HABILITATION À DIRIGER DES RECHERCHES
de l'École Normale Supérieure de Lyon

Présentée le 19 avril 2024
par **Édouard Bonnet**

Twin-Width and Contraction Sequences

Rapporteurs :

Thomas Colcombet . . . . . DR CNRS, Université Paris Cité
David Eppstein . . . . . Professeur, Université de Californie, Irvine
Frédéric Havet . . . . . DR CNRS, Université Nice–Sophia-Antipolis

Examinateur·rices :

Angela Bonifati . . . . . Professeur, Université Lyon 1
Claire Mathieu . . . . . DR CNRS, Université Paris Cité
Bruno Salvy . . . . . DR INRIA, ENS de Lyon

Table of Contents

Acknowledgments

I am extremely grateful to Thomas Colcombet, David Eppstein, and Frédéric Havet for their review of the current manuscript, and to Angela Bonifati, Claire Mathieu, and Bruno Salvy for completing my habilitation jury. I crucially benefited from the help and guidance of Bora and Russ (and his infinite patience) to make this habilitation happen. A warm thank to them.

In the decade following my PhD defense, I have been fortunate to meet and work with inspirational and elevating people. I keep an excellent memory of my two years in Budapest, mentored by Dániel Marx. I learned a lot from Dániel, but also from every single fellow postdoc I overlapped with. At the very least, I should mention my frequent collaborators Till Miltzow and Paweł Rzążewski, but my thoughts extend to every one of them. I thank my second postdoc mentor, Panos Giannopoulos, for our fruitful collaboration and for introducing me to the computational geometry community.

Since 2018, I have enjoyed an ideal work environment at the LIP of ENS Lyon. The lab owes a lot to the excellent administrative staff headed by Marie B. I am personally indebted to her, Laure, Marie N., and Chiraz. I obviously want to thank all the members of our MC2 team for the last six years (or less if you arrived after me, or left). Nicolas and Stéphan, thank you for mentoring me these past years, scientifically or otherwise, and the entertainment value that your dynamic duo effortlessly generates. Rémi, thank you for these six years of world-class officemate-ness, and our shared research and organizational projects. Eunjung, you visited enough to be considered a member of our team, so there you go, this is where you get acknowledged. Carl, thank you for our discussions in and outside work. Jean-Florent, welcome! Our team would not be the same without people not obsessing over graphs, at least not always. This negation overload is meant to thank Michaël, Natacha, Nicolas S., Omar, Pascal, etc. Of course, so far I have omitted an essential part of my colleagues, those who come and go sometimes in the blink of an eye: our brilliant master, PhD students, and postdocs.

Amadeus, Colin, Dibyayan, Hugues, Julien, Kristóf, Pierre, Robert, Romain, Ugo, it was/is/will be great to have worked/work with you. I also have a thought for our Lyonese colleagues at LIRIS, Aline, Laurent, Nicolas B., Théo, etc. for the work they do to tie our local community together.

As the current thesis is about twin-width, I wish to thank all my coauthors on this topic. If your name already appeared, as an unfair rule to keep the text shorter, it will not reappear in this list (sorry!). Thanks to Florent Foucaud, Noleen Köhler, O-joung Kwon, Tuomo Lehtilä, Raul Lopes, Jaroslav Nešetřil, Patrice Ossona de Mendez, Sebastian Siebertz, Pierre Simon, Romain Tessera, Szymon Toruńczyk, and David Wood. Wait, I haven't thanked Florian Sikora yet? I've had the chance to be hosted in Ljubljana, Warsaw, Daejon, and Liverpool for several research visits. For that I'm grateful to Sergio Cabello, Marcin and Michał Pilipczuk, Sang-il Oum, Viktor Zamaraev, and John Sylvester.

I thank my family for their constant support, notably my parents, sister, brother-in-law, niece, and grandparents. And of course, the amazing Alexandra Covaci.

Nota Bene

Thomas Colcombet's *knowledge* package was used to handle intra-document links. If you encounter a notion you do not know or forgot the definition of, try clicking the corresponding word or symbol. A link should exist and bring you to the line its definition starts. A notable exception is for *twin-width* with its very many occurrences, and which the reader is less likely to forget. After reading the definition, you will want to quickly go back to where you were in the text. There should be a suitable shortcut in your PDF viewer. For instance with Okular, by default, Alt+Shift+Left does the job. Without further ado, I wish you a pleasant reading!

CHAPTER 1

Introduction

1. The origin of twin-width

A little bit over a decade ago, Sylvain Guillemot and Dániel Marx, in a technical and conceptual breakthrough [**51**], answered by the positive the following question: Is there an efficient algorithm to detect, given a small permutation σ and a large permutation τ, whether the permutation matrix of τ contains that of σ as a submatrix? A common name for this problem is Permutation Pattern as we look for *pattern* σ in the *host* permutation τ. Figure 1.1 illustrates this problem with the practical convention, often used in the context of permutation patterns, that the first row of a matrix is displayed as the bottommost. This way, the 1-entry corresponding to i and its image by σ is represented by a black square at position $(\sigma(i), i)$.

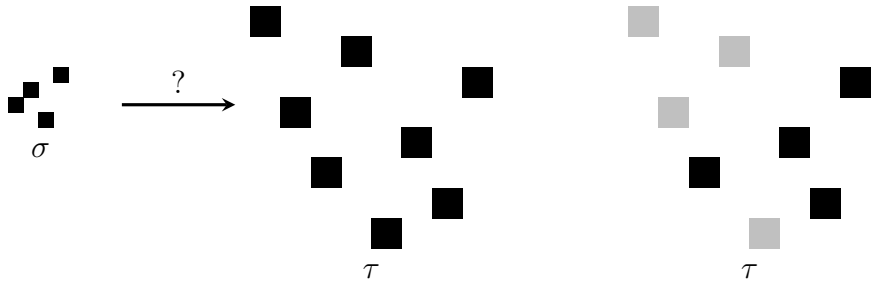


Figure 1.1. Is pattern 3124 in 57362841? Yes, 5**736**2**8**41.

Sure there is an algorithm that solves Permutation Pattern in time $O_\ell(n^\ell)$ when σ has size ℓ, and τ has size n, by simply going through the $\binom{n}{\ell}$ choices of ℓ 1-entries in the permutation matrix of τ, in search for σ. But this is not exactly efficient. Guillemot and Marx designed an algorithm running in time $O_\ell(n)$. This required the introduction of a new kind of permutation decomposition, and its associated width; analogous to, yet different from tree-decompositions and the treewidth of graphs. In a nutshell, Guillemot and Marx's algorithm works as follows. If τ has large width, then it is shown that every permutation

of size ℓ (in particular σ) appears in τ. The algorithm can then answer positively. Otherwise, a decomposition of τ with small width allows for a fast search of σ. If this step is somewhat different from efficient algorithms leveraging tree-decompositions of low width, a parallel can legitimately be drawn.

Guillemot and Marx concludes their introduction observing that "*It would be interesting to see if there is a corresponding graph-theoretic analog for this scheme, which might be useful for solving some graph-theoretical problem.*" Four years ago, with my colleagues Eunjung Kim, Stéphan Thomassé, and Rémi Watrigant, we indeed extended this width from permutations to graphs [**26**], which we dubbed *twin-width*, and together with several collaborators, have been exploring the applications of this new graph invariant. The present thesis is devoted to summarizing these explorations.

Many of these collaborators are or were young researchers (at the time, master student, PhD student, or postdoc) affiliated with or hosted by our lab, the LIP, at ENS Lyon. I wish to salute their essential contributions beyond the "official" acknowledgments section, and list their names: Pierre Bergé, Romain Bourneuf, Dibyayan Chakraborty, Hugues Déprés, Julien Duron, Colin Geniet, Ugo Giocanti, Kristóf Huszár, Noleen Köhler, Raul Lopes, Amadeus Reinald. In addition, we hope to have inspired or encouraged to work on the topic many more, through lectures, talks, or discussions.

2. The Guillemot–Marx permutation width on graphs

This section is not necessary to understand the rest of the document. I chose to include it to substantiate the fact that the twin-width of graphs organically comes from the permutation width of Guillemot and Marx, and to serve as a teaser to their beautiful paper [**51**]. Contraction sequences and twin-width will be defined formally and directly on graphs in Section 6 of Chapter 2.

2.1. The permutation width. The definition Guillemot and Marx give of their permutation width is surprisingly geometric. They first introduce merge sequences of permutations, as follows. Like in Figure 1.1, they see the permutation τ of size n as n axis-parallel filled squares at the positions of the 1-entries in its permutation matrix. For what follows, it is important that pairs of squares project along the axes to disjoint sets. These squares will evolve, more generally, into axis-parallel filled rectangles, in the following way.

