is fixed-parameter tractable on classes with partial contraction sequences to a class of bounded degree (resp. bounded expansion), provided the partial contraction sequence is given. Efficiently finding such partial sequences could turn out simpler than finding a (complete) sequence.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Fixed parameter tractability

Keywords and phrases Twin-width, contraction sequences, width parameters, FO and MSO model checking, matroids

1 Introduction

A *trigraph* is a graph with some of its edges being distinguished, typically called *red edges*, while the rest of the edges are called *black*. The (vertex) *contraction* (or *identification*) of two non-necessarily adjacent vertices u and v in a trigraph consists of merging these two vertices into a new vertex w, keeping every edge wx black if both ux and vx were black edges, and turning all the other edges incident to w red. The rest of the trigraph does not change. A *contraction sequence* of an n-vertex (tri)graph G is a sequence of trigraphs $G = G_n, G_{n-1}, \ldots, G_1$ such that G_i is an i-vertex trigraph, obtained by performing one contraction in G_{i+1}. A *d-sequence* is a contraction sequence such that every trigraph of the sequence has maximum red degree at most d. The *twin-width* of a graph is defined via contraction sequences: It is the minimum integer d such that G admits a d-sequence. See Figure 1 for an example of a graph with a 2-sequence. Not to hinder the flow of this

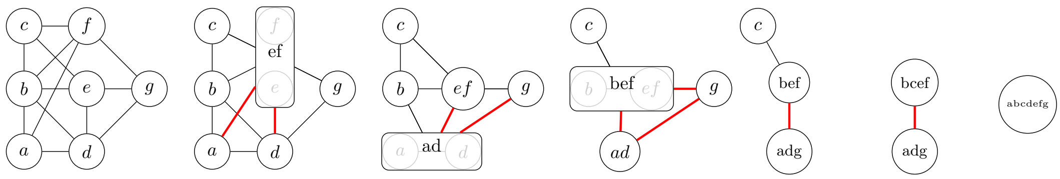


Figure 1 A 2-sequence witnessing that the initial graph has twin-width at most 2.

introduction, we will try to limit further definitions. The reader is deferred to Section 2 if encountering some unknown terminology.[1]

Classes of bounded twin-width are surprisingly diverse. They include for instance classes of bounded tree-width, or even bounded rank-width, proper minor-closed classes, hereditary proper subclasses of permutation graphs, subgraphs of $O(1)$-dimensional grids [6], as well as $\Omega(\log n)$-subdivisions of n-vertex graphs, classes with bounded queue or stack number, and some families of expanders [3]. Nevertheless classes of bounded twin-with have interesting properties: They are closed under taking first-order (FO) transductions [6], χ-bounded [4], and allow a fixed-parameter tractable (FPT) algorithm for FO model checking [6], provided $O(1)$-sequences are given in input, and more practical FPT algorithms on specific problems like k-Independent Set or k-Dominating Set [4]. Twin-width naturally extends to matrices over finite alphabets and binary structures in general [6, 7]. Efficiently approximating twin-width (that is, returning an $f(d)$-sequence when the twin-width of the input is at most d) can be done for classes of *totally ordered* binary structures [5], but remains an open challenge for *unordered* graphs.

This paper investigates variations on the theme of contraction sequences. We first show that bounded rank-width and bounded linear rank-width can be defined by means of contractions sequences. Instead of requiring the maximum red degree to be bounded, one shall strengthen the condition to bounded-size red components, and bounded total number of red edges,[2] respectively. In the sparse regime of biclique-free classes, this characterizes bounded tree-width and bounded path-width. See Figures 2c and 2d for some illustration.

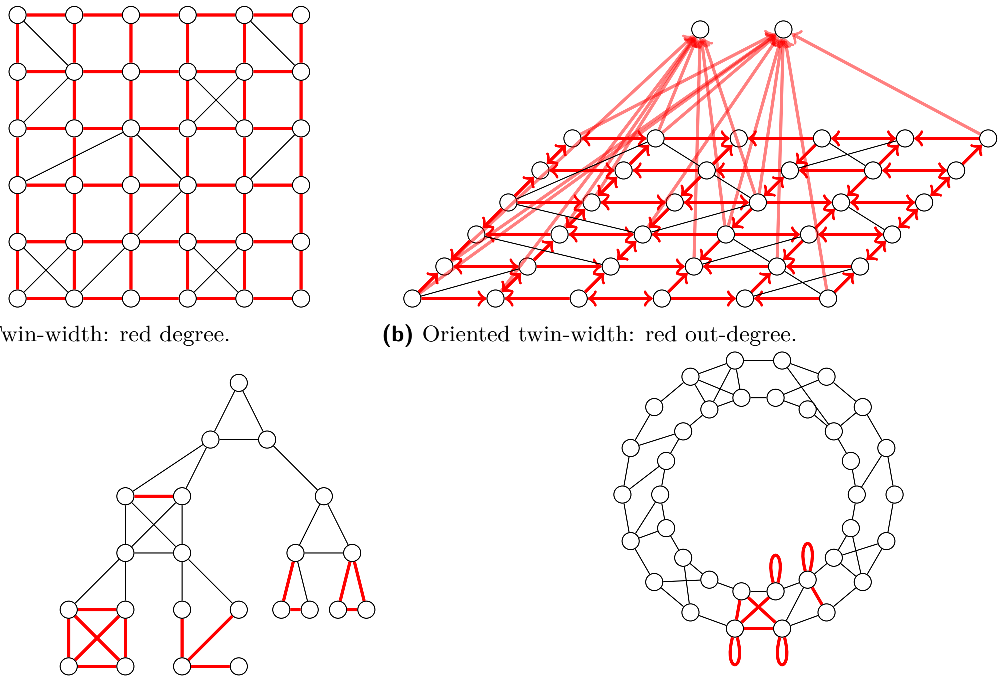
Let us elaborate on that with an alternative formalism. A useful equivalent viewpoint on contraction sequences is the notion of *partition sequence*, that is, a sequence $\mathcal{P}_n, \ldots, \mathcal{P}_1$ of

[1] If a term is still not defined there, it is not important for the rest of the paper.
[2] modulo the technicality of introducing red loops (see Section 2)

partitions of the vertex set $V(G)$, where $\mathcal{P}_n$ is the partition into singletons, $\mathcal{P}_1 = \{V(G)\}$, and each $\mathcal{P}_i$ is obtained by merging two parts of $\mathcal{P}_{i+1}$. Then a *width* $w : \mathcal{P}(V(G)) \to \mathbb{N}$ (i.e., a function from the partitions of $V(G)$ to the naturals) can be naturally lifted to the graph G as the minimum integer t such that there exists a partition sequence $\mathcal{P}_n, \ldots, \mathcal{P}_1$ satisfying $w(\mathcal{P}_i) \leqslant t$ for every i. This defines the *width of G associated to w*.

What are partitions of "good quality" or *small width*? Probably partitions $\mathcal{P}_i$ such that the quotient $G/\mathcal{P}_i$ "captures" the edge set of G quite well, minimizing the extra amount of information one needs to fully recover G. In that respect, the ideal scenario is when a pair of parts $X, Y \in \mathcal{P}_i$ is *homogeneous*, that is, X and Y are completely adjacent or completely non-adjacent in G. Consider the auxiliary graph with vertex set the parts of $\mathcal{P}_i$, and edge set all pairs of parts $X, Y \in \mathcal{P}_i$ which are *not* homogeneous. Note that this auxiliary graph is precisely the *red graph* of trigraph G_i (as previously defined), that is, obtained from G_i by keeping the red edges only. Three width measures come relatively naturally: the maximum degree, w_d, the maximum component-size, w_c, and the total number of edges, w_t, in the red graph of G_i. We obtain the following invariant on graphs.

- The *twin-width* of G as the width $\text{tww}(G)$ associated to w_d (see Figure 2a).
- The *component twin-width* of G as the width $\text{ltww}(G)$ associated to w_c (see Figure 2c).
- The *total twin-width* of G as the width $\text{ttww}(G)$ associated to w_t (see Figure 2d).



(a) Twin-width: red degree.
(b) Oriented twin-width: red out-degree.
(c) Component twin-width: size of red components.
(d) Total twin-width: number of red edges.

Figure 2 Examples of graphs with low value of the four corresponding twin-width variants.

As we will see in Section 3, the striking fact is that the latter two parameters already exist, up to functional equivalence.

▶ Theorem 1. *The following parameters are functionally equivalent:*
- *Component twin-width and rank-width;*
- *Total twin-width and linear rank-width.*

This phrases the classic width measures (tree-width, rank-width, path-width, linear rank-width) in the language of twin-width and contraction sequences. This unifying lens has two main benefits.

The first benefit is simplicity and renewal. We propose some examples where our characterizations somewhat simplify matters or bring a new, slightly but resolutely different perspective. We give a short alternative proof of the celebrated theorem by Courcelle, Makowsky, and Rotics [11] that monadic second-order (MSO) model checking (with adjacency relation only) is fixed-parameter tractable on classes with bounded rank-width. More precisely, $G \models \varphi$ is decidable in time $f(|\varphi|, \mathrm{cw})|V(G)|$ on graphs G given with a clique-width expression with cw labels, and MSO sentences φ. This is known to generalize Courcelle's theorem that MSO model checking with incidence relation is fixed-parameter tractable on classes with bounded tree-width.

Let us sketch how our algorithm goes. Instead of parsing a clique-width expression (or a tree-decomposition), we scan the contraction sequence from $G = G_n$ to G_1. We maintain *types* that are "local to the red graph", that is, the theory up to quantifier depth q of all the sentences of depth q that are true on a given red component. As the component twin-width is bounded, note that there is a bounded number of vertices per red component. Initially in $G_n = G$, the red components are single vertices, hence the local theory is easy to determine. Eventually in G_1, the whole graph has been merged into a single vertex, thus the local theory of the unique vertex of G_1 matches the "global" theory of G. As this is precisely what we are after, the crux lies in updating the local theories when moving from trigraph G_{i+1} to trigraph G_i. When contracting $u, v \in V(G_{i+1})$, up to $d + 1$ red components of G_{i+1} are fused into one in G_i, where d is the upper bound on the component twin-width. We show that the local theory on these red components, combined with the black edges sitting on them, is enough to determine unambiguously the local theory of the new red component. The relative simplicity of component twin-width brings our proof down to a minimum: one lemma in the vein of the Feferman-Vaught theorem [14]. For completeness, we prove this folklore lemma with Ehrenfeucht-Fraïssé games for MSO.

We also present an analogue to dynamic programming over clique-width expressions or tree-decompositions, with the approach of contraction sequences. We exemplify it with a practical algorithm for the particular MSO-expressible problem q-Coloring. As the MSO model checking algorithm, it can be described as dynamic programming over the contraction sequence; except it is obviously much more practical and simple than the former. We then list some advantages that our approach hold over the classic dynamic-programming algorithms. In [4], this scheme was used to solve some particular FO-expressible problems on graphs of bounded twin-width given with an $O(1)$-sequence.

This brings us to the second benefit, which is an unexpected collapse of the meta-algorithmic techniques dedicated to handle first-order and monadic second-order logic. Dynamic programming over contraction sequences tackles in one sweep problems that were seemingly as different and required as disparate techniques as k-Subgraph Isomorphism on planar graphs[3] and q-Coloring on graphs of bounded rank-width [12]. Another realization of that collapse is a similar algorithm efficiently solving FO model checking on graphs of bounded twin-width given with $O(1)$-sequences [6], and MSO model checking on graphs of bounded rank-width/bounded component twin-width. This motivates searching for characterizations or generalizations of classes of bounded expansion or nowhere dense classes by means of contraction sequences, as a way to push further their unifying power.