A *merge* operation takes an unordered pair of rectangles and fuse them in the minimal axis-parallel rectangle containing their union. Let us fix some labeling of the rectangles. Initially the rectangle (square) at position $(\tau(i), i)$ gets label i for every $i \in [n]$. Every new rectangle gets a fresh label, say, the smallest positive integer that was not a label so far. The merge of rectangles i and j forming rectangle k can be written as $(\{i, j\}, k)$.

As each merge operation decreases the number of rectangles by 1, after $n - 1$ merges, a single rectangle is left. A *merge sequence* of a permutation of size n is thus a sequence of n rectangle sets, one before the first merge, and one after each of the $n - 1$ successive merge operations. See Figure 1.2 for an illustration of a merge sequence of the permutation τ of Figure 1.1.

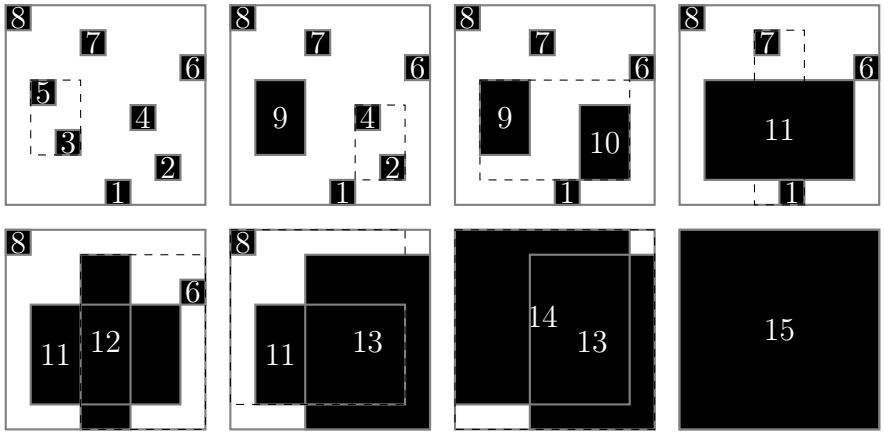


FIGURE 1.2. A merge sequence $(\{3, 5\}, 9), (\{2, 4\}, 10), (\{9, 10\}, 11), (\{1, 7\}, 12), (\{6, 12\}, 13), (\{8, 11\}, 14), (\{13, 14\}, 15)$ of τ.

Given a set $\mathcal{R}$ of axis-parallel rectangles, the *error degree* of a rectangle $R \in \mathcal{R}$ is the number of rectangles in $\mathcal{R} \setminus \{R\}$ whose projection on the x- or y-axis (or both) intersects that of R. For instance, the error degree of rectangle 12 in the bottom-left rectangle set of Figure 1.2 is equal to 2 since the projections on the y-axis of rectangles 6 and 12 intersect, while rectangles 11 and 12 intersect (meaning that both their projections on the x- and y-axis intersect).

The *width* of a merge sequence is then the maximum, taken over every rectangle set $\mathcal{R}$ of the sequence and every rectangle $R \in \mathcal{R}$, of

the error degree of R. Finally the *width*[1] of a permutation τ is the minimum, taken over every merge sequence $\mathcal{S}$ of τ, of the width of $\mathcal{S}$. For example, the width of the merge sequence of Figure 1.2 is equal to 2, as the reader may check that every rectangle of every rectangle set has error degree at most 2. This implies that the width of τ is at most 2. The width of τ is in fact equal to 1, as witnessed by the merge sequence $(\{3,5\},9),(\{2,4\},10),(\{1,10\},11),(\{9,11\},12),(\{6,12\},13),(\{7,13\},14),(\{8,14\},15)$, itself of minimum width.

2.2. Permutation graphs. As we want to extend the permutation width to graphs, it is natural to consider the so-called *permutation graphs* and revisit on them the definition presented in Section 2.1.[2] Every n-element permutation τ generates the following n-vertex permutation graph. For every $i \in [n]$, draw the straight-line segment with endpoints $(i,0)$ and $(\tau(i),1)$ in the real plane. We identify this segment to the 1-entry $(\tau(i),i)$ of the permutation matrix of τ. The permutation graph of τ is then the intersection graph of this collection of n segments, with one vertex per segment, and one edge between every pair of intersecting segments.

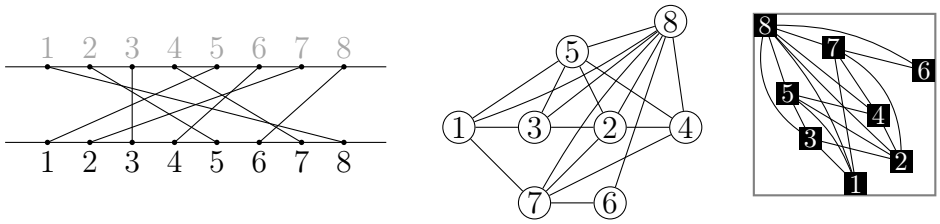


FIGURE 1.3. The graph of the permutation τ of Figure 1.1 (center), the same graph embedded in the matrix of τ (right), and its geometric representation (left).

Note that there is an edge between two 1-entries if and only if they form a decreasing sequence; see right of Figure 1.3. In particular, given two axis-parallel rectangles R and R' whose projections on the axes do not intersect, the adjacency of a pair $x \subseteq R$, $y \subseteq R'$ of initial 1-entries can be unambiguously reconstructed and only depends on R and R'. On the contrary, if R and R' instead have some intersecting projection along an axis, then the adjacencies between pairs $x \subseteq R$, $y \subseteq R'$ now

[1]A reader familiar with [**51**] may notice that the definition given here is slightly different from the original one.

[2]An n-vertex permutation graph need not fully determine a permutation on n elements, but this is inconsequential to our purpose.

depend on the particular choice of x, y and are irremediably lost at this coarser level of representation of the permutation; see Figure 1.4. Note that if, contrary to the drawn configuration, the projection of R on the y-axis would be included in that of R', then the pair of edges and non-edges between R and R' would simply be found by swapping the role of R and R'.

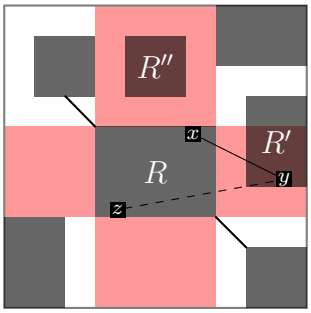


FIGURE 1.4. Rectangles without an intersecting projection with R, i.e., avoiding R and the red regions, have their enclosed 1-entries either fully adjacent or fully non-adjacent to those of R. On the contrary $x \in R, y \in R'$ form an edge, while $z \in R, y \in R'$ do not. The same happens *in columns* with R and R''.

Therefore, the width of Guillemot and Marx extends to graphs as follows. A contraction sequence of an n-vertex graph iteratively contracts (merges, or identifies) pairs of vertices until, after $n-1$ contractions, a single vertex remains. Along the way, vertices correspond to subsets of initial vertices. Two subsets X, Y are linked by a regular edge (represented in black) if every vertex in X is linked to every vertex in Y, or by an error edge (represented in red) if this does not hold but yet there is a vertex in X adjacent to a vertex in Y. The red degree of a vertex/subset is the number of error edges (or red edges) incident to it. The *twin-width* is then the minimum taken over every contraction sequence of the maximum red degree of a vertex/subset appearing in the contraction sequence. A formal definition is given in Section 6 of Chapter 2. Figure 1.5 depicts the contraction sequence corresponding to the merge sequence of Figure 1.2.

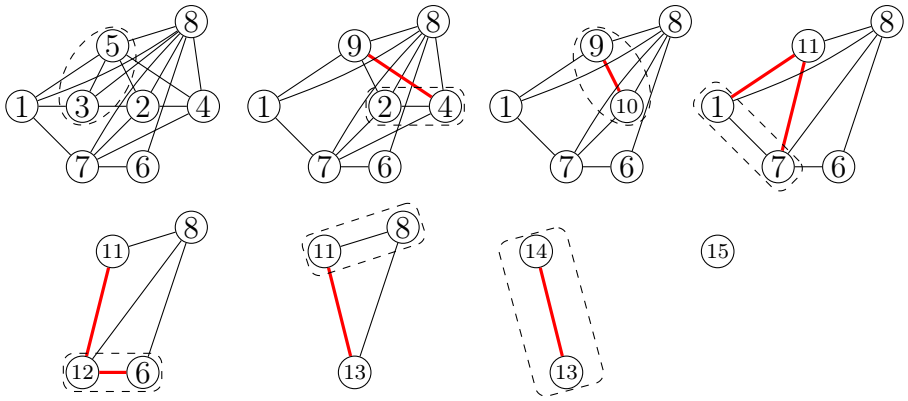


FIGURE 1.5. The merge sequence of Figure 1.2 as the contraction sequence of a graph.

3. Motivations

Graph theorists are drawn by the intrinsic beauty of formidable open questions emerging from the arguably simplest[3] non-trivial language, but also by the numerous connections and applications to the rest of mathematics and computer science, and virtually every other field. An obvious and prolific tie of graph theory is with algorithms. Most computational problems either are directly phrased in terms of graphs, or, in whole or in part, naturally reduce to such problems. The more we understand graphs, the better we are equipped to solve these problems faster and/or more accurately.

Oftentimes, one cannot realistically hope for a uniformly fast algorithm across all graphs. The good news is that typical inputs of real-life graph problems—as one can find in computational biology, motion planning, network science, linguistics, etc.—are not arbitrary and do present some "structure." Algorithmic graph theory (an area of graph theory) aims to detect and exploit that very structure. In this line of work, the most famous example is the inception and use of treewidth.

To a first approximation, the treewidth of a graph is usually said to measure how close it is to a tree. Treewidth was introduced by Bertelè and Brioschi [**11**], rediscovered by Halin [**54**], and popularized under its current name by the monumental work of Robertson and Seymour [**78**]. As a natural follow-up to the paper of Bertelè and

[3]A single binary relation.

Brioschi, many NP-hard problems were shown to admit polynomial-time algorithms when restricted to input graphs of bounded treewidth. Courcelle gave a unified reason for this phenomenon, proving that this indeed holds for any graph problem expressible in monadic second-order logic [**37**]. In interesting developments, treewidth was leveraged to efficiently solve problems within classes of *unbounded* treewidth, such as for instance planar graphs with the *bidimensionality* technique (see for instance [**45**]). The success of treewidth was not limited to algorithm design. For example, the original motivation of Robertson and Seymour for treewidth was the roadmap that eventually led them to solve Wagner's conjecture (that the minor relation is a well-quasi-order on graphs).

While starting to explore the new twin-width parameter, we would want to find similar features: an invariant with a wide scope of uses, both algorithmic and non-algorithmic. The next section surveys some little successes we have had with twin-width. Their exposition is the content of the thesis.

4. Overview of the thesis

Let us take a brief tour of the forthcoming chapters.

4.1. Theoretical base and first properties. The reader will find in Chapter 2 all the basic definitions and notations relevant to the thesis. In Chapter 3 we familiarize ourselves with twin-width by stating and proving useful simple facts on this new notion.

Chapter 4 contains the crux of twin-width theory: a characterization of low twin-width in the simplicity of judiciously-chosen adjacency matrices. This characterization crucially relies on a celebrated theorem in combinatorics, the Marcus–Tardos theorem [**69**], which was also essential to Guillemot and Marx's work [**51**]. We then explore in Chapter 5 how widespread classes of bounded twin-width are. We draw a relatively fine line between graph classes of bounded twin-width and graph classes of unbounded twin-width. In this endeavor, the characterization via adjacency matrices proves particularly useful. Classes of bounded twin-width are surprisingly general, which makes our applications—algorithmic, structural, model-theoretic, and enumerative properties of these classes—satisfying unifications and extensions of facts previously proven in some particular classes of bounded twin-width.

4.2. Other graph parameters via contraction sequences. Before moving on to applications, we take a step back and realize in Chapter 6 that contraction sequences may define other interesting graph parameters. We will see that, incidentally, some of these parameters are functionally equivalent (i.e., bounded on the same graph classes) to well-established graph invariants such as clique-width and linear clique-width. This will allow us in Chapter 7 to present algorithms for graphs of bounded clique-width and for graphs of bounded twin-width in a unified way.

On the other hand, many derived parameters are new, and we give a motivation for their study. A lot more on this topic will appear in the doctoral thesis of my PhD student Julien Duron.

4.3. Algorithmic applications. Chapter 7 presents various algorithms. In large part, this chapter can be thought as addressing the question: *Which problems can be solved faster or approximated better on graphs of low twin-width than on general graphs, and how?* Likely, its acme is a *theoretically efficient* algorithm for any graph problem expressible in first-order logic, when given, in addition to the input graph, a contraction sequence witnessing low twin-width. This has the merit of unifying a lot of the knowledge on this very question. Indeed several classes, individually known to admit such algorithms, have bounded twin-width, and on them good[4] contraction sequences can be efficiently computed. We will also see that several optimization problems can be approximated in graphs of bounded twin-width within better factors than in general graphs. The same caveat that a contraction sequence is required by the approximation algorithms applies.

This caveat may be disappointing since we currently do not know how (if at all possible) to efficiently find good contraction sequences. The tools later developed in Chapter 10 make this achievable for matrices (on which we will also have a twin-width notion), when their row sets and column sets are thought as ordered. We will thus see that we can multiply two $n \times n$ matrices of low twin-width over a finite field in $O(n^2 \log n)$ time. Furthermore if the matrices are given in some compact form occupying only $O(n)$ space (as matrices of bounded twin-width can be), then they can be multiplied in $O(n)$ time; hence with sublinear complexity in their number of entries.

[4]This is how we informally refer to contraction sequences whose width is within a fixed function of the actual twin-width.

We will finally present some algorithms that leverage twin-width but work outside the classes of bounded twin-width, while also *not* requiring an oracle to good contraction sequences.

4.4. First-order logic and twin-width. Chapter 8 explores the connection between twin-width and first-order logic. It features two main results. The first says that every class built from a class of bounded twin-width by means of a first-order "logical reinterpretation," called transduction, itself has bounded twin-width. This showcases that, as far as first-order logic is concerned, twin-width is a robust notion. It also makes, like the characterization of Chapter 4, for a powerful tool to show that a class indeed has bounded twin-width.

The second result is a counterpart to theorems of Colcombet asserting that classes of bounded linear clique-width coincide with first-order transductions of a linear order, and classes of bounded clique-width, with first-order transductions of a *tree order* [**35**], i.e., the ancestor–descendant partial order of some tree. We indeed establish that classes of bounded twin-width are precisely the first-order transductions of proper permutation classes. A *permutation class* is a set of permutations whose matrices are closed under taking submatrices that are permutation matrices, and a *proper permutation class* is one that is not the class of all permutations. This result can in fact be strengthened: There is a single and fixed proper permutation class $\mathcal{P}$ such that every (graph) class of bounded twin-width is a first-order transduction of $\mathcal{P}$. This also nicely echoes that, as we detailed in Sections 1 and 2, twin-width originates in permutations.

4.5. Growth of classes and labeling schemes. In Chapter 9 we show that every class of bounded twin-width has, up to isomorphism, at most single-exponentially in n many n-vertex graphs; property later referred to as single-exponential unlabeled growth or tinyness. This unifies the same fact, proven for particular classes of bounded twin-width, such as graph classes excluding a fixed minor [**13**] or proper permutation classes [**69**]. The latter is a consequence of the abovementioned Marcus–Tardos theorem, known until 2004 as the Stanley–Wilf conjecture.

Bounded twin-width and tinyness are very close notions. It is indeed quite difficult to find a hereditary graph class with single-exponential unlabeled growth but unbounded twin-width. We will see that, for this purpose, one can use a counting argument on the number of monotone classes built by carefully selecting random graphs

in a specific edge-density regime [**20**], or an elaborate construction of random groups [**73**]. Intriguingly, no separation of bounded twin-width and tinyness is known that does not involve a counting argument, but rather exhibits an explicit (deterministic) separating class.

The number of n-vertex graphs in a class is often compared to how compactly its graphs can be encoded. One can further require that the encoding is equitable in storing a small number of bits *for each* vertex. This is formalized in terms of adjacency labeling schemes or universal graphs. Up to algorithmic considerations, these two frameworks are equivalent. In a nutshell, labeling schemes consist of attributing to each vertex a bit string (or label) of minimum size so that the presence of an edge can be reconstructed solely based on the labels of its endpoints. Universal graphs for a class $\mathcal{C}$ are an infinite family of graphs $U_1, U_2, \dots$ such that U_n contains every n-vertex member of $\mathcal{C}$ as an induced subgraph. It can be observed that a class admits labeling schemes of size $f(n)$ if and only if it has universal graphs on $2^{f(n)}$ vertices. We will see that classes of twin-width at most d admit labeling schemes of logarithmic size $O_d(\log n)$, and hence, universal graphs of polynomial size. It is an open question whether the multiplicative factor of $\log n$ can be independent of d, or even equal to $1 + o_d(1)$. But we do rule out this possibility for monotone tiny classes.