[3] see the work of Eppstein [13] building up on Baker's technique [1] and leading to low tree-width decompositions, and [4] for the approach with contraction sequences

We also define a notion of *spanning twin-width*,[4] intermediate between bounded tree-width and bounded twin-width, which exactly captures classes excluding a minor, among monotone classes.

▶ **Theorem 2.** *A monotone graph class $\mathcal{C}$ has bounded spanning twin-width if and only if it is proper minor-closed.*

So far we explored what happens when *restricting* the notion of bounded twin-width. Here is an attempt to *generalize* it. One may observe that homogeneity of two distinct parts X, Y of $\mathcal{P}_i$ is in fact a directed relation. Let us say that Y is *homogeneous to* X if the pair $X, \{y\}$ is homogeneous for all $y \in Y$. We can now form the *directed red graph* D_i whose vertices are the parts of $\mathcal{P}_i$, and with arcs $X \to Y$ for all pairs $X, Y \in \mathcal{P}_i$ with Y *not* homogeneous to X. The *oriented width* of $\mathcal{P}_i$ is the maximum out-degree w_o of D_i. This defines the *oriented twin-width* of G denoted $\text{otww}(G)$ (see Figure 2b). Note that $\text{otww}(G) \leqslant \text{tww}(G)$ since contracting parts can only create red arcs which are directed from the contracted vertex.

Oriented twin-width is in some sense "fairer" than twin-width, as far as the "error count" is concerned. When an error (red edge) occurs while contracting two parts, it is only accounted to the newly contracted part, and not to other adjacent parts (since only their red in-degree may increase). Surprisingly, we will see that:

▶ **Theorem 3.** *Oriented twin-width and twin-width are functionally equivalent.*

The proof simply revisits the equivalence between bounded twin-width and so-called $O(1)$-*mixed freeness* (see Section 2), and can almost integrally be found in [6]. Our contribution here is mainly conceptual, in identifying the overlooked oriented twin-width. The indirect nature of the proof, which does not immediately provide an $O(1)$-sequence from a partition sequence with bounded oriented width, suggests that something non-trivial is at play. Indeed this greatly simplifies, as we then exemplify, the proof that K_t-minor free graphs have bounded twin-width presented in [6]. We also observe that planar graphs have oriented twin-width at most 9. This translates to the current best upper bound for the *twin-width* of planar graphs.

Another direction to generalize bounded twin-width is to allow contraction sequences to end at "simple" (tri)graphs instead of the 1-vertex graph. Of course for this notion to be new, "simple" should not imply bounded twin-width. Bounded-degree and bounded-expansion are reasonably "tractable" classes with unbounded twin-width [3]. We say that a class $\mathcal{C}$ is *collapsible to a class* $\mathcal{D}$ if graphs of $\mathcal{C}$ admit *partial* $O(1)$-sequences to (tri)graphs in $\mathcal{D}$. We showcase the flexibility of the FO model-checking algorithm in [6]: Collapsible classes to bounded degree and collapsible classes to bounded expansion admit respectively a fixed-parameter tractable FO and $\exists$FO model-checking algorithm, provided a corresponding partial $O(1)$-sequence is given. This is a relatively elementary fusion of the algorithm in [6] and classic techniques from the meta-algorithmic toolbox, namely *Gaifman's locality theorem* and *low tree-depth covers*. On the one hand, it can be seen as a first attempt to unify and extend tractable FO model-checking algorithms on "sparse" classes (bounded degree, bounded expansion) and on possibly "dense" classes (bounded twin-width). On the other hand, we explain why efficiently finding the corresponding partial $O(1)$-sequences, may turn out simpler than computing (complete) $O(1)$-sequences.

[4] the exact definition is somewhat technical and deferred to Section 6

Organization of the paper.

In Section 2 we recall the relevant background. In Section 3 we show Theorem 1, present an alternative proof of Courcelle's theorems, and give a practical algorithm for q-Coloring on graphs of bounded component twin-width (i.e., bounded rank-width). In Section 4, we prove Theorem 3 and use it to bound the twin-width of K_t-minor free graphs. In Section 5, we present some FO model-checking algorithms using partial contraction sequences to classes of bounded degree and bounded expansion. In Section 6, we show the equivalence between proper minor-closed and bounded spanning twin-width, for monotone classes.

2 Preliminaries

We denote by $[i, j]$ the set of integers $\{i, i + 1, \dots, j - 1, j\}$, and $[k]$ is a short-hand for $[1, k]$. We use the standard graph-theoretic definitions and notations. Given a graph G, its vertex set is denoted by $V(G)$ and its edge set by $E(G)$. Given a subset S of $V(G)$, $G[S]$ denotes the subgraph of G induced by S.

Two parameters w and w' defined on graphs (or more generally on matrices) are *functionally equivalent* if there exists a function f such that for every graph G, we have $w(G) \leqslant f(w'(G))$ and $w'(G) \leqslant f(w(G))$. Among classical pairs of functionally equivalent parameters, let us mention branch-width and tree-width, or rank-width and clique-width. When speaking of a *class* $\mathcal{C}$ of graphs, we mean closed under isomorphism. When $\mathcal{C}$ is furthermore closed under taking subgraphs, we speak of *monotone class* and when $\mathcal{C}$ is closed under induced subgraphs, we speak of *hereditary class*. A class $\mathcal{C}$ of graphs is *sparse* if there is a t for which no graph of $\mathcal{C}$ contains the complete bipartite graph $K_{t,t}$ as a subgraph (i.e., not necessarily induced).

2.1 Branch decompositions

A *branch decomposition* of a graph G is a ternary tree T in which the leaves are in one-to-one correspondence with $V(G)$. In particular every edge e of T corresponds to the vertex bipartition B_e of G defined by the two sets of leaves of the connected components of $T \setminus e$. Given now any function m from graph bipartitions to the non-negative integers, we obtain a parameter $b_m(G)$ which is the minimum, over all branch-decompositions T of G of the maximum of $m(B_e)$ over all edges e of T.

For instance, when $m(B_e)$ is the rank (computed in $\mathbb{F}_2$) of the adjacency matrix of the bipartite subgraph of G spanned by the vertex bipartition B_e, the parameter b_m is the *rank-width* of G. We obtain the *linear rank-width* of G by keeping the same rank function m but insisting that branch-decompositions T are ternary trees where the internal vertices form a path. A class has bounded tree-width (resp. bounded path-width) if and only if it is sparse (i.e., $K_{t,t}$-free) and has bounded rank-width (resp. bounded linear rank-width) [17]. Hence tree-width and path-width can be seen as the sparse restrictions of rank-width and linear rank-width.

Let us now introduce a parameter which is equivalent to rank-width. The *boolean-width* $\mathrm{bool}(X, Y)$ of a bipartition (X, Y) of $V(G)$ is the logarithm in base 2 of the number of subsets of Y (equivalently, of X) that are the neighborhood of some subset of X (resp. of Y). The *boolean-width* of a graph G is the parameter $b_{\mathrm{bool}}(G)$. Boolean-width is functionally equivalent to rank-width. When we only consider branch decompositions in which internal vertices form a path, we speak of *linear boolean-width*, which is similarly equivalent to *linear rank-width*.

We will use the next observation implicitly, which is proved in [8].

▶ **Observation 4.** *Let (X, Y) be a vertex bipartition of G, and let q be the maximum number of vertices in X which have distinct neighborhoods in the bipartite graph $G(X, Y)$ (i.e., forgetting edges inside X and Y). The boolean-width of (X, Y) is at most q and at least $\log_2 q$.*

It is often convenient to root a branch-decomposition T of G at some arbitrary non leaf node. Given then an internal node v of T, the leaves of the subtree of T rooted at v is denoted by A_v, and the bipartition $(A_v, V(G) \setminus A_v)$ of G is said to be *associated with* v. Note that there is a unique bipartition associated with an internal node of T.

Branch-decompositions are a very popular concept and can be adapted to partitions of the edge set instead of the vertex set (leading to the branch-width), or to the ground set of a matroid using as parameter its connectivity function (yielding matroid branch-width). One very appealing feature is that parameters defined by branch-decompositions usually admit dual parameters via tangles. However, the main limitation is that planar graphs have unbounded value for parameters defined with branch-decompositions. Indeed every balanced bipartition of a planar graph is complex, and thus return a large value for the classical parameters m, which in turn gives an unbounded b_m. The main way to overcome the "planar barrier" is to measure the complexity of vertex partitions instead of vertex bipartitions.

2.2 Partition sequences

A *partition sequence* of a graph G is a sequence $\mathcal{S} = \mathcal{P}_n, \ldots, \mathcal{P}_1$ of partitions of $V(G)$ where $\mathcal{P}_n := \{\{v\} : v \in V(G)\}$ is the partition of $V(G)$ into singletons, $\mathcal{P}_1 = \{V(G)\}$ is the whole set, and each $\mathcal{P}_i$ is obtained by merging two parts of $\mathcal{P}_{i+1}$. In particular each $\mathcal{P}_i$ consists of i subsets of $V(G)$. A *vertex-ordering compatible with* $\mathcal{S}$ is any total order $\leqslant$ on $V(G)$ (often also seen as a permutation σ) such that for every $\mathcal{P}_i$ and every part X in $\mathcal{P}_i$, the elements of X are consecutive along $\leqslant$.

A function w from vertex-partitioned graphs into the non-negative integers is called a *width*. The *partition-width associated to* w of a graph G is the minimum integer t such that there exists a partition sequence $\mathcal{P}_n, \ldots, \mathcal{P}_1$ of G such that $w(P_i) \leqslant t$ for every $i \in [n]$. We denote it by $p_w(G)$.

We say that two disjoint subsets of vertices X, Y of a graph G are *homogeneous* if there are all edges or no edge between them. More generally, if G is a binary multirelation, we insist that every ordered pairs xy and $x'y'$ where $x, x' \in X$ and $y, y' \in Y$ induce the same structure. By extension, X is homogeneous with X when X is a singleton, but is not homogeneous with X when X has at least two elements. Indeed if $x, x' \in X$, the type of x, x and the type of x, x' are not the same.

Given a graph G and a partition $\mathcal{P}_i$, we consider an auxiliary trigraph G_i, called *quotient trigraph* and denoted by $G/\mathcal{P}_i$, with vertices the part of $\mathcal{P}_i$, *red edges* all pairs of parts X, Y which are not homogeneous, and *black edges* all pairs of parts X, Y for which X, Y is a complete bipartite graph in G. The *red graph* (resp. *black graph*) of G_i has vertex set $V(G_i)$, and edge set its red edges (resp. black edges). By our convention, we add a loop (X, X) to every part X which is not a singleton. Loops count as degree 1. Note that a vertex u of G_i corresponds to a subset of vertices of G which we denote by $u(G)$. It will also be convenient to speak of the *total degree* of u in G_i which is the total number of red and black edges incident to u. The graph obtained from G_i by forgetting the colors (i.e., the graph which is the union of the red and black edges) is called *total graph*.

A *module* in a graph G is a subset of vertices X such that for every $x, x' \in X$ and $y \in V(G) \setminus X$, both xy and $x'y$ induce an edge or both induce a non-edge. More generally, if G is a binary multirelation, we insist that the ordered pairs xy and $x'y$ induce the same structure. When X, Y are two disjoint subsets of vertices, we say that Y *is homogeneous to* X if Y is a module in $G[X \cup Y]$. We obtain a directed version of G_i as follows: The *directed trigraph* D_i is obtained from G_i by orienting red edges XY as $X \to Y$ whenever X is not homogeneous to Y and keeping black edges unchanged. Note that some red edges of D_i can be directed in both ways.