4.6. Ordered graphs and matrices. Chapter 10 is devoted to ordered graphs, or more generally, ordered binary structures. This is a perfect setting for twin-width, as we are then able to show the following facts. Good contraction sequences can be found efficiently. Thus on ordered graphs and matrices, algorithms based on twin-width do not require that a contraction sequence is given with the input graph, as discussed when introducing Chapter 7. Furthermore the frontier *bounded twin-width / unbounded twin-width* coincides on hereditary classes of ordered graphs with several notable dividing lines, between:

- classes with an efficient algorithm for any problem defined in first-order logic, and those likely without,
- monadically dependent and monadically independent classes,
- classes whose unlabeled growth is at most single-exponential, and those whose unlabeled growth is at least factorial.

The first item parallels the situation with treewidth and monadic second-order logic (since Courcelle's theorem [**37**] has a similar converse [**80**]), here with twin-width on ordered graphs and first-order logic. The third item shows a "jump" in the growth of hereditary classes of ordered

graphs. This jump was conjectured with a sharp bound by Balogh, Bollobás, and Morris **[6]**, conjecture that our results settle.

CHAPTER 2

Background and Twin-Width

1. Sets, partitions, functions

We denote by $[i, j]$ the set of integers $\{i, i+1, \ldots, j-1, j\}$, and by $[i]$ the set of integers $[1, i]$. If $\mathcal{X}$ is a set of sets, we denote by $\cup\mathcal{X}$ their union. If S is a set, we may denote the power set of S by 2^S, and if further k is a natural number, the set of subsets of S of cardinality k by $\binom{S}{k}$. We denote by $\mathbb{N}$ the set of non-negative integers.

The *finest partition* of a set X is $\{\{x\} : x \in X\}$, whereas the *coarsest partition* of X is $\{X\}$. We say that a partition $\mathcal{P}'$ *refines* $\mathcal{P}$ (or is a *refinement* of $\mathcal{P}$) if every part of $\mathcal{P}'$ is contained in a part of $\mathcal{P}$. In other words, $\mathcal{P}'$ can be obtained by partitioning the parts of $\mathcal{P}$ further. Reciprocally $\mathcal{P}$ *coarsens* $\mathcal{P}'$ (or is a *coarsening* of $\mathcal{P}'$). Partition $\mathcal{P}'$ *s-refines* $\mathcal{P}$ if $\mathcal{P}'$ refines $\mathcal{P}$ and every part of $\mathcal{P}$ contains at most s parts of $\mathcal{P}'$.

We may sometimes denote by $dom(f)$ the domain of a function f. If f, g are two functions with disjoint domains, let $f \uplus g$ be the *joint function* with domain $\mathrm{dom}(f) \cup \mathrm{dom}(g)$, defined as $(f \uplus g)(x) = f(x)$ for every $x \in \mathrm{dom}(f)$, and $(f \uplus g)(x) = g(x)$ for every $x \in \mathrm{dom}(g)$. We say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is *tetrational* if f is upper bounded by a tower of exponentials of polynomial height in its arguments.

2. Graphs

Except if stated otherwise, we consider graphs that are finite, undirected, and *simple*, i.e., without multiple edges or self-loops. If G is a graph, we denote by $V(G)$, respectively $E(G)$, its set of vertices, respectively of edges. The *incidence graph* of G is the graph with bipartition $V(G) \uplus E(G)$ and an edge between $v \in V(G)$ and $e \in E(G)$ whenever v and e are incident, i.e., v is an endpoint of e. In all the below notations with a graph subscript, we may omit this subscript whenever the graph is clear from the context.

2.1. Neighborhood. For $S \subseteq V(G)$, we denote the *open neighborhood* (or simply *neighborhood*) of S by $N_G(S)$, i.e., the set of neighbors of S deprived of S, and the *closed neighborhood* of S by $N_G[S]$, i.e., the set $N_G(S) \cup S$. We simply write $N_G(v)$ for $N_G(\{v\})$, and $N_G[v]$ for $N_G[\{v\}]$. Two distinct vertices u, v such that $N(u) = N(v)$ are called *false twins*, and *true twins* if instead $N[u] = N[v]$. In particular, true twins are adjacent. Two vertices are *twins* if they are false twins or true twins.

An *independent set* is a subset of vertices that are pairwise non-adjacent. A *clique* is a subset of pairwise-adjacent vertices. The *complement* of a graph G, denoted by $\overline{G}$, is the graph $(V(G), \binom{V(G)}{2} \setminus E(G))$. The *degree* $d_G(v)$ of a vertex $v \in V(G)$ is the size of $N_G(v)$, and the *maximum degree* of G, denoted by $\Delta(G)$, is defined as $\max_{v \in V(G)} d_G(v)$. A graph is said *subcubic* if its maximum degree is at most 3, and *cubic* if all its vertices have degree 3.

2.2. Adjacency matrix. If G is an n-vertex graph and $\prec$ is a linear order over $V(G)$, say, $v_1 \prec \ldots \prec v_n$, then $A_{\prec}(G)$ denotes the adjacency matrix of G in the order $\prec$. Thus the entry in the i-th row and j-th column is a 1 if $v_i v_j \in E(G)$, and a 0 otherwise. Two graphs G, G' are *isomorphic* if there are $\prec, \prec'$ linear orders over $V(G), V(G')$, respectively, such that $A_{\prec}(G) = A_{\prec'}(G')$.

For $A, B \subseteq V(G)$, $E_G(A, B)$ denotes the set of edges in $E(G)$ with one endpoint in A and the other one in B. The associated *biadjacency matrix* has one row per vertex of A, one column per vertex of B, and a 1-entry at row i column j if and only if the corresponding vertices are adjacent.

2.3. Containment. A graph H is a *subgraph* of a graph G if H can be obtained from G by deleting vertices and edges. Then G is called a *supergraph* of H. A *spanning subgraph* and a *spanning supergraph* are such that $V(G) = V(H)$. A graph H is an *induced subgraph* of a graph G if H can be obtained from G by deleting vertices (but preserving *all* the edges whose both endpoints are among the kept vertices). We denote by $G[S]$ the subgraph of G induced by S, that is, the graph obtained from G by removing all the vertices outside S. We set $G - S := G[V(G) \setminus S]$, and may simply write $G - v$ when $S = \{v\}$.

An *edge contraction* of two adjacent vertices u, v consists of merging u and v into a single vertex adjacent to $N(\{u, v\})$ (and deleting u and v). A graph H is a *minor* of a graph G if H can be obtained from G after

vertex and edge deletions, and edge contractions. A graph G is said *H-minor-free* if H is not a minor of G.

2.4. Paths and connectedness. A *path* in a graph G is a sequence of distinct vertices $v_1v_2 \dots v_h$ such that for every $i \in [h-1]$, $v_iv_{i+1} \in E(G)$. The *length* of a path in an unweighted graph is simply the number of edges of the path. A *Hamiltonian path* in a graph G is a path of length $|V(G)| - 1$, i.e., visiting every vertex exactly once. A graph is *traceable* if it admits a Hamiltonian path. Given two vertices $u, v \in V(G)$, we denote by $\mathrm{dist}_G(u, v)$, the *distance* between u and v in G, that is, the length of a shortest path between u and v. The *diameter* of a graph is the longest distance between a pair of its vertices.

The *connected components* of a graph G are the inclusion-wise maximal induced subgraphs of G such that every pair of vertices within an induced subgraph are linked by a path. A graph is *connected* if it has a single connected component (itself). By extension, a set $X \subseteq V(G)$ is connected (in G) if $G[X]$ is connected.

We sometimes refer to two dense parameterized graphs: the *complete graph* on t vertices, denoted by K_t, obtained by making adjacent every pair of distinct vertices on t vertices, and the *complete bipartite graph* or *biclique* $K_{t,t}$, with bipartition (A, B) such that $|A| = |B| = t$ obtained by making every vertex of A adjacent to every vertex of B.

2.5. Graph classes. A *graph class* is a collection of graphs closed under isomorphism. A graph class is said *hereditary* if it is closed under taking induced subgraphs, that is, every induced subgraph of a member of the class is itself in the class. Similarly, a graph class is *monotone* or subgraph-closed if it is closed under taking subgraphs. Note that a monotone class is hereditary, but not all hereditary classes are monotone. Finally, a graph class is *minor-closed* if it is closed under taking minors. A *proper minor-closed* class is one that is minor-closed and not equal to the class of all graphs. Thus a proper minor-closed class is *H-minor-free* for at least some graph H, in the sense that all its graphs are H-minor-free.

More generally, we naturally extend graph properties to class properties: a class *is* X if all its graphs are X. A graph class is said *weakly sparse* if there is some finite integer t such that no graph in the class admits $K_{t,t}$ as a subgraph. The *hereditary closure* of a graph class $\mathcal{C}$, denoted by $\mathit{Her}(\mathcal{C})$, is the class of all graphs that are induced subgraphs of members of $\mathcal{C}$. The *subgraph closure* of a graph class $\mathcal{C}$, denoted by $\mathit{Sub}(\mathcal{C})$, is the class of all graphs that are subgraphs of members of $\mathcal{C}$.