Given G and a partition $\mathcal{P}_i$, we define four possible widths: w_o is the maximum red out-degree of D_i, w_d is the maximum red degree of G_i, w_c is the maximum number of vertices in a red connected component of G_i, and w_t is the total number of red edges in G_i. Note that $w_o \leqslant w_d \leqslant w_c \leqslant w_t$. We now obtain the associated width-parameters:

- The *oriented twin-width* of G as the partition width $\text{otww}(G)$ associated to w_o.
- The *twin-width* of G as the partition width $\text{tww}(G)$ associated to w_d.
- The *component twin-width* of G as the partition width $\text{ltww}(G)$ associated to w_c.
- The *total twin-width* of G as the partition width $\text{ttww}(G)$ associated to w_t.

Observe that $\text{otww}(G) \leqslant \text{tww}(G) \leqslant \text{ltww}(G) \leqslant \text{ttww}(G)$. Note that there is a slight variation with our original definition of twin-width since loops add one to the degree. For the sake of consistency, we will actually drop the red loops for oriented twin-width and twin-width. We will see in Sections 3 and 4 that we did not create new parameters:

▶ **Theorem.** *The following parameters are functionally equivalent:*

- *Twin-width and oriented twin-width.*
- *Component twin-width and rank-width.*
- *Total twin-width and linear rank-width.*

It is somewhat comforting to be on charted territory since the choices of the widths w_o, w_d, w_c, w_t are natural. The equivalence between oriented twin-width and twin-width is very handy, as we will in Section 4.

2.3 The matrix viewpoint

Given an $n \times m$ matrix M, a *row-partition* (resp. *column-partition*) is a partition of the rows (resp. columns) of M. A (k, ℓ)*-partition* (or simply *partition*) of a matrix M is a pair $(\mathcal{R} = \{R_1, \dots, R_k\}, \mathcal{C} = \{C_1, \dots, C_\ell\})$ where $\mathcal{R}$ is a row-partition and $\mathcal{C}$ is a column-partition. A *contraction* of a partition $(\mathcal{R}, \mathcal{C})$ of a matrix M is obtained by performing one contraction in $\mathcal{R}$ or in $\mathcal{C}$.

We distinguish two extreme partitions of an $n \times m$ matrix M: the *finest partition* where $(\mathcal{R}, \mathcal{C})$ have size n and m, respectively, and the *coarsest partition* where they both have size one. A *contraction sequence* of an $n \times m$ matrix M is a sequence of partitions $(\mathcal{R}^1, \mathcal{C}^1), \dots, (\mathcal{R}^{n+m-1}, \mathcal{C}^{n+m-1})$ where

- $(\mathcal{R}^1, \mathcal{C}^1)$ is the finest partition,
- $(\mathcal{R}^{n+m-1}, \mathcal{C}^{n+m-1})$ is the coarsest partition, and
- for every $i \in [n+m-3]$, $(\mathcal{R}^{i+1}, \mathcal{C}^{i+1})$ is a contraction of $(\mathcal{R}^i, \mathcal{C}^i)$.

Given a subset R of rows and a subset C of columns in a matrix M, the *zone* $R \cap C$ denotes the submatrix of all entries of M at the intersection between a row of R and a column of C. A *zone* of a partition pair $(\mathcal{R}, \mathcal{C}) = (\{R_1, \dots, R_k\}, \{C_1, \dots, C_\ell\})$ is any $R_i \cap C_j$ for $i \in [k]$ and $j \in [\ell]$. A zone is *constant* if all its entries are identical. The *error value* of R_i is the number of non constant zones among all zones in $\{R_i \cap C_1, \dots, R_i \cap C_\ell\}$. We adopt a similar definition for the *error value* of C_j. The *error value* of $(\mathcal{R}, \mathcal{C})$ is the maximum error value taken over all R_i and C_j. The *twin-width* of a matrix M is the minimum t for which there exists a contraction sequence of M consisting of partitions with error value at most t.

In a contraction sequence of a matrix M, one can always reorder the rows and the columns of M in such a way that all parts of all partitions in the contraction sequence consist of consecutive rows or consecutive columns. To mark this distinction, a *row-division* is a row-partition where every part consists of consecutive rows; with the analogous definition for *column-division.* A (k, ℓ)*-division* (or simply *division*) of a matrix M is a pair $(\mathcal{R}, \mathcal{C})$ of a row-division and a column-division with respectively k and ℓ parts. A *division sequence* is a contraction sequence in which all partitions are divisions.

A matrix $M = (m_{i,j})$ is *vertical* (resp. *horizontal*) if $m_{i,j} = m_{i+1,j}$ (resp. $m_{i,j} = m_{i,j+1}$) for all i, j. Observe that a matrix which is both vertical and horizontal is constant. We say that M is *mixed* if it is neither vertical nor horizontal. A crucial remark is that a matrix is mixed if and only if it contains a *corner*, that is any 2-by-2 mixed submatrix of the form $(m_{i,j}, m_{i+1,j}, m_{i,j+1},\, m_{i+1,j+1})$. A t*-mixed minor* in M is a division $(\mathcal{R}, \mathcal{C}) = (\{R_1, \dots, R_t\}, \{C_1, \dots, C_t\})$ such that every zone $R_i \cap C_j$ is mixed (hence contains a corner). A matrix without t-mixed minor is t*-mixed free.* The minimum t for which one can reorder the column and the rows of M to form a t-mixed free matrix is called the *mixed value* of M.

▶ **Theorem 5** ([6]). *Twin-width and mixed value are functionally equivalent for matrices.*

Given a graph G and a permutation σ of its vertex set, we denote by $Adj_\sigma(G)$ the adjacency matrix of G in which the columns and the rows are ordered according to σ. As usual, given two vertices u, v, the entry $Adj_\sigma(G)_{u,v}$ is equal to 1 if uv is an edge and 0 otherwise. By extension, we say that the *mixed value* of a graph G is the minimum t for which $Adj_\sigma(G)$ is t-mixed free, taken over all permutations σ. The link between mixed value and twin-width for graphs was proved in [6]:

▶ **Theorem 6** ([6]). *Twin-width and mixed value are functionally equivalent for graphs.*

2.4 Bounded expansion and tree-depth covers

We recall some definitions from a paper by Plotkin, Rao, and Smith [28] and from the sparsity program of Ossona de Mendez and Nešetřil [24]. One possible way of defining a minor of a graph G is by a collection of disjoint sets $B_1, B_2, \dots, B_h \subseteq V(G)$, called *branch sets*, such that $G[B_i]$ is connected for all $i \in [h]$. A *minor* of G is then any graph H, say on vertex set $[h]$, such that $ij \in E(H)$ implies that there is an edge in G with one endpoint in B_i and the other endpoint in B_j. A *depth-r minor* (also called *r-shallow minor*) of a graph G is a minor H of G that can be obtained such that each branch set induces in G a subgraph with radius at most r. Let us denote by $\nabla_r(G)$ the set of all the depth-r minors of G. In particular $\nabla_r(G)$ is subgraph-closed. Given a non-decreasing function $f : \mathbb{N} \to \mathbb{N}$, we say that a graph G has *expansion* f if for every $r \in \mathbb{N}$, $\nabla_r(G)$ has (maximum) average degree at most $f(r)$. A graph class $\mathcal{C}$ has *expansion* f if all its graphs have expansion f, and $\mathcal{C}$ have *bounded expansion* if it has expansion f for some function f. Note that saying that a graph has bounded expansion is meaningless (they all do, individually) but the fact that, for a specific function f, a single graph has expansion f is meaningful.

The *tree-depth* of a graph G is the minimum integer td such that there is rooted forest F of height td on vertex set $V(G)$ with every edge of G being in an ancestor-descendant relationship in F. Bounded tree-depth is more restrictive than bounded tree-width, so in particular, bounded tree-depth graphs have bounded twin-width. There is a very useful connection between bounded tree-depth and bounded expansion, in the form of *low tree-depth* *covers*, or the related low tree-depth decompositions [22, 23]. A *low tree-depth cover* with parameters k, f of a graph G is a family of $h = f(k)$ subsets $X_1, \ldots, X_h \subseteq V(G)$ such that, for every $i \in [h]$, $G[X_i]$ has tree-depth at most k, and every subset of $V(G)$ of size at most k is fully included in at least one X_i. A graph class $\mathcal{C}$ has *low tree-depth covers* if there is a function f depending only on $\mathcal{C}$ such that for every $G \in \mathcal{C}$ and integer k, G has a low tree-depth cover with parameters k, f.

▶ **Theorem 7** ([23]). *Every monotone class has low tree-depth covers if and only if it has bounded expansion. Furthermore a low tree-depth cover of any graph drawn from a class of bounded expansion can be computed in linear time.*

2.5 Finite model theory

We recall some relevant background from finite model theory. We denote by FO_τ and MSO_τ the set of first-order, respectively monadic second-order, formulas on signature τ. In first-order, every variable is interpreted as an element of the universe. In monadic second-order, a first-order variable is interpreted as an element, while a second-order variable is interpreted as a subset of the universe. We will mainly consider signatures consisting of unary and binary relation symbols only. Typically the signature τ will be one of the following:

- $\{E\}$, where E is binary: the language of graphs with possible edge orientations and loops;
- $\{E, \sim\}$, where $\sim$ is interpreted as an equivalence relation: the language of graphs with an unlabeled partition;
- $\{E, U_1, \ldots, U_d\}$, where $U_1, \ldots, U_d$ are unary relations interpreted as a partition of the universe: the language of colored graphs, or graphs with a labeled partition;
- $\{\mathrm{inc}\}$, where inc is interpreted as an vertex-edge incidence graph.

$\mathrm{MSO}_{\{E\}}$ is usually denoted by MSO_1, and $\mathrm{MSO}_{\{\mathrm{inc}\}}$ by MSO_2. A *sentence* is a formula without free variables. A *relational τ-structure $\mathscr{A}$* on universe A gives an interpretation $R^{\mathscr{A}} \subseteq A^r$ to every r-ary relation symbol $R \in \tau$. A structure $\mathscr{A}$ is a *model* of a sentence φ, denoted by $\mathscr{A} \models \varphi$, if φ holds when interpreted on $\mathscr{A}$. We will only consider *finite models* where the *universe* A is a finite set. The *FO model checking* (resp. *MSO model checking*) asks given a τ-structure $\mathscr{A}$ and a sentence $\varphi \in \mathrm{FO}_\tau$ (resp. $\varphi \in \mathrm{MSO}_\tau$) whether $\mathscr{A} \models \varphi$ holds. The fragment $\exists$FO (existential first-order logic) consists of the formulas with no universal quantifier and all the negations pushed down to atomic formulas.

The *Gaifman graph* of a τ-structure $\mathscr{A}$ has vertex set its universe A, and edges ab whenever a and b appear in the same relation $R^{\mathscr{A}}$ for some $R \in \tau$. The *quantifier depth* (or *quantifier rank*) of a formula φ is the largest number of quantifiers that are nested in φ. $\mathrm{FO}_\tau[q]$ (resp. $\mathrm{MSO}_\tau[q]$) denotes the set of formulas in FO_τ (resp. MSO_τ) with quantifier depth at most q. When the signature is irrelevant or clear from the context, we may omit it, and simply write FO, MSO, FO$[q]$, MSO$[q]$.

If two finite τ-structures are not isomorphic, then there is a sentence that holds in one but not in the other (for instance the sentence that fully describes the former structure). However it is very well possible that two non-isomorphic τ-structures satisfy the exact same sentences of FO$[q]$ or MSO$[q]$, for some (finite) integer q. Ehrenfeucht-Fraïssé games characterize exactly when that happens. Initially the game was defined for first-order logic. We call it the *EF game* and start with its description. We will then present its extension *MSO-EF* for monadic second-order.