2.6. Grids and subdivisions. For k, ℓ two positive integers, the $k \times \ell$ *grid* is the graph on $k\ell$ vertices, say, $v_{i,j}$ with $i \in [k], j \in [\ell]$, such that $v_{i,j}$ and $v_{i',j'}$ are adjacent whenever either $i = i'$ and $|j - j'| = 1$ or $j = j'$ and $|i - i'| = 1$; see Figure 2.1.

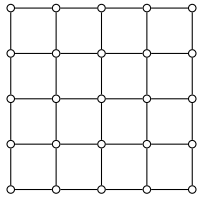


FIGURE 2.1. The 5×5 grid.

A *subdivision* of a graph H is a graph G obtained by replacing each edge of H by a path on at least one edge. If ℓ is a positive integer, the *ℓ-subdivision* of H is obtained by replacing each edge of H by a path on exactly $\ell + 1$ edges. Similarly, a *($\geqslant \ell$)-subdivision* (resp. *($\leqslant \ell$)-subdivision*) is obtained by replacing each edge of H by a path on at least $\ell + 1$ edges (resp. at least one and at most $\ell + 1$ edges). The vertices of H in the subdivision are called *branching vertices*, whereas the new vertices are called *subdivision vertices*.

2.7. Other width parameters. A *tree-decomposition* of a graph G is a pair (T, β) where T is a tree and β is a map from $V(T)$ to $2^{V(G)}$ satisfying the following properties:

- for every $uv \in E(G)$, there is an $x \in V(T)$ such that $\{u, v\} \subseteq \beta(x)$,
- and, for every $v \in V(G)$, the set of nodes $x \in V(T)$ such that $v \in \beta(x)$ induces a non-empty subtree of T.

The *width* of (T, β) is defined as $\max_{x \in V(T)} |\beta(x)| - 1$, and the *treewidth* of G, denoted by $tw(G)$, is the minimum width of (T, β) taken among every tree-decomposition (T, β) of G. The *path-decomposition* and *pathwidth*, $pw(G)$, of a graph G are defined analogously, by further imposing that T is a path.

Most of the graph widths are defined within the same framework of tree layouts. Denoting by $L(T)$ the set of leaves of a rooted tree T, a *tree layout* of a graph G is a pair (T, γ) where T is a rooted binary tree, and γ is a bijective mapping from $L(T)$ to $V(G)$. Every edge $e \in E(T)$ corresponds to the *bipartition* (X_e, Y_e) of $V(G)$ defined by the two subsets of $L(T)$ within the connected components of T deprived of e. For any function f from graphs G vertex bipartitioned by (X, Y) to the natural or real numbers, satisfying $f(G, X, Y) = f(G, Y, X)$, one

can define the graph invariant *f-width*(G) as the minimum over every tree layout of G of the maximum of $f(G, X_e, Y_e)$ over every $e \in E(T)$.

For example, if $f(G, X, Y)$ is the rank of the biadjacency matrix of $E_G(X, Y)$ over the binary field $\mathbb{F}_2$, then f-width(G) is called the *rank-width* of G, and denoted by *rw*(G). If $f(G, X, Y)$ is the base-2 logarithm of the number of distinct neighborhoods in Y (resp. in X) of subsets of vertices in X (resp. in Y), then f-width(G) is called the *boolean-width* of G, denoted by *boolw*(G). The *linear f-width*(G) is defined similarly to f-width(G) except the rooted tree T is required to be a *comb*, i.e., a binary tree whose internal nodes form a path.

We recall the definition of clique-width for the curious reader, although it will not really be needed. This notion introduces (non-necessarily proper) colorings of a graph, i.e., maps from its vertex set to the natural numbers. A graph and its coloring may be called *colored graph*. The *clique-width* of a graph G, denoted by *cw*(G), is the least number of colors needed to form a coloring of G from colored single-vertex graphs, with the following operations:

- making the disjoint union of two colored graphs,
- recoloring every vertex colored i with color j, for some i, j, and
- making adjacent every vertex colored i with every vertex colored j, for some $i \neq j$.

The operations form a tree structure called *clique-width expression*. The *linear clique-width* is defined similarly but restricts the *disjoint union* operation to be between two graphs one of which has a single vertex.

2.8. Partial order on parameters. A graph parameter p is *functionally bounded* by a graph parameter q (on a class $\mathcal{C}$), denoted by $p \sqsubseteq q$ (resp. $p \sqsubseteq_{\mathcal{C}} q$), if there is a function f such that for every graph G (resp. for every graph $G \in \mathcal{C}$), $p(G) \leqslant f(q(G))$. Parameters p, q are *functionally equivalent* or tied (on $\mathcal{C}$) if $p \sqsubseteq q$ and $q \sqsubseteq p$ (resp. $p \sqsubseteq_{\mathcal{C}} q$ and $q \sqsubseteq_{\mathcal{C}} p$), denoted by $p \equiv q$ (resp. $p \equiv_{\mathcal{C}} q$). We finally denote by $p \sqsubset q$ the fact that $p \sqsubseteq q$ holds, but $q \sqsubseteq p$ does not.

The parameters rank-width, boolean-width, and clique-width are all functionally equivalent,[1] whereas treewidth is functionally bounded by rank-width. Similarly, linear rank-width, linear boolean-width, and linear clique-width are functionally equivalent, whereas pathwidth is functionally bounded by linear rank-width. Furthermore, a class has

[1]Hence, for our intents and purposes, the reader is welcome to pick among these three the width they are most comfortable with.

bounded treewidth (resp. bounded pathwidth) if and only if it weakly sparse and has bounded rank-width, or boolean-width, or clique-width (resp. bounded linear rank-width, or linear boolean-width, or linear clique-width) [**52**]. Thus rw $\sqsubset$ tw but rw $\equiv_{\mathcal{C}}$ tw for any weakly sparse class $\mathcal{C}$.

3. First-order and monadic second-order logic

A finite *relational signature* or *vocabulary* is a set $\Sigma = \{R_1, \ldots, R_h\}$ of *relation symbols* given with their arity, with $\mathrm{ar}(R_i) \in \mathbb{N}$ denoting the arity of R_i. A *first-order formula* φ over Σ, also called Σ-*formula*, is any string generated from letter ψ by the grammar:

$$\psi \to \ (\psi),\ \neg\psi,\ \psi \vee \psi,\ \psi \wedge \psi,\ \exists x\psi,\ \forall x\psi,\ R_i(x, \ldots, x),\ x = x, \text{ and}$$
$$x \to x_1, x_2, \ldots \text{ an infinite set of fresh variable labels.}$$

We denote by $\mathrm{FO}(\Sigma)$ the set of first-order Σ-formulas.

Variables under the scope of a quantifier are called *quantified*. Variables that are not quantified are called *free*. We usually denote by $\varphi(x_{f_1}, \ldots, x_{f_h})$ a formula whose free variables are precisely $x_{f_1}, \ldots, x_{f_h}$. A formula without quantified variables is said *quantifier-free*. A *sentence* is a formula without free variables.

A *relational structure* $\mathcal{M}$ (Σ-*structure*, or *structure* for short) over vocabulary Σ specifies a *domain of discourse* D for the variables, and a relation $R_i^{\mathcal{M}} \subseteq D^{\mathrm{ar}(R_i)}$ for each symbol R_i. It is a *binary structure* if all the relation symbols of Σ have arity at most 2. It is said *finite* if the domain D is finite. A sentence φ interpreted by $\mathcal{M}$ is *true*, denoted by $\mathcal{M} \models \varphi$, if it evaluates to true with the usual semantics for $=$, quantified Boolean logic, and $R_i(d_1, \ldots, d_{\mathrm{ar}(R_i)})$ being true whenever $(d_1, \ldots, d_{\mathrm{ar}(R_i)}) \in R_i^{\mathcal{M}}$. A *model* $\mathcal{M}$ of a sentence φ is a structure such that φ is true in $\mathcal{M}$.

A *monadic second-order formula* (or MSO formula for short) over Σ is a generalization of a first-order formula, where one can additionally quantify over unary relations, that is, over subsets of the domain. Second-order variables are written in capital letters to distinguish them from first-order variables. We denote by $\mathrm{MSO}(\Sigma)$ the set of monadic second-order Σ-formulas.

The *quantifier rank* of a formula φ is its maximum number of nested quantifiers. We denote it by $qr(\varphi)$.

For example,

$$\forall x \forall y\, (\neg E(x, y) \vee \exists z (E(z, x) \wedge E(z, y)))$$

is a first-order sentence over $\Sigma = \{E\}$ with $\mathrm{ar}(E) = 2$, which has quantifier rank equal to 3. Among simple undirected graphs, it says that every edge (xy) is part of a triangle (formed together with z). A possible model of this sentence is the *diamond*, i.e., K_4 minus an edge.

Let us also give an example of an MSO formula:

$$\varphi(S, x) := \exists T(\forall y(T(y) \to S(y)) \wedge (E(x, y) \leftrightarrow T(y))).$$

Formula $\varphi \in \mathrm{MSO}(\Sigma)$ has two free variables, one *set variable* S and one *vertex variable* x, and $\mathrm{qr}(\varphi) = 2$. The formula is satisfied by pairs S, x such that the neighborhood of x is included in S. Indeed, φ asks for a subset T of S that is the set of neighbors of x.

4. Model checking

In the *first-order model checking* problem over signature Σ, given a first-order sentence $\varphi \in \mathrm{FO}(\Sigma)$ and a finite Σ-structure $\mathcal{M}$, one has to decide whether $\mathcal{M} \models \varphi$ holds. The input size is $|\varphi| + |\mathcal{M}|$, the number of bits necessary to encode the sentence φ and the model $\mathcal{M}$. The brute-force algorithm decides $\mathcal{M} \models \varphi$ in time $|\mathcal{M}|^{O(|\varphi|)}$, by building the tree of all possible variable assignments.

As $|\varphi|$ is often small compared to $|\mathcal{M}|$, we wish to find an algorithm with running time $f(|\varphi|)|\mathcal{M}|^{O(1)}$, for some computable function f, where the exponent of $|\mathcal{M}|$ (ideally equal to 1) does not depend on the sentence φ. In the language of parameterized complexity, such an algorithm is called *fixed-parameter tractable* (for parameter $|\varphi|$).

FO(Σ) Model Checking	**Parameter:** $\lvert\varphi\rvert$
Input: A Σ-structure $\mathcal{M}$ and a sentence φ of FO(Σ).	
Question: Does $\mathcal{M} \models \varphi$ hold?	

We will mostly consider first-order model checking on binary structures over a finite domain, for which twin-width will be defined. Let us give a simple example with $\Sigma = \{E\}$ and $\mathrm{ar}(E) = 2$. If φ is the sentence

$$\exists x_1 \exists x_2 \cdots \exists x_k \bigwedge_{i<j} \neg(x_i = x_j) \wedge \bigwedge_{i \neq j} \neg E(x_i, x_j),$$

and G is a Σ-structure, then $G \models \varphi$ holds if and if (the underlying undirected graph) G has an independent set of size k. Parameterized complexity theory asserts that this particular problem is highly unlikely to admit an algorithm with running time $f(k)n^{O(1)}$ on n-vertex graphs,

for any function f. However, it does admit such algorithms on restricted inputs, such as planar graphs or bounded-degree graphs.

The *monadic second-order model checking* problem is defined analogously to first-order model checking with monadic second-order sentences instead.

5. Interpretations, transductions, and dependence

We survey here the relevant notions from model theory.

5.1. Interpretations. Let Σ and Γ be two relational signatures. A *simple interpretation* I (or *interpretation* for short) from Σ to Γ consists of the following Σ-formulas: a domain formula $\nu(x)$, and for each relation symbol $R \in \Gamma$, a formula $\varphi_R(x_1, \dots, x_{\mathrm{ar}(R)})$. If $\mathcal{M}$ is a Σ-structure with domain D, the Γ-structure $\mathsf{I}(\mathcal{M})$ has domain

$$\nu(\mathcal{M}) = \{a \in D \ : \ \mathcal{M} \models \nu(a)\}$$

and for every symbol $R \in \Gamma$, relation

$$R^{\mathsf{I}(\mathcal{M})} = \{(a_1, \dots, a_{\mathrm{ar}(R)}) \in \nu(\mathcal{M})^{\mathrm{ar}(R)} : \mathcal{M} \models \varphi_R(a_1, \dots, a_{\mathrm{ar}(R)})\}.$$

If $\mathcal{C}$ is a class of Σ-structures, we denote $\{\mathsf{I}(\mathcal{M}) \ : \ \mathcal{M} \in \mathcal{C}\}$ by $\mathsf{I}(\mathcal{C})$. If all the formulas ν and φ_R are in $\mathrm{FO}(\Sigma)$ (resp. $\mathrm{MSO}(\Sigma)$), we say that I is a first-order (resp. monadic second-order) interpretation. A class $\mathcal{C}$ *interprets* a class $\mathcal{D}$ if there is an interpretation I such that $\mathsf{I}(\mathcal{C}) \supseteq \mathcal{D}$.

Let us see some example on graphs since it is our main focus; hence with $\Sigma = \Gamma = \{E\}$ and $\mathrm{ar}(E) = 2$. If $\nu(x)$ is the formula

$$\exists y \exists z \ \neg(y = z) \wedge E(x, y) \wedge E(x, z)$$

and $\varphi(x, y)$ is the formula

$$\forall z \exists t \ (z = t \vee E(z, t)) \wedge ((E(x, t) \wedge \neg E(y, t)) \vee (\neg E(x, t) \wedge E(y, t))),$$

then on every finite, simple, undirected graph G, the interpretation I redefines the edge set as the pairs x, y whose private neighbors (i.e., adjacent to exactly one of x, y) form a dominating set (i.e., a set whose closed neighborhood is the whole domain), and then delete all the vertices with degree at most 1 in G.

5.2. Transductions. Let $\Sigma \subseteq \Sigma^+$ be relational signatures. The Σ-*reduct* of a Σ^+-structure $\mathcal{M}$ is the structure obtained from $\mathcal{M}$ by deleting all the relations not in Σ. We denote this interpretation as Reduct_Σ. A *monadic lift* of a class $\mathcal{C}$ of Σ-structures is a class $\mathcal{C}^+$ of Σ^+-structures, where Σ^+ is the union of Σ and a set of unary relation

symbols, and $\mathcal{C} = \{\mathsf{Reduct}_\Sigma(\mathcal{M}) : \mathcal{M} \in \mathcal{C}^+\}$. A class $\mathcal{C}$ of Σ-structures *transduces* a class $\mathcal{D}$ if a monadic lift of $\mathcal{C}$ interprets $\mathcal{D}$.

We may also call *transduction* the corresponding monadic lift and interpretation. If not specified, a transduction or interpretation uses first-order logic. It can be noted that transductions and interpretations can be composed. Thus, if $\mathcal{C}$ interprets (resp. transduces) $\mathcal{D}$, and $\mathcal{D}$ interprets (resp. transduces) $\mathcal{E}$, then $\mathcal{C}$ interprets (resp. transduces) $\mathcal{E}$.

As an example, let us see that the class $\mathcal{S}$ of the 1-subdivisions of all complete graphs first-order transduces the class of all graphs. Let $\mathcal{S}^+$ be the monadic lift of $\mathcal{S}$ with two extra unary relations U_1, U_2 such that in every $\{E, U_1, U_2\}$-structure $\mathcal{M}$ of $\mathcal{S}^+$, $\mathcal{M} \models U_1(v)$ if and only if v is a branching vertex in $\mathsf{Reduct}_{\{E\}}(\mathcal{M})$, and for every $G \in \mathcal{S}$ and every subset X of its subdivision vertices, there is $\mathcal{M} \in \mathcal{S}^+$ such that $\mathsf{Reduct}_{\{E\}}(\mathcal{M}) = G$ and $\mathcal{M} \models U_2(v)$ if and only if $v \in X$. We then choose the following interpretation I. The domain formula is simply $U_1(x)$, and $\varphi_E(x, y)$ is $\neg(x = y) \wedge \exists z\ U_2(z) \wedge E(x, z) \wedge E(y, z)$. One can observe that $\mathsf{Reduct}_{\{E\}}(\mathsf{I}(\mathcal{S}^+))$ is the class of all graphs.

Another good exercise is to show that the class of all grids MSO transduces the class of all bipartite graphs, thus the class of all graphs, and then deduce that the class of planar graphs MSO interprets the class of all graphs.