In the EF game, two players *Spoiler* and *Duplicator* confront each other over two τ-structures $\mathscr{A}$ and $\mathscr{B}$. They play a succession of rounds, when Spoiler wants to show that $\mathscr{A}$ and $\mathscr{B}$ are not isomorphic, whereas Duplicator tries to argue the opposite. The i-th round goes like this. Spoiler chooses a structure $\mathscr{A}$ or $\mathscr{B}$, and picks one element in it, say $a_i \in A$ (or $b_i \in B$). Duplicator answers by picking an element in the other structure, say $b_i \in B$ (resp. $a_i \in A$). If after q rounds, $a_i \mapsto b_i$ (for $i \in [q]$) is still an isomorphism between the induced substructures $(\mathscr{A}, =)[a_1, \ldots, a_q]$ and $(\mathscr{B}, =)[b_1, \ldots, b_q]$, we say that Duplicator has *survived q rounds* of the EF game.

We write $\mathscr{A} \equiv_q^{\mathrm{FO}} \mathscr{B}$ if Duplicator has a strategy such that she can survive (at least) q rounds. The Ehrenfeucht-Fraïssé theorem states that this is equivalent to $\mathscr{A}$ and $\mathscr{B}$ agreeing on all the sentences of $\mathrm{FO}_\tau[q]$.

▶ **Lemma 8** (Ehrenfeucht-Fraïssé, see Theorem 3.9 in [21]). *Let $\mathscr{A}$ and $\mathscr{B}$ be two τ-structures. $\mathscr{A}$ and $\mathscr{B}$ satisfy the same sentences of $FO_\tau[q]$ if and only if $\mathscr{A} \equiv_q^{FO} \mathscr{B}$.*

The MSO-EF game is similar to the EF-game, but Spoiler can alternatively decide to play a subset of A (or a subset of B). To which Duplicator answers with a subset of B (resp. of A). Now after q rounds, a tuple of e elements have been played in both $\mathscr{A}$ and $\mathscr{B}$, say $(a_1, \ldots, a_e)$ and $(b_1, \ldots, b_e)$ in this order, as well as a tuple of s sets, say $(A_1, \ldots, A_s)$ in $\mathscr{A}$ and $(B_1, \ldots, B_s)$ in $\mathscr{B}$, with $q = e + s$. Duplicator has survived these q rounds if $a_i \mapsto b_i$ (for $i \in [e]$) is an isomorphism between $(\mathscr{A}, =, A_1, \ldots, A_s)[a_1, \ldots, a_e]$ and $(\mathscr{B}, =, B_1, \ldots, B_s)[b_1, \ldots, b_e]$. Similarly we write $\mathscr{A} \equiv_q^{\mathrm{MSO}} \mathscr{B}$ if Duplicator has a strategy allowing her to survive (at least) q rounds of the MSO-EF game. The same characterization holds for MSO and the MSO-EF game.

▶ **Lemma 9** (Ehrenfeucht-Fraïssé for MSO, see Corollary 7.8 in [21]). *Let $\mathscr{A}$ and $\mathscr{B}$ be two τ-structures. $\mathscr{A}$ and $\mathscr{B}$ satisfy the same sentences of $MSO_\tau[q]$ if and only if $\mathscr{A} \equiv_q^{MSO} \mathscr{B}$.*

3 From branch-decompositions to contraction sequences

We start this section by showing Theorem 1, that is, the functional equivalence between boolean-width (equivalently rank-width) and component twin-width, and between linear boolean-width and total twin-width.

3.1 Classical width parameters as contraction sequences

▶ **Theorem 10.** *Boolean-width and component twin-width are functionally equivalent.*

Proof. Let G be a graph. We first show that the component twin-width of G is bounded in terms of the boolean-width of G.

Let T be a rooted branch-decomposition of G whose leaves are bijectively mapped to $V(G)$, and assume that T has boolean width at most d. We make a sequence of contractions $G_n, \ldots, G_1$ such that the size of any red component in the trigraph sequence is at most 2^{d+1}. For this, a rooted branch-decomposition T_i for each trigraph G_i is constructed along the contraction sequence while the next invariant is maintained:

(♣) For each node v of T_i with $|A_v| \geqslant 2^d + 1$, the boolean-width of the bipartition $(A_v, V(G_i) \setminus A_v)$ is at most d and all edges of G_i crossing $(A_v, V(G_i) \setminus A_v)$ are black.

The invariant (♣) clearly holds for $G_n = G$ and $T_n = T$. Moreover, if T_i has no proper rooted subtree with at least $2^d + 1$ leaves, then contracting an arbitrary pair of vertices of G_i trivially preserves the invariant. Furthermore G_i has at most 2^{d+1} vertices, hence component twin width is at most 2^{d+1} for G_i and all G_j with $j \leqslant i$. Therefore, we may assume that (♣)

holds for a trigraph G_{i+1} and a rooted branch-decomposition T_{i+1} where $i+1 \geqslant 2^{d+1}+1$ and show how to construct G_i and T_i.

Observe that there exists a node v of T_{i+1} such that $2^d+1 \leqslant |A_v| \leqslant 2^{d+1}$. Just consider for this a node v such that A_v has size at least 2^d+1 which is the furthest from the root. By Observation 4 and the first part of (♣) applied to v, there are two distinct vertices x, y of G_{i+1} which belong to A_v such that x, y have the same (black) neighborhood in $V(G_{i+1}) \setminus A_v$. Now contract x, y to yield G_i. Let T_i be a branch-decomposition of G_i obtained by deleting y and identifying the node x to the new vertex of G_i resulting from the contraction of x and y. Due to the choice of x, y and the second part of (♣) on $i+1$, the edges between A_v (of the new tree T_i) and $V(G_i) \setminus A_v$ are all black. This means that any bipartition of $V(G_i)$ that can potentially contain a newly created red edge of G_i must be associated with a strict descendant of v. By the definition of v, any strict descendant of v has at most 2^d leaves (both in T_{i+1} and T_i) and is thus out of the scope of the invariant (♣). Therefore, the second part of (♣) is maintained.

This also means that for any node u of T_i which is not a strict descendant of v, the bipartition $(A_u, V(G_i) \setminus A_u)$ associated with u is the same as the bipartition associated with u in T_{i+1} after deleting one vertex of G_{i+1}, namely y. Since the boolean-width of a bipartition does not increase after vertex deletion, we conclude that the first part of (♣) is maintained as well. Finally, we observe that the invariant (♣) indicates that G has component twin-width at most 2^{d+1} since any red component of G_i is included inside some A_v with size at most 2^{d+1}.

To see the other direction, let $\mathcal{P}_n, \dots, \mathcal{P}_1$ be a partition sequence of G such that every connected component of the red graph G_i has at most d vertices. Let $\mathcal{P}'_i$ be the coarsening of $\mathcal{P}_i$ such that each part of $\mathcal{P}'_i$ corresponds to a red component of G_i, i.e., is the union of parts of $\mathcal{P}_i$ which form a red component in G_i. Slightly abusing the notation, we call a part of $\mathcal{P}'_i$ a red component of $\mathcal{P}_i$.

Let T_n be a star tree rooted at its center r, whose n leaves are bijectively mapped to $V(G)$. We will iteratively transform a rooted tree T_{i+1} to T_i in a way that mirrors the merging of parts in $\mathcal{P}'_i$. The root r will be unchanged throughout the transformations. During iterative transformations we maintain the following invariants:

(a) The leaves of each connected component of $T_i - r$ are mapped to each part of $\mathcal{P}'_i$.
(b) The root r has as many children as $|\mathcal{P}'_i|$ and all other internal nodes have two children.
(c) For every edge of T_i, the associated bipartition has boolean width at most 2^d.

When $i = n$, the invariants (a)-(c) clearly hold. Suppose $T_n, \dots, T_{i+1}$ satisfy the invariants (a)-(c), and $i \geqslant 1$. Notice that $\mathcal{P}'_i$ is a coarsening of $\mathcal{P}'_{i+1}$ (possibly $\mathcal{P}'_{i+1} = \mathcal{P}'_i$), that there is a unique red component $C \in \mathcal{P}'_i$, obtained as the union of some (possibly one) red components $C_1, \dots, C_s$ of $\mathcal{P}'_{i+1}$, and that $\mathcal{P}'_i \setminus C = \mathcal{P}'_{i+1} \setminus \{C_1, \dots, C_s\}$. By the invariant (a), there are subtrees of $T_{i+1} - r$ whose leaves are mapped to parts $C_1, \dots, C_s$ of $\mathcal{P}'_{i+1}$. Let $t(C_j)$ be the root of the subtree of $T_{i+1} - r$ corresponding to C_j for $j \in [s]$. Now, we construct T_i from T_{i+1} as follows: replace the edges connecting the root r_{i+1} and $t(C_j)$ for $j \in [s]$ by a subcubic tree rooted at $t(C)$ with s leaves, whose root $t(C)$ becomes the child of r_{i+1} and whose s leaves are identified (arbitrarily) with $t(C_j)$ for $j \in [s]$.

By the induction hypothesis and the construction of T_i, the invariants (a)-(b) are maintained. Furthermore, since $C = \bigcup_{i=1}^{s} C_i$, the red component $C \in \mathcal{P}'_i$ consists of at most d parts of $\mathcal{P}_i$, thus at most $d+1$ parts of $\mathcal{P}_{i+1}$, and we have $s \leqslant d+1$. To see that the boolean width of T_i is at most 2^d, it suffices to check that for all $I \subseteq [s]$, the boolean-width of the bipartition $(\bigcup_{j \in I} C_j, V(G) \setminus \bigcup_{j \in I} C_j)$ is at most 2^d. For a proper subset I of $[s]$, we know that $\bigcup_{j\in I} C_j$ consists of at most d parts of $\mathcal{P}_{i+1}$ and each of these parts of $\mathcal{P}_{i+1}$ has the same neighborhood across the bipartition $(\bigcup_{j\in I} C_j, V(G) \setminus \bigcup_{j\in I} C_j)$ since each C_j is a red component. Hence, the vertex set $\bigcup_{j\in I} C_j$ has at most d vertices with distinct neighborhood across the bipartition and thus the boolean width is at most 2^d. For $I = [s]$, the same argument applies once it is noted that $\bigcup_{j\in I} C_j = C$ consists of at most d parts of $\mathcal{P}_i$.

As $\mathcal{P}'_1 = \mathcal{P}_1 = \{V(G)\}$, the invariants (a) and (b) at $i = 1$ imply that T_1 is a subcubic tree whose leaves are bijectively mapped to $V(G)$. With the invariant (c), we conclude that T_1 and the bijection form a boolean decomposition of width at most 2^d. ◀

▶ **Theorem 11.** *Linear boolean-width and total twin-width are functionally equivalent.*

Proof. Let G be a graph. We first bound total twin-width in terms of linear boolean-width. Let T be a linear branch-decomposition of G (i.e. in which the internal nodes form a path P) with boolean-width at most d. We root T at an internal node which has degree 1 in the path P. We follow the same proof as for Theorem 10 and observe that every tree T_i is now a linear branch-decomposition. Indeed, the linearity of T_i implies that there is a unique choice of a minimal rooted subtree of T_i with at least $2^d + 1$ leaves. Moreover, the invariant (♣) of Theorem 10 means that the endpoints of any red edge are restricted to the leaves of this subtree. Note that there are at most $2^d + 1 + \binom{2^d+1}{2}$ red edges in any G_i.

Let $\mathcal{P}_n, \ldots, \mathcal{P}_1$ be a partition sequence of G achieving total twin-width at most d. Let $V_i \subseteq V(G)$ be the set of (original) vertices which were ever contracted in the contraction sequence, or equivalently, V_i is the union of all non-singleton parts of $\mathcal{P}_i$. Because a part of $\mathcal{P}_i$ is incident with a red edge if and only if it is not a singleton (recall that we add a red loop), at most d parts of $\mathcal{P}_i$ are incident to red edges. Note that $V_i \setminus V_{i+1}$ has at most two vertices, and it has two vertices only when we contract two singleton parts. Consider now any total order $\prec$ on $V(G)$ such that $u \prec v$ if u was contracted before v. Indeed, we have $V_{n-1} \prec V_{n-2} \setminus V_{n-1} \prec \cdots \prec V_2 \setminus V_3 \prec V_1 \setminus V_2$. Now consider the linear branch decomposition T corresponding to $\prec$ and observe that the boolean-width of T is bounded by $d+1$; every bipartition corresponds either to a cut $(V_i, V(G) \setminus V_i)$ or to $(V_i \cup \{v\}, V(G) \setminus (V_i \cup \{v\}))$ for some $v \in V_{i+1} \setminus V_i$. Note that in both cases there are at most $d+1$ equivalence classes, where each class has the same neighborhood across the cut. Therefore, with Observation 4 we deduce that the boolean-width of T is at most $d+1$. ◀

Here again, we can find equivalent parameters in the sparse regime:

▶ **Theorem 12.** *In the class of $K_{t,t}$-free graphs:*
- *tree-width and component twin-width are functionally equivalent.*
- *path-width and total twin-width are functionally equivalent.*

Note that if we do not add red loops to contracted vertices, linear rank-width and total twin-width are not equivalent because of cographs. Thus the addition of loops may seem a bit artificial and even made to force the equivalence. There is however a good reason for loops: From the adjacency-matrix viewpoint (which is both used for rank-width and twin-width), the main diagonal represents equality and thus should not be confused with edges or non-edges.

3.2 Alternative proof of Courcelle's theorems

We will give an alternative proof to the celebrated result by Courcelle, Makowsky, Rotics [11] that MSO_1 model checking can be solved in linear time on bounded clique-width graphs, given with an $O(1)$-expression. This generalizes the original Courcelle's theorem [9] that

MSO_2 model checking can be solved in linear time on bounded tree-width graphs. Indeed MSO_2 is not more expressible than MSO_1 on graphs of bounded tree-width [10, Theorem 9.37], and there is a linear-time FPT algorithm returning a tree-decomposition of optimal width [2]. We observe that there are other alternative proofs to the central result of Courcelle, Makowsky, Rotics; one based on automata [16], and one game-theoretic [20].

We call *MSO rank-q type* (or *type* for short) any set of sentences

$$\text{mso-tp}_q(G) := \{\varphi \in \text{MSO}_{\{E\}}[q] \ : \ G \models \varphi\}$$

where G is a graph, and we recall, $\text{MSO}_{\{E\}}[q]$ denotes the set of MSO sentences on a signature with a single binary relation E, and quantifier depth at most q. Then $\text{mso-tp}_q(G)$ is called the MSO rank-q type of G, or type of G for short.

We fix a positive integer d, upperbounding the component twin-width on the class we want to tackle. We call *local MSO rank-q partitioned type* (or *local partitioned type* for short) any set of sentences

$$\text{loc-mso-tp}_{q,d}(G, \mathcal{P}^o, C) := \{\varphi \in \text{MSO}_{\{E,U_1,\ldots,U_d\}}[q] \ : \ (G[C], \mathcal{P}^o[C]) \models \varphi\}$$

where G is a graph with a labeled vertex-partition $\mathcal{P}^o$, and C is the set of initial vertices of G landing in a red component (i.e., a connected component in the red graph of the quotient trigraph $G/\mathcal{P}$) with at most d parts. The unary relations $U_1, \ldots, U_d$ are interpreted as the labeled partition $\mathcal{P}^o[C]$ of $G[C]$. Thus some $U_{d'+1}, U_{d'+2}, \ldots, U_d$ may possibly be empty if $\mathcal{P}^o[C]$ has $d' < d$ parts. We use the superscript o (for **o**rdered) for the labeled partition $\mathcal{P}^o$ since we will usually fix the labeling by giving an ordering of the parts. We may sometimes identify C with the corresponding red component in $G/\mathcal{P}$, and $\mathcal{P}$ will denote the partition $\mathcal{P}^o$ ignoring the labels.

It is not difficult to show that there is only finitely many MSO sentences of quantifier depth q on finitary signatures, up to logical equivalence (see for instance [21, Proposition 7.5]). Furthermore there is an algorithm (taking time function of q and signature τ) that lists all the sentences of depth q, up to logical equivalence. Therefore the number of (local) MSO rank-q (partitioned) types is bounded by a function of q and d only. Again all the (local) MSO rank-k (partitioned) types can be listed in time function of q and d only. Instead of deciding $G \models \varphi$ for a particular sentence φ with quantifier depth q, we will compute $\text{mso-tp}_q(G)$. By the previous observation, this allows to decide $G \models \varphi$ for *every* sentence φ with quantifier depth q in constant time, if q and d are treated as a constant.

The algorithm will only compute local partitioned types. Note that for an n-vertex graph G with a partition sequence $\mathcal{P}_n, \ldots, \mathcal{P}_1$, $\text{loc-mso-tp}_{q,d}(G, \mathcal{P}_n^o = \{\{v\} \ : \ v \in V(G)\}, C)$ is easy to determine, with any labeling $\mathcal{P}_n^o$ of $\mathcal{P}_n$, since $G/\mathcal{P}_n$ is isomorphic to the *graph* G and each possible C is a singleton $\{w\}$ (for some $w \in V(G)$). Thus these local partitioned types all coincide to the one of the 1-vertex graph with its unique labeled partition. Furthermore $\text{loc-mso-tp}_{q,d}(G, \mathcal{P}_1 = \{V(G)\}, V(G))$ matches[5] $\text{mso-tp}_q(G)$, which is the set we are after.

Thus we only need to compute all the local partitioned types $\left(\text{loc-mso-tp}_{q,d}(G, \mathcal{P}_i^o, C)\right)_C$ from the knowledge of $\left(\text{loc-mso-tp}_{q,d}(G, \mathcal{P}_{i+1}^o, C')\right)_{C'}$. It is at all possible since the local partitioned types, the contracted pair of parts (X, X'), and the black edges of the quotient trigraph are enough to reconstitute the local partitioned type of the new red component containing $X \cup X'$. We show that fact with the characterization via the Ehrenfeucht-Fraïssé

[5] Strictly speaking $\text{loc-mso-tp}_{q,d}(G, \mathcal{P}_1 = \{V(G)\}, V(G))$ is a superset of $\text{mso-tp}_q(G)$, but its projection to sentences ignoring the (trivial) labeled partition is exactly $\text{mso-tp}_q(G)$.

game for MSO (see Lemma 9). Recall that given two input structures $\mathscr{A}, \mathscr{B}$, Duplicator has a strategy to survive q rounds of the MSO-EF game if and only if $\mathscr{A}$ and $\mathscr{B}$ satisfy the same sentences of MSO[q], hence have the same rank-q type.

The subsequent Lemma 13 has a technical and lengthy pre-condition that we chose to state outside the lemma environment for the sake of legibility. It starts here. Let $(G^1, (X^1_1, \ldots, X^1_{\ell_1})), \ldots, (G^k, (X^k_1, \ldots, X^k_{\ell_k}))$ be k graphs G^j given with a labeled partition of size ℓ_j. Let $(H^1, (Y^1_1, \ldots, Y^1_{\ell_1})), \ldots, (H^k, (Y^k_1, \ldots, Y^k_{\ell_k}))$ be such that

$$(G^j, (X^j_1, \ldots, X^j_{\ell_j})) \equiv^{\mathrm{MSO}}_q (H^j, (Y^j_1, \ldots, Y^j_{\ell_j})), \quad \text{for every } j \in [k].$$

Let $(G, \mathcal{P}^o)$ be a graph with a labeled vertex-partition made from the disjoint union

$$\bigcup_{j \in [k]} (G^j, (X^j_1, \ldots, X^j_{\ell_j}))$$

with parts labeled by the order σ, say,

$$(X^1_1, \ldots, X^1_{\ell_1}, X^2_1, \ldots, X^2_{\ell_2}, \ldots, X^k_1, \ldots, X^k_{\ell_k}),$$

and adding the biclique between some pairs of parts $X^j_h, X^{j'}_{h'}$ prescribed by a meta-graph B on vertex set $\mathcal{P}^o$.

The natural bijection $\iota : X^j_h \mapsto Y^j_h$ (for $j \in [k]$ and $h \in [\ell_j]$) allows to transpose σ and B to the union of the H^j. Let $(H, \mathcal{Q}^o)$ be the graph with a labeled partition made from the disjoint union $\bigcup_{j \in [k]} (H^j, (Y^j_1, \ldots, Y^j_{\ell_j}))$ with the parts labeled along $\iota(\sigma)$, and adding the bicliques prescribed by $\iota(B)$. Finally we distinguish two parts (the parts to be contracted) X, X' in $\mathcal{P}^o$, and we distinguish the homologous parts $Y := \iota(X), Y' := \iota(X')$ in $\mathcal{Q}^o$.

▶ **Lemma 13.** $(G, \mathcal{P}^o, X, X') \equiv^{MSO}_q (H, \mathcal{Q}^o, Y, Y')$.

Proof. The *global* strategy of Duplicator simply follows the corresponding local strategy if a vertex is played, and if a set S is played, the union of the local answers to each projection of S on the red components is replied.

More precisely, if Spoiler plays $x_s \in X^j_h$ (or $y_s \in Y^j_h$), then Duplicator answers $y_s \in Y^j_h$ (resp. $x_s \in X^j_h$) accordingly to her local strategy on $(G^j, (X^j_1, \ldots, X^j_{\ell_j})), (H^j, (Y^j_1, \ldots, Y^j_{\ell_j}))$. Importantly we know that Duplicator replies a vertex of Y^j_h to a vertex of X^j_h played by Spoiler, since otherwise the *local* unary relation U_h over G^j contradicts the partial isomorphism ensured by $(G^j, (X^j_1, \ldots, X^j_{\ell_j})) \equiv^{\mathrm{MSO}}_q (H^j, (Y^j_1, \ldots, Y^j_{\ell_j}))$. (Duplicator also remembers that moves (x_s, y_s) have been added to the local j-th game, in case more moves are played there.) If Spoiler plays a set $S_p \subseteq V(G)$, Duplicator considers all the non-empty sets $S_p \cap V(G^j)$ (for $j \in [k]$) and replies $T_p := \bigcup_j A_j$ where A_j is the local answer to $S_p \cap V(G^j)$. Duplicator builds similarly an answer $T_p \subseteq V(H)$ to a move $S_p \subseteq V(G)$ by Spoiler.

Since $(G^j, (X^j_1, \ldots, X^j_{\ell_j})) \equiv^{\mathrm{MSO}}_q (H^j, (Y^j_1, \ldots, Y^j_{\ell_j}))$, the projection of the mapping $x_s \mapsto y_s$ (for s indexing the vertex moves) onto G^j, H^j is a partial isomorphism between the two corresponding local structures. Since there is the same (black) graph B on the parts of $\mathcal{P}$, as $\iota(B)$ on the parts of $\mathcal{Q}$, there is an edge in G between $x_s \in X^j_h$ and $x_{s'} \in X^{j'}_{h'}$ if and only if there is an edge in H between $y_s \in X^j_h$ and $x_{s'} \in X^{j'}_{h'}$. For every s and p, $x_s \in S_p$ if and only if $y_s \in T_p$ otherwise the moves (x_s, y_s) and $(S_p \cap V(G^j), T_p \cap V(H^j))$, played in some order, where $x_s \in V(G^j)$ would make Duplicator lose the local j-th game. Finally since the parts X, X' and Y, Y' are homologous (under the bijection ι), $x_s \in X$ (resp. $x_s \in X'$) if and only if $y_s \in Y$ (resp. $y_s \in Y'$). Otherwise we already observed that (x_s, y_s) would have been a losing pair of moves for Duplicator in the corresponding local game. Thus the mapping

$x_s \mapsto y_s$ (for s indexing the vertex moves) is a partial isomorphism between $(G, \mathcal{P}^o, X, X')$ and $(H, \mathcal{Q}^o, Y, Y')$. ◀

By Lemma 9, we have just established that the local partitioned type of a new red component C' obtained by the merge of two parts X, X' is function of the local partitioned type of every component $C_1, \ldots, C_\ell$ ending up in C' after the contraction, the contracted pair (X, X'), and the transversal black edges (bicliques) linking some pairs of parts in two distinct C_i's.