5.3. Dependence and monadic dependence. Let $\varphi(\overline{x}, \overline{y})$ be a Σ-formula and let $\mathcal{C}$ be a class of Σ-structures. Note that the overline indicates that $\overline{x}$ and $\overline{y}$ are tuples of variables, whose length is denoted by $|\overline{x}|$ and $|\overline{y}|$. The formula φ is *independent* over $\mathcal{C}$ if for every binary relation $R \subseteq A \times B$ between two finite sets A and B there exists a Σ-structure $\mathcal{M} \in \mathcal{C}$ on domain D, some tuples $(\overline{u}_a)_{a \in A}$ in $D^{|\overline{x}|}$, and $(\overline{v}_b)_{b \in B}$ in $D^{|\overline{y}|}$ such that

$$\mathcal{M} \models \varphi(\overline{u}_a, \overline{v}_b) \Leftrightarrow R(a, b) \qquad \text{for all } a \in A \text{ and } b \in B.$$

By extension, the class $\mathcal{C}$ is independent if there is a Σ-formula $\varphi(\overline{x}, \overline{y})$ that is independent over $\mathcal{C}$. Otherwise, the class $\mathcal{C}$ is said *dependent*. A class $\mathcal{C}$ of Σ-structures is *monadically dependent* if every monadic lift of $\mathcal{C}$ is dependent, and *monadically independent* otherwise. Dependent classes are also called NIP (for Not the Independence Property).

As a result of a theorem by Baldwin and Shelah [**5**], a class is monadically independent if and only if it transduces the class of all graphs. Besides, it was proven by Braunfeld and Laskowski that among hereditary classes, dependence and monadic dependence coincide [**31**].

6. Contraction or partition sequences, and twin-width

One can equivalently define twin-width via sequences of trigraphs or of partitioned graphs. We give both definitions as they provide complementary viewpoints.

6.1. Trigraphs and contraction sequences. A *trigraph* G has vertex set $V(G)$ and two disjoint edge sets: $E(G)$, its set of *black edges*, and $R(G)$, its set of *red edges*. The *red graph* $\mathcal{R}(G)$ of a trigraph G is the graph $(V(G), R(G))$ obtained by removing its black edges. Similarly the *black graph* $\mathcal{B}(G)$ of a trigraph G is the graph $(V(G), E(G))$ obtained by removing its red edges. The *total graph* $\mathcal{T}(G)$ of G is the graph $(V(G), E(G) \cup R(G))$.

A *red neighbor* (*black neighbor*, *neighbor*, respectively) of $v \in V(G)$ in a trigraph G is any neighbor of v in $\mathcal{R}(G)$ (in $\mathcal{B}(G)$, in $\mathcal{T}(G)$, respectively). Similarly to graphs, the *subtrigraph* of G induced by $S \subseteq V(G)$ is denoted by $G[S]$, with

$$V(G[S]) = S,\ E(G[S]) = E(G) \cap \binom{S}{2},\ \text{and } R(G[S]) = R(G) \cap \binom{S}{2}.$$

An *induced subtrigraph* of G is any trigraph obtained from G by removing vertices. We also write $G - S$ as a short-hand for $G[V(G) \setminus S]$.

A (vertex) *contraction* in a trigraph G consists of merging two (non-necessarily adjacent) vertices, say, $u, v \in V(G)$ into a single vertex with a new label, say w, and updating the trigraph in the following way. The trigraph $G - \{u, v\}$ does not change, and the neighborhood of w is the union X of the neighborhoods of u and v, deprived of $\{u, v\}$. For every $z \in X$, vertex z is a black neighbor of w if and only if $uz \in E(G)$ and $vz \in E(G)$, and it is a red neighbor of w, otherwise.

A *contraction sequence* of an n-vertex graph G is a sequence of trigraphs $G = G_n, \ldots, G_1$ such that G_i is obtained from G_{i+1} by performing one contraction. In particular, G_1 is the 1-vertex trigraph. We number the sequence from n down to 1, for G_i to have exactly i vertices, for each $i \in [n]$. A *d-sequence* is a contraction sequence in which every vertex of every trigraph has at most d red neighbors. In other words, every red graph of the sequence has maximum degree at most d. The *twin-width* of G, denoted by $tww(G)$, is then the minimum integer d such that G admits a d-sequence. Figure 2.2 gives an example of a graph with a 2-sequence, i.e., having twin-width at most 2.

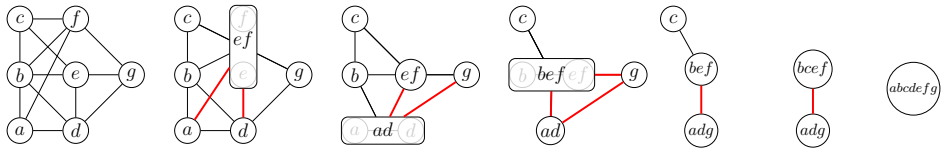


FIGURE 2.2. A 2-sequence witnessing that the initial graph has twin-width at most 2.

We may say that a vertex $u \in V(G_i)$ is *contracted in* a vertex $v \in V(G_j)$ with $j < i$, if the contraction of u with one or several other vertices results in v. The partition viewpoint makes this more transparent.

6.2. Partition sequences. Partition sequences yield an equivalent viewpoint to contraction sequences. A *partition sequence* $\mathcal{P}_n, \dots, \mathcal{P}_1$, of an n-vertex graph is such that $\mathcal{P}_i$ is a partition of $V(G)$ for every $i \in [n]$, $\mathcal{P}_n = \{\{v\} \mid v \in V(G)\}$, and for every $i \in [n-1]$, $\mathcal{P}_i$ is obtained from $\mathcal{P}_{i+1}$ by merging two parts $X, Y \in \mathcal{P}_{i+1}$ into one, $X \cup Y$. In particular $\mathcal{P}_1 = \{V(G)\}$.

The red graph $\mathcal{R}(G_i)$ of G_i, which we may also denote by $\mathcal{R}(\mathcal{P}_i)$, can be redefined as the graph whose vertices are the parts of $\mathcal{P}_i$, and whose edges link two parts $X \neq Y \in \mathcal{P}_i$ whenever there is $u, u' \in X$ and $v, v' \in Y$ such that $uv \in E(G)$ and $u'v' \notin E(G)$. We may call two such parts X, Y *inhomogeneous*. On the contrary, two parts X, Y are *homogeneous* in G when every vertex of X is adjacent to every vertex of Y (corresponding to the black edges), or no vertex of X is adjacent to a vertex of Y. Finally, the *twin-width* of G is the least integer d such that G admits a partition sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ with the property that for every $i \in [n]$, $\mathcal{R}(\mathcal{P}_i)$ has maximum degree at most d.

We denote by $G/\mathcal{P}$ the trigraph defined by the partition $\mathcal{P}$ on G.

6.3. Partial sequences and extension to binary structures. A *partial d-sequence*, be it in the contraction or partition viewpoint, is the same as a d-sequence except its last trigraph has not necessarily a single vertex (resp. its last partition has not necessarily a single part). We also observe that the definition of *twin-width* readily generalizes to trigraphs (resp. partitioned graphs). We may then say that there is a partial d-sequence *from* its first (tri)graph (resp. partitioned graph) *to* its last trigraph (resp. partitioned graph).

We now extend *twin-width* to binary structures over finite relational signatures. For that, the partition viewpoint may be a bit more

convenient. Let $U_1, \dots, U_p$ be the unary relations, and $E_1, \dots, E_q$, the binary relations of such a binary structure G (where we omit the G superscript). We see G as a vertex- and edge-colored graph, and still denote by $V(G)$ its domain. Let $\mathcal{P}_u$ be the partition of $V(G)$ into the equivalence classes of the relation $\sim$ defined by $v \sim v'$ whenever for every $i \in [p]$, $U_i(v) \Leftrightarrow U_i(v')$. That is, $\mathcal{P}_u$ is the coarsest partition each part of which has vertices of the same *unary atomic type*, i.e., satisfying the same unary relations.

The twin-width of G is defined as the least integer d such that there is a partial d-sequence from the finest partition of $\mathsf{Reduct}_{\{E_1,\dots,E_q\}}(G)$ to $\mathcal{P}_u$. We finally need to define the red edges for a structure made of several binary relations, instead of a single one, as the rest of the definition remains the same. Two parts $X \neq Y$ of a partition $\mathcal{P}$ of $V(G)$ are *inhomogeneous*, and linked by a red edge, whenever there is $x, x' \in X$, $y, y' \in Y$ and $i \in [q]$ such that $R_i(x, y)$ holds but $R_i(x', y')$ does not; in other words, if *not* all the pairs of $X \times Y$ have the same binary atomic type.

It is convenient that the contraction/partition sequences of graphs *and* binary structures end on a singleton. Thus a d-sequence for the binary structure G, appends any finishing sequence to the partial d-sequence from the finest partition of $\mathsf{Reduct}_{\{E_1,\dots,E_q\}}(G)$ to $\mathcal{P}_u$. We are not concerned with the *exact* twin-width of binary structures beyond graphs. We will only care if classes of binary structures have bounded or unbounded twin-width. Therefore, we will not discuss if the twin-width of G should rather be d, $d + p$, $d + 2^p$, or any other value in $O_{p,q}(d)$.