The crucial place where the upper bound d on the component twin-width comes into play is in the time that the update from $\left(\text{loc-mso-tp}_{q,d}(G, \mathcal{P}^o_{i+1}, C')\right)_{C'}$ to $\left(\text{loc-mso-tp}_{q,d}(G, \mathcal{P}^o_i, C)\right)_C$ takes. Let $Z \in \mathcal{P}_i$ be the result of the merge of the two parts $X, X' \in \mathcal{P}_{i+1}$. Since all the red components have size at most d, the set Z is in a red component with a set $\mathcal{Q}$ of at most $d-1$ other parts of $\mathcal{P}_i$. The black edges in $G/\mathcal{P}_{i+1}$ on the vertex subset $\mathcal{Q}' := \{X, X'\} \cup \mathcal{Q}$, the pair of parts (X, X'), and the local partitioned type of each red component within $\mathcal{Q}'$, account for less than $2^{\binom{d+1}{2}} \binom{d+1}{2} (d+1) f(q)$ outcomes, for some function f. Thus the transition table giving the new local type can be precomputed in time depending only on d and q. (The red components of $G/\mathcal{P}_i$ not containing Z do not need an update.) Treating d and q as constant, the update takes constant time, and the overall algorithm, $O(n)$ time.

If the partition sequence is not given with the input graph, we rely on an algorithm approximating rank-width [26] to find the sequence. Our proof looks like the original one by Courcelle, Makowsky, Rotics [11], except it does not need to use transductions to deal with the label-joins and relabelings of the clique-width expression. Instead everything is concentrated in Lemma 13, a statement similar to Feferman-Vaught theorem [14], which is invoked in [11] to handle the disjoint union of two labeled graphs.

3.3 Simpler algorithm for a particular MSO$_1$ problem: q-Coloring

Like for tree-width and clique-width, one can design more practical algorithms for particular MSO-expressible problems, when the component twin-width is bounded, still utilizing the viewpoint of contraction sequences. This gives rise to a different dynamic-programming scheme than the one on tree-decompositions or on clique-width expressions. It comes naturally *positively-instance driven* [30], that is, generating only positive subproblems. This is known to have significantly sped up some exact algorithms, as the computation of tree-width and tree-decompositions (see again the work of Tamaki [30]). The approach by contraction sequences has other advantages that we will list after we give a particular example. We present an algorithm for q-Coloring which, given an n-vertex graph G and a contraction sequence $G = G_n, \ldots, G_1$ witnessing that its component twin-width is at most d, runs in time $O((2^q-1)^d d^2 n)$.

If $C \subseteq V(G_i)$ is a red component, that is, a connected component in the red graph of G_i, we denote by $C(G)$ the set $\bigcup_{u \in C} u(G)$. A *q-coloring profile* (or *profile* for short) of C is a function $\gamma : V(C) \to 2^{[q]} \setminus \{\emptyset\}$ such that there is a proper q-coloring c of $G[C(G)]$ satisfying, for every $u \in C$, that $c(u(G)) = \gamma(u)$. Thus γ gives the exact set of colors used by a (contracted) vertex of the red component. We will maintain for each red component C the complete set of profiles of C.

Description of the algorithm. Initially in G_n, there are n red components isomorphic to the 1-vertex graph. Thus for each $u \in V(G)$, we store the set of profiles $\{u \mapsto \{1\}, u \mapsto \{2\}, \ldots, u \mapsto \{q\}\}$. This corresponds to the q ways a vertex can be colored. Eventually in

G_1, if there is a profile for the unique red component (again a 1-vertex graph), it means that G is q-colorable.

We shall just update the profiles as the red components evolve. Let u, v be the vertices contracted into a vertex z when going from G_{i+1} to G_i. Let C be the red component of G_i containing z, and $C_1, \dots C_{d'}$ be the red components in G_{i+1} such that $C = \bigcup_{j \in [d']} C_j \setminus \{u, v\} \cup \{z\}$. Since $|C| \leqslant d$, it holds that $|\bigcup_{j \in [d']} C_j| \leqslant d + 1$, and in particular $d' \leqslant d + 1$. Say that $u \in C_a$ and $v \in C_b$.

The update only consists of computing a set of profiles for C (and destroying the set of profiles of $C_1, \dots, C_{d'}$). For every $\gamma_1, \dots, \gamma_{d'}$ in the profile set of $C_1, \dots, C_{d'}$, respectively, we check in time $O(d^2)$ is there is a black edge between a pair $x \in C_j, y \in C_{j'}$ with $\gamma_j(x) \cap \gamma_{j'}(y) \neq \emptyset$. If there is no such edge, we add the corresponding union profile γ to the profile set of C, i.e., $\gamma(z) = \gamma_a(u) \cup \gamma_b(v)$, and $\gamma(x) = \gamma_j(x)$ if $x \neq z$ and $x \in C_j$. This finishes the description of the algorithm.

Correctness and running time. The correctness comes from the invariant that every red component is associated to its set of profiles. Indeed a black edge between $xy \in E(G_i)$ means that there is a biclique between $x(G)$ and $y(G)$, thus these sets cannot use a shared color. The running time is as indicated since

$$\prod_{j \in [d']} \#\text{profile}(C_j) \leqslant \prod_{j \in [d']} (2^q - 1)^{|C_j|} = (2^q - 1)^{\sum\limits_{j \in [d']} |C_j|} = (2^q - 1)^{d+1},$$

where $\#\text{profile}(C_j)$ is the number of profiles of red component C_j. To actually compute a coloring, one can simply augment profiles with one representative coloring.

Advantages. Assuming the SETH,[6] this new approach will for instance not improve the theoretically best algorithm for q-Coloring parameterized by clique-width, since Lampis showed that running time $O^*((2^q - 2)^{\text{cw}})$ is achievable and essentially optimal [19]. However our algorithm presents some practical advantages.

The first remarkable feature is its simplicity. Contrary to dynamic programming on clique-width expressions which has to deal with unions, joins, and relabelings (or tree-decompositions with their forget, introduce, and join internal nodes), we have only one operation to handle: the contraction of two vertices, where all optimization efforts can be invested. We have only $n - 1$ operations in total, while tree-decompositions and clique-width parse trees typically have $O(n)$ nodes, incurring a multiplicative overhead.

We do not maintain partial solutions that turn out to be locally infeasible. When a red component C has at least one profile, we know that $G[C(G)]$ is q-colorable. On the contrary, in the usual algorithm parameterized by clique-width, a join between two labels sharing at least one color can happen long after the corresponding vertices were introduced. This causes to maintain a lot of unnecessary partial solutions.

4 Oriented twin-width

Oriented twin-width is "fairer" than twin-width in the following sense: In the partition sequence, when merging two parts X, Y of $\mathcal{P}_{i+1}$ to form $\mathcal{P}_i$, the only red arcs of D_i which are created are directed from $X \cup Y$. Indeed, if Z is homogeneous to X and to Y, it is

[6] For Strong Exponential Time Hypothesis; the assumption that for every $\varepsilon > 0$, there is an integer k such that n-variable k-SAT cannot be solved in time $(2 - \varepsilon)^n$ by a classical algorithm.

also homogeneous to $X \cup Y$. Thus any error due to a contraction is only attributed to the contracted vertices and does not wildly spread to their neighbors. This locality of error makes one's life much easier to design partition sequences. Let us illustrate why.

Given a graph G and two non necessarily adjacent vertices x, y, we denote by $G/\{x,y\}$ the graph obtained by contracting x, y into one vertex $\{x,y\}$ and joining it to all neighbors of x and y in G. We say that a class $\mathcal{C}$ of graphs is *d-contractible* if for every graph G of $\mathcal{C}$ there are two vertices x, y such that $G/\{x,y\}$ is also in $\mathcal{C}$ and is such that the degree of $\{x,y\}$ is at most d. For instance the following lemma due to Norine et al. [25] implies that K_t-minor free graphs are $2^{\tilde{O}(t)}$-contractible.

▶ **Lemma 14** (Lemma 2.2. in [25]). *Let G be a K_t-minor free graph. Then there are two vertices $u, v \in V(G)$, both of degree $2^{\tilde{O}(t)}$, that are either false twins or adjacent.*

Moreover, by a direct application of the discharging method, Kotzig [18] could show that planar graphs are 9-contractible (and the bound is attained by the so-called *stellated icosahedron*).

▶ **Lemma 15.** *Every d-contractible class of graphs $\mathcal{C}$ has oriented twin-width at most d.*

Proof. Let G be a graph on n vertices in $\mathcal{C}$ and x, y two vertices such that $G/\{x,y\}$ is in $\mathcal{C}$ and $\{x,y\}$ has degree at most d. To start the partition sequence, consider $\mathcal{P}_{n-1}$ consisting of singletons and part $\{x,y\}$. Note that the only red arcs created by the contraction stem from $\{x,y\}$, yielding out-degree at most d in D_{n-1} (recall Section 2.2). We inductively iterate the argument on $G/\{x,y\}$ to form a partition sequence in which every vertex in D_i has out-degree at most d. ◀

In particular, K_t-minor free graphs have oriented twin-width $2^{\tilde{O}(t)}$, and planar graphs have oriented twin-width at most 9.

Is bounded oriented twin-width a new notion? Surprisingly, the answer turns out to be negative. It is quite fortunate since it allows for more flexibility when looking for contraction sequences. One may just worry about the red out-degree. In contrast, with the above easy arguments the original proof that proper minor-closed classes have bounded twin-width [6] is quite tedious, involving a carefully chosen depth-first-search tree. Up to our knowledge, no classic result on minor-closed classes directly implies bounded twin-width. Moreover, the known upper bound on the twin-width of planar graphs is very large.

▶ **Theorem 16.** *Oriented twin-width and twin-width are functionally equivalent.*

Proof. We already observed that a class with twin-width d has a fortiori oriented twin-width as most d. Moreover mixed value and twin-width are functionally equivalent for graphs by Theorem 6. Thus to show that

1. twin-width,
2. oriented twin-width,
3. mixed value

are pairwise functionally equivalent, we need to argue that Item 2 implies Item 3. Actually this is similar to the proof that Item 1 implies Item 3 presented in [6]. We reproduce the arguments here for completeness.

We show the contrapositive. Let G be a graph with mixed value greater than $2d+2$, hence such that every adjacency matrix of G has a $2d+2$-mixed minor. Fix a partition sequence $\mathcal{S} = P_n, \dots, P_1$ of G. Let σ be a vertex ordering compatible with $\mathcal{S}$. Let $\mathcal{D} = (\mathcal{R} = \{R_1, \dots, R_{2d+2}\}, \mathcal{C} = \{C_1, \dots, C_{2d+2}\})$ be a $2d+2$-mixed minor of $M := \mathrm{Adj}_\sigma(G)$. By design, the partition sequence $\mathcal{S}$ defines a symmetric division sequence over M since when merging two subsets of vertices, one can contract (simultaneously) the corresponding columns and the corresponding rows.

Recall that the vertices of the red directed graphs D_i are subsets of vertices of G. Let ℓ be the maximum index such that a vertex of D_ℓ fully contains a part P of $\mathcal{D}$. Without loss of generality, we may assume that P is a column part, thus $P = C_j$ for some $j \in [2d+2]$. As there is a corner in every cell $M[R_i, C_j]$ with $i \in [2d+2]$, there is in particular at least one *row* r_i in each R_i such that $M[r_i, C_j]$ contains two distinct values. In D_ℓ, the $d+1$ vertices $v_1, v_3, \ldots, v_{2d+1}$ respectively corresponding to rows $r_1, r_3, \ldots, r_{2d+1}$ are all in different parts, except possibly one pair v_{2h-1}, v_{2h+1}. Indeed as we performed a symmetric division sequence on M, and stopped the first time a part of the $2d+2$-mixed minor $\mathcal{D}$ was contained in a part of the sequence, there is at most one part R_h which is contained in a part of P_ℓ. (One may observe that $\mathcal{D}$ need not be symmetric, so h is not necessarily equal to j.) Thus the vertex of D_ℓ corresponding to C_j is the source of at least d red arcs. Therefore G has oriented twin-width at least d. ◀

Note that our proof of Theorem 16 shows that $\mathrm{otww}(\mathcal{C}) \leqslant \mathrm{tww}(\mathcal{C}) \leqslant \exp(\exp(O(\mathrm{otww}(\mathcal{C}))))$. It would be interesting to improve the bound given by the second inequality and/or to complement it by a lower bound. As a consequence, d-contractible classes have twin-width $2^{2^{O(d)}}$, and K_t-minor free graphs have twin-width $2^{2^{2^{\tilde{O}(t)}}}$.

5 Partial contraction sequences to a target class

In this section, we present a couple of FO model-checking algorithms based on partial contraction sequences. It consists of pipelining the algorithm of [6] with other elements of the meta-algorithmic toolbox.

Partial sequences. For two non-negative integers $d, \varDelta$, let $\mathcal{D}_{d,\varDelta}$ be the class of graphs admitting a partial d-sequence to a trigraph of total degree at most $\varDelta$. A class $\mathcal{C}$ is said to be *collapsible to bounded degree* if there are two integers $d, \varDelta$ such that C is included in $\mathcal{D}_{d,\varDelta}$. For a non-negative integer d and a non-decreasing function $f : \mathbb{N} \to \mathbb{N}$, let $\mathcal{E}_{d,f}$ be the class of graphs admitting a partial d-sequence to a trigraph whose total graph has expansion bounded by f. We refer the reader to Section 2.4 for the definition of expansion. A class $\mathcal{C}$ is said *collapsible to bounded expansion* if there is an integer d and a function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{C}$ is included in $\mathcal{E}_{d,f}$. Similarly we may say that a class $\mathcal{C}$ is *collapsible to class* $\mathcal{C}'$ if there is an integer d such that every graph $G \in \mathcal{C}$ has a partial d-sequence to a trigraph whose total graph is in $\mathcal{C}'$.

The FO model checking algorithm in [6]. We will not need a full description of the algorithm. It is enough to recall the following. Let $G_n, \ldots, G_1$ be the contraction sequence of G, and $\mathcal{P}_n, \ldots, \mathcal{P}_1$ the corresponding partition sequence. Like the algorithm presented in this paper for MSO model checking in Section 3.2, we maintain the *local theory* of sentences of quantifier depth q rooted at each vertex u of each trigraph G_i of the sequence. In the case of bounded component twin-width, the *local theory* was naturally limited to the red component of u. Now that the red graphs can have arbitrary large components, the *local theory* is limited to vertices at distance less than 3^q from u in the red graph of G_i. Since the red degree is assumed to be bounded by d, this represents a set, say, $S_{q,d}(u)$ of less than d^{3^q} vertices.

In [6] the *local theory* is not materialized by types but by a tree (called *reduced morphism-tree*), denoted here by $\mathcal{T}_{q,d}(u)$, of depth q and total size function of q only, containing all the *possible games* in the partitioned graph $(G, \mathcal{P}_i)[\bigcup_{v \in S_{q,d}(u)} v(G)]$ up to *equivalent moves*. More precisely, the root of $\mathcal{T}_{q,d}(u)$ is labeled by the empty sequence, and every child adds a new vertex of $\bigcup_{v \in S_{q,d}(u)} v(G)$ (new move) to the current sequence (branch from the root to the current node). At this stage, it is not determined yet if a move is played by $\exists$ or $\forall$ player. One can define by induction what two *equivalent moves* are. At the level of leaves (depth q) two equivalent moves are siblings defining the same induced substructures in $(G, \mathcal{P}_i)[\bigcup_{v \in S_q(u)} v(G)]$ (with equality). Then two sibling internal nodes are equivalent if there is a bijection between their children such that the paired children would be equivalent if they had the same parent.

Initially, the tree $\mathcal{T}_{q,d}(v)$ for each vertex $v \in V(G_n) = V(G)$ is easy to compute: it is a path of length q where the only possible move is v. The tree $\mathcal{T}_{q,d}(u)$ when u is the unique vertex of G_1 is enough to decide $G \models \varphi$ for every sentence φ with quantifier depth k. Indeed such sentence can be effectively rewritten as a prenex[7] sentence of depth $q := \text{Tower}(k + \log^* k + 3)$ [27, Theorem 2.2 and inequalities (32)]. As usual the crux of the algorithm is how to update the trees $\mathcal{T}_{q,d}(u)$ after one contraction is performed. Per tree, this takes time function of q and d only, while at most d^{3^q} trees may require an update after one contraction; hence the overall running time of $f(d, q)n$ for some computable function f. We will however not need to detail how the update is done.

Algorithms based on partial sequences. We first observe that we can pipeline the FO model checking algorithm on graphs given with a (complete) $O(1)$-sequence, developed in [6], with Gaifman's locality theorem. Thus, given a corresponding partial sequence, FO model checking is FPT on collapsible classes to bounded degree. We recall Gaifman's locality theorem.

▶ **Theorem 17** (Gaifman's Locality Theorem [15]). *Every FO sentence φ is equivalent to a Boolean combination of sentences of the form*

$$\exists x_1 \ldots \exists x_k \bigwedge_{1 \leqslant i < j \leqslant k} d(x_i, x_j) > 2r \wedge \bigwedge_{1 \leqslant i \leqslant k} \phi(x_i)$$

*where ϕ is an r-*local formula*, i.e., $\forall \mathscr{A}, a$: $\mathscr{A} \models \phi(a)$ if and only if $\mathscr{A}[N^r_{\mathscr{A}}[a]] \models \phi(a)$, where $N^r_{\mathscr{A}}[a]$ is the r-neighborhood of a, that is, the set of elements at distance at most r of a in the Gaifman graph of $\mathscr{A}$.*

Furthermore if the sentence φ has quantifier depth k, then the formulas ϕ have quantifier depth at most $q = f(k)$ for a computable function f.

In the previous statement, $d(x_i, x_j) > 2r$ is a short-hand for the fact that there is no path of length at most $2r$ between x_i and x_j.

▶ **Theorem 18.** *Given $G \in \mathcal{D}_{d,\Delta}$ with a partial sequence $G = G_n, \ldots, G_s$ such that G_s has total degree Δ, and a sentence $\varphi \in FO_E[k]$, one can decide whether $G \models \varphi$ in time $f(d, \Delta, k)n$ for some computable function f.*

Proof. We run the algorithm of [6] on the partial sequence $G_n, \ldots, G_s$ with a small nuance. When the total degree of a vertex $u \in V(G_i)$ becomes at most Δ, we turn all its black incident edges into red. We denote this new trigraph G'_i and proceed to the next contraction on G'_i (not G_i). By that process, the red degree may exceed d but remains bounded by $d + \Delta$ (in the extreme case when Δ black edges were turned red). We thus maintain trees

[7] with all the quantifiers as a prefix of the sentence, followed by a quantifier-free formula

$\mathcal{T}_{q,d+\Delta}(u)$, with q function of k as given in Theorem 17. When we reach the trigraph G'_s, by design all its edges are red, since its total degree is at most Δ. We may therefore interpret G'_s as a mere graph. Up to this point the algorithm takes time $g(d,\Delta,q)(n-s)$ for some computable function g.

To apply Gaifman's locality theorem directly, we adopt the partition viewpoint on the contraction sequence. Recall that there is a partial partition sequence $\mathcal{P}_n, \ldots, \mathcal{P}_s$ corresponding to the partial trigraph sequence $G_n, \ldots, G_s$. We consider the structure $\mathscr{A} := (G, \mathcal{P}_s, D := \{ab \ : \ a \in u(G), b \in v(G), uv \in R(G'_s)\})$. We add the "dummy" graph D so that the Gaifman graph of $\mathscr{A}$ is simply a blow-up of the Gaifman graph of G'_s. We apply Theorem 17 with $r = 3^q$ on $\mathrm{FO}_{E,\sim,D}$ sentences that are *not* using the relation D. The tree $\mathcal{T}_{q,d+\Delta}(u)$ for every $u \in V(G'_s)$ allows us to determine every such r-local sentence $\phi(x)$ of quantifier depth at most q. We can therefore mark the vertices $u \in V(G'_s)$ such that $\phi(a)$ holds for at least one vertex $a \in u(G)$. This takes time linear in s. We conclude as in the FO model-checking algorithm for bounded-degree structures of Seese [29], by observing that finding a $2r$-scattered set of size k (i.e., k vertices pairwise at distance more than $2r$) among the marked vertices in the graph G'_s can be done in time $h(k,r)s$ for some computable function h (with a bounded search tree). Hence the overall running time. ◀

Typical graphs collapsible to bounded degree –but of unbounded twin-width and unbounded degree– are blow-ups (replace every vertex by a clique module of arbitrary size) of bounded-degree graphs; and more generally any modular decomposition where all the modules have bounded twin-width, while the core has bounded degree. We believe that Theorem 18 should hold more generally for collapsible classes to bounded expansion. We will only show the result for *existential* first-order sentences.

▶ **Theorem 19.** *Given $G \in \mathcal{E}_{d,g}$ with a partial sequence $G = G_n, \ldots, G_s$ such that the total graph of G_s has expansion g, and a sentence $\varphi = \exists x_1 \exists x_2 \ldots \exists x_q \psi \in \exists FO_E[q]$ with ψ a quantifier-free formula, one can decide whether $G \models \varphi$ in time $f(d,g,q)n$ for some computable function f.*

Proof. We run the algorithm of [6] on the partial sequence $G_n, \ldots, G_s$. When we reach G_s, we compute a *low tree-depth cover* $X_1, \ldots, X_h$ of the total graph G'_s of G_s with parameters $h, f = f(\mathcal{E}_{d,g})$ such that, we recall, $G'_s[X_j]$ has tree-depth at most q for every $j \in [h]$, every subset of $V(G'_s)$ of size at most q is fully contained in at least one X_j, and $h = f(k)$ Theorem 7. As $G'_s[X_j]$ has tree-depth at most q for every $j \in [h]$, it has twin-width bounded by a function of q, and a (complete) $f'(q)$-sequence can be found in polynomial time for some function f'. For every $j \in [h]$, we perform the following *run*. We trim all the trees $\mathcal{T}_{q,d}(u)$ with $u \in V(G_s)$ by deleting every move which is not in X_j (and its subtree). We resume the algorithm of [3] with the *trigraph* $G_s[X_j]$ and the twin-width bound set to $f'(q)$.