We may refer to the *width* of a partition (resp. partition/contraction sequence) on the domain of a binary structure as the maximum degree of its red graph (resp. as the overall maximum degree of its red graphs). Given a vertex of a trigraph or a part of a partitioned graph, we also call *red degree* its degree in the corresponding red graph.

CHAPTER 3

First Properties

The goal of this chapter is threefold. Section 1 is here to familiarize the reader with contraction sequences and twin-width through simple facts. In Sections 2 and 3, we instill two useful alternative viewpoints to contraction sequences: split sequences and twin-decompositions. We conclude the chapter with the technical lemmas of Section 4, which will prove useful in the subsequent chapters.

1. Operations preserving bounded twin-width

The following fact holds since the presence of at least one edge and at least one non-edge between two disjoint subsets X, Y, corresponding to the red edges, is preserved under complementation.

Observation 3.1. *The twin-width of a graph is equal to the twin-width of its complement.*

The next observation holds since for every pair X, Y of homogeneous subsets, and for every $X' \subseteq X, Y' \subseteq Y$, the pair X', Y' is homogeneous.

Observation 3.2. *Let G be a graph (resp. trigraph), and H, an induced subgraph (resp. induced subtrigraph) of G. Then $tww(H) \leqslant tww(G)$.*

Thus the hereditary closure of a class of bounded twin-width has itself bounded twin-width. This is not true for the subgraph closure. For instance, one can remark that every complete graph has twin-width 0, while the subgraph closure of the class of all complete graphs is the class of all graphs. Indeed we will see that the class of all graphs have unbounded twin-width.

The next statement formalizes that turning some red edges of a trigraph into black edges or non-edges also can only decrease its twin-width.

Observation 3.3. *Let G, H be two trigraphs such that $V(G) = V(H)$, $R(H) \subseteq R(G)$, and $E(G) \subseteq E(H) \subseteq E(G) \cup R(G)$. Then $tww(H) \leqslant tww(G)$.*

The *disjoint union* $G \uplus H$ of two graphs G, H simply puts them side by side, while the *complete sum* $G + H$ further adds an edge between every pair $u \in V(G), v \in V(H)$. One can note that $\mathrm{tww}(G \uplus H) = \mathrm{tww}(G + H) = \max(\mathrm{tww}(G), \mathrm{tww}(H))$. Thus *cographs*, which are the graphs constructible from 1-vertex graphs via disjoint unions and complete sums, have twin-width 0. Actually the class of graphs with twin-width 0 is exactly the class of cographs. Here it is convenient to know another characterization of cographs as the graphs excluding P_4, the 4-vertex path, as an induced subgraph. Thus graphs that are not cographs have, by Observation 3.2, at least the twin-width of P_4, which they contain as induced subgraph. Finally as P_4 has no pair of twins, its twin-width is at least 1, and in this case, exactly 1.

Let us now see a wide generalization of the preservation of twin-width by disjoint unions and complete sums. Given two graphs G, H and a vertex $v \in V(G)$, the *substitution* in G of H at v, which we denote by $G[v \leftarrow H]$, is the graph with vertex set $(V(G) \setminus \{v\}) \cup V(H)$ and edge set

$$E(G - v) \cup E(H) \cup \{xy \ : \ x \in N_G(v), y \in V(H)\},$$

formalizing the idea of replacing v by the graph H.

Lemma 3.4. *Let G, H be two non-empty graphs, and $v \in V(G)$. Then, $\mathit{tww}(G[v \leftarrow H]) = \max(\mathit{tww}(G), \mathit{tww}(H))$.*

Proof. First, $\mathrm{tww}(G[v \leftarrow H]) \geqslant \max(\mathrm{tww}(G), \mathrm{tww}(H))$, by Observation 3.2, since G and H are induced subgraphs of $G[v \leftarrow H]$; here it matters that H has at least one vertex. On the other hand, $\mathrm{tww}(G[v \leftarrow H]) \leqslant \max(\mathrm{tww}(G), \mathrm{tww}(H))$ by concatenating a partial $\mathrm{tww}(H)$-sequence from $G[v \leftarrow H]$ to G, with a $\mathrm{tww}(G)$-sequence of G. □

Readers familiar with *modular decompositions* (see for instance **[53]**) will notice that repeated applications of Lemma 3.4 show that the twin-width of a graph is the maximum twin-width of the *modules* and *quotients* occurring in its modular decomposition.

2. Split sequences and proper colorings

In most of the algorithmic applications presented in Chapter 7, one exploits the contraction sequences going forward. However, it may sometimes be helpful to consider the contraction process backward. One starts at a single vertex, and progressively undoes the contractions

until the initial graph is "created." We call this a *split sequence*, and a contraction seen backward, a *split*.

Let us present an example where this seemingly minor perspective shift proves useful. A graph is said *triangle-free* if it does not contain K_3 as a subgraph, or equivalently, as an induced subgraph. A graph is *k-colorable* if each of its vertices can be given one of k colors in such a way that no edge has two endpoints of the same color. The map from the vertex set to the color set is called a *proper k-coloring*. The *chromatic number* of a graph G, denoted by $\chi(G)$, is the least integer k such that G is k-colorable.

There are triangle-free graphs of arbitrarily large chromatic number. Actually many constructions of infinite families of triangle-free graphs with increasing chromatic number have been found, and perhaps the most classic one is the Mycielskian **[70]**. No such construction can however be of bounded twin-width.

Theorem 3.1 (**[21]**). *Every triangle-free graph of twin-width d is $d+2$-colorable.*

Proof. Let G be an n-vertex triangle-free graph of twin-width d. The idea is to follow a split sequence $G_1, \dots, G_n = G$ of width d, and to iteratively find a proper $d+2$-coloring of the total graph $\mathcal{T}(G_i)$ for i going from 1 to n. The total graph $\mathcal{T}(G_1)$ has only one vertex, so admits a proper coloring with one color. The total graph $\mathcal{T}(G_n)$ is equal to G, thus a proper $d+2$-coloring for it would prove the theorem.

We are then left with the task of turning a proper $d+2$-coloring of $\mathcal{T}(G_i)$ into one for $\mathcal{T}(G_{i+1})$, for every $i \in [n-1]$. Let us distinguish two cases.

First case: the vertex $z \in V(G_i)$ that is split into $u, v \in V(G_{i+1})$ has at least one black neighbor, say w, in G_i; see Figure 3.1.

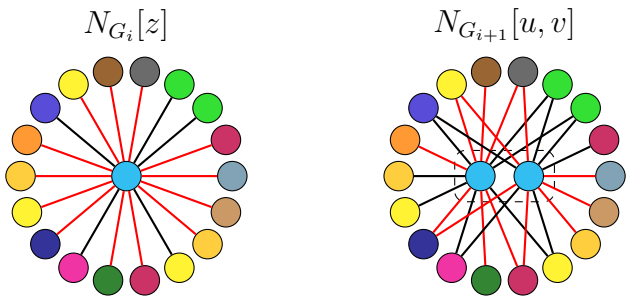


Figure 3.1. Split, when z has a black neighbor.

In this case, the new coloring of $\mathcal{T}(G_{i+1})$ gives the color of z to both u and v, while leaving unchanged the color of the other vertices. Why is this coloring proper? For that we only need to justify that u and v cannot be linked by a red edge or a black edge in G_{i+1}. If the latter edge would exist, then one could find two vertices $u' \in V(G)$ contracted in u, and $v' \in V(G)$ contracted in v, such that $u'v' \in E(G)$. Besides, by definition of contractions in trigraphs, uw and vw are black edges in G_{i+1}. Thus any $w' \in V(G)$ contracted in w is such that $u'w', v'w' \in E(G)$, thus $u'v'w'$ contradicts that G is triangle-free.

Second case: z has no black neighbor. Then z has at most d neighbors in $\mathcal{T}(G_i)$. In that case, u can retain the color of z, and v be given a different color not appearing among the neighbors of z, while the other vertices again keep their color; see Figure 3.2. This is always

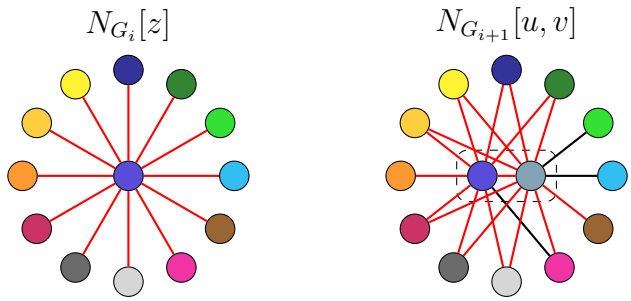


FIGURE 3.2. Split, when z has no black neighbor.

possible with $d+2$ colors. □

Surprisingly, this simple coloring scheme is almost tight