If $G \models \varphi$ indeed holds, let X_j be such that $G'_s[X_j]$ contains q vertices $(v_1, \ldots, v_q)$ such that $G \models \psi(v_1, \ldots, v_q)$ (recall that φ is existential). The corresponding runs detects a solution. If $G \models \varphi$ does not hold, every run is negative. The claimed overall running time is easy to derive. ◀

Theorems 18 and 19 tackle more general graph classes than the FO model-checking algorithm of [3], combining the features of graphs with bounded twin-width and of sparse graphs. The interest of such algorithms might also be to ease the computation of the (partial) sequence. It is still unknown if there is an approximation algorithm outputting $f(d)$-sequences for graphs of twin-width at most d, in say, fixed-parameter time. A reason why this may be a delicate issue is that $\Omega(\log n)$-subdivisions of n-vertex graphs have bounded twin-width, while

$o(\log n)$-subdivisions do not [3]. Furthermore the $O(1)$-sequences for $\Theta(\log n)$-subdivisions require arguments related to optimal sorting networks, which are difficult to conciliate with typical obstructions (like brambles, tangles, or grid minors) usually enabling to approximate width parameters. Now we observe that $\Omega(\log n)$-subdivisions are *not* an issue to compute partial $O(1)$-sequences to bounded degree or bounded expansion. Indeed there is a relatively easy argument for the former case (not involving sorting networks), whereas the latter case is immediate, the expansion being already bounded.

6 Spanning twin-width

Let $\preccurlyeq$ be a partial order on a set X. When $x \preccurlyeq y$, we say that x is an *ancestor* of y, that y is a *descendant* of x, and that x, y are *comparable.* A *forest order* $\prec$ on X is a partial order such that whenever $x, y \preccurlyeq z$, then x, y are comparable. A *tree order* is a forest order with a minimum element. We write $x \prec^c y$ when $x \prec y$ and there is no z such that $x \prec z \prec y$. The binary relation $(X, \prec^c)$ is the *Hasse diagram* of $\preccurlyeq$. Notice that the Hasse diagram of a tree order is a tree.

Let G be a connected graph. A tree order $\preccurlyeq$ on $V(G)$ is *compatible with* G if uv is an edge of G whenever $u \prec^c v$. Put in another way, a tree order compatible with G can be seen as the transitive closure of some oriented rooted spanning tree of G.

Adding a tree order to a graph can help to design partition sequences. For instance, the key ingredient in the proof of [6] that minor-closed classes have bounded twin-width lies in the fact that if G is K_t-minor free, then it has a tree order $\preccurlyeq$ (a kind of Lex-DFS) such that $(G, \preccurlyeq)$ seen as a binary multirelation has bounded twin-width. This is one appealing feature of twin-width: One often has to guess which additional information will guide the sequence. Here the tree order can be seen as an intermediate step between the mere graph G and the full partition sequence.

In order to refine the landscape between bounded tree-width and bounded twin-width, a natural candidate is the *spanning twin-width* of a connected graph G which is the minimum twin-width of $(G, \preccurlyeq)$ taken over all tree orders $\preccurlyeq$ compatible with G. We extend the notion to disconnected graphs by taking the maximum over all connected components. Observe that this parameter is not monotone. Indeed the spanning twin-width of a subgraph (even induced) can be larger than the one of the host graph. Nevertheless for monotone classes, it exactly captures classes excluding a minor.

▶ **Theorem 20.** *A monotone graph class $\mathcal{C}$ has bounded spanning twin-width if and only if it does not contain some K_t as a minor.*

Proof. As already mentioned, the backward implication is proved in [6] using a Lex-DFS. To show the forward direction, we exhibit graphs in $\mathcal{C}$ with arbitrarily high spanning twin-width. Here an *induced subdivision* of a graph H is any graph obtained from arbitrarily subdividing the edges of H (including the possibility of not subdividing the edge and just keeping it).

▷ Claim 21. If a monotone class $\mathcal{C}$ contains arbitrarily large clique minors, then $\mathcal{C}$ contains an induced subdivision of every cubic graph.

Proof of the Claim: To see this, observe that if H is a cubic graph on t vertices and G contains a K_t minor, then G contains a minor which contains H as a subgraph. Consider then an edge-minimal subgraph G' of G which contains a minor which contains H as a subgraph and observe now that G' is exactly an induced subdivision of H. ◊

The strategy is now to show that if an induced subdivision S of a (connected) cubic graph H is given together with a tree order $\preccurlyeq$, then one can retrieve H from $(S, \preccurlyeq)$ using a first-order transduction[8] via some fixed length formula. Since FO-transductions keep twin-width bounded [6], the assumption that $\mathcal{C}$ has bounded spanning twin-width would directly imply that the class of cubic graphs has also bounded twin-width, which is false [3].

We only have to show how to recover H from $(S, \preccurlyeq)$. FO-transductions allow to introduce a bounded number of unary relations, so we can first identify the set V of vertices of S which have degree three (which are the original vertices of H). The technical task is now to decide if two such vertices $x, y \in V$ are joined or not in H. This is the case if and only if there is an induced path $x = u_0, \dots, u_\ell = y$ of S such that all internal vertices are not in V. We denote this path by P. Note that when $\ell = 1$ querying the existence of such a path can be directly done by asking if xy is an edge of S. Let us focus on the case when we want to retrieve an edge xy which has been subdivided at least once, i.e., let us see how the tree order can help to retrieve the edge xy when $\ell > 1$.

As $\preccurlyeq$ is a tree order compatible with S, xy is an edge of H which is subdivided if and only if one of the following holds for P:

1. $u_0 \prec^c \cdots \prec^c u_\ell$,
2. $u_0 \succ^c \cdots \succ^c u_\ell$, or
3. there exists $i \in [\ell - 1]$ such that $u_0 \prec^c \cdots \prec^c u_i$ and $u_{i+1} \succ^c \cdots \succ^c u_\ell$.
4. there exists $i \in [\ell - 1]$ such that $u_i \prec^c \cdots \prec^c u_0$ and $u_i \prec^c \cdots \prec^c u_\ell$ (exceptional case when u_i is the root of the tree-order).
5. there exists $i \in [\ell - 1]$ such that $u_i \prec^c \cdots \prec^c u_0$ and $u_\ell \prec^c \cdots \prec^c u_{i+1}$ (exceptional case when u_i is the root of the tree-order).
6. there exists $i \in [\ell - 1]$ such that $u_i \prec^c \cdots \prec^c u_\ell$ and $u_0 \prec^c \cdots \prec^c u_{i-1}$ (exceptional case when u_i is the root of the tree-order).

Since all these conditions can be tested with a (long but bounded) first-order formula, H is a first-order transduction of $(S, \prec)$. ◀

An interesting direction would be to investigate which hereditary classes have bounded spanning twin-width as it could indicate some possible generalization of minor-closed classes to the dense setting. But even sparse hereditary classes $\mathcal{C}$ with bounded spanning twin-width are somewhat mysterious. For instance the induced subgraphs of the grid with diagonals have bounded spanning twin-width (and arbitrarily large clique minors).

We believe that subdivisions of arbitrary cubic graphs could be the key in this study and that, for instance, any hereditary class with girth at least five avoiding any subdivision of a fixed cubic graph could have bounded twin-width (and maybe bounded spanning twin-width). In that direction, we suggest the following problem as a first step.

▶ **Conjecture 22.** *The class of segment graphs with girth at least five has bounded spanning twin-width.*

We do not even know if this class has bounded twin-width.

References

1 Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. `doi:10.1145/174644.174650`.

[8] that is, by means of a reinterpretation of the edge set with an FO formula with two free variables, and based on the old edge set and non-deterministic unary relations

2 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

3 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1977–1996, 2021. doi:10.1137/1.9781611976465.118.

4 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width III: Max Independent Set, Min Dominating Set, and Coloring. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.35.

5 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Toruńczyk. Twin-width IV: ordered graphs and matrices. *CoRR*, abs/2102.03117, 2021. URL: https://arxiv.org/abs/2102.03117, arXiv:2102.03117.

6 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 601–612. IEEE, 2020. doi:10.1109/FOCS46700.2020.00062.

7 Édouard Bonnet, Jaroslav Nesetril, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations. *CoRR*, abs/2102.06880, 2021. URL: https://arxiv.org/abs/2102.06880, arXiv:2102.06880.

8 Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Boolean-width of graphs. *Theoretical Computer Science*, 412(39):5187–5204, 2011. URL: https://www.sciencedirect.com/science/article/pii/S030439751100418X, doi:https://doi.org/10.1016/j.tcs.2011.05.022.

9 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12 – 75, 1990. URL: http://www.sciencedirect.com/science/article/pii/089054019090043H, doi:https://doi.org/10.1016/0890-5401(90)90043-H.

10 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2012. URL: http://www.cambridge.org/fr/knowledge/isbn/item5758776/?site_locale=fr_FR.

11 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

12 Marek Cygan, Fedor V Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 4. Springer, 2015.

13 David Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.*, 3(3):1–27, 1999. doi:10.7155/jgaa.00014.

14 Solomon Feferman and Robert L Vaught. The first order properties of products of algebraic systems. *Journal of Symbolic Logic*, 32(2), 1967.

15 Haim Gaifman. On local and non-local properties. In *Studies in Logic and the Foundations of Mathematics*, volume 107, pages 105–135. Elsevier, 1982.

16 Robert Ganian and Petr Hlinený. On parse trees and myhill-nerode-type tools for handling graphs of bounded rank-width. *Discret. Appl. Math.*, 158(7):851–867, 2010. doi:10.1016/j.dam.2009.10.018.

17 Frank Gurski and Egon Wanke. The tree-width of clique-width bounded graphs without $K_{n,\ n}$. In Ulrik Brandes and Dorothea Wagner, editors, *Graph-Theoretic Concepts in Computer Science, 26th International Workshop, WG 2000, Konstanz, Germany, June 15-17, 2000,*

Proceedings, volume 1928 of *Lecture Notes in Computer Science*, pages 196–205. Springer, 2000. doi:10.1007/3-540-40064-8_19.

18 Anton Kotzig. Contribution to the theory of eulerian polyhedra. *Math. Slovaca*, 5:101–113, 1955.

19 Michael Lampis. Finer tight bounds for coloring on clique-width. *SIAM J. Discret. Math.*, 34(3):1538–1558, 2020. doi:10.1137/19M1280326.

20 Alexander Langer, Peter Rossmanith, and Somnath Sikdar. Linear-time algorithms for graphs of bounded rankwidth: A fresh look using game theory - (extended abstract). In Mitsunori Ogihara and Jun Tarui, editors, *Theory and Applications of Models of Computation - 8th Annual Conference, TAMC 2011, Tokyo, Japan, May 23-25, 2011. Proceedings*, volume 6648 of *Lecture Notes in Computer Science*, pages 505–516. Springer, 2011. doi:10.1007/978-3-642-20877-5_49.

21 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. URL: http://www.cs.toronto.edu/%7Elibkin/fmt, doi:10.1007/978-3-662-07003-1.

22 Jaroslav Nesetril and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *Eur. J. Comb.*, 27(6):1022–1041, 2006. doi:10.1016/j.ejc.2005.01.010.

23 Jaroslav Nesetril and Patrice Ossona de Mendez. On low tree-depth decompositions. *Graphs Comb.*, 31(6):1941–1963, 2015. doi:10.1007/s00373-015-1569-7.

24 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

25 Serguei Norine, Paul D. Seymour, Robin Thomas, and Paul Wollan. Proper minor-closed families are small. *J. Comb. Theory, Ser. B*, 96(5):754–757, 2006. doi:10.1016/j.jctb.2006.01.006.

26 Sang-il Oum. Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms*, 5(1):10:1–10:20, 2008. doi:10.1145/1435375.1435385.

27 Oleg Pikhurko and Oleg Verbitsky. Logical complexity of graphs: a survey. *Model theoretic methods in finite combinatorics*, 558:129–179, 2011.

28 Serge A. Plotkin, Satish Rao, and Warren D. Smith. Shallow excluded minors and improved graph decompositions. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia, USA*, pages 462–470, 1994. URL: http://dl.acm.org/citation.cfm?id=314464.314625.

29 Detlef Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.

30 Hisao Tamaki. Positive-instance driven dynamic programming for treewidth. *J. Comb. Optim.*, 37(4):1283–1311, 2019. doi:10.1007/s10878-018-0353-z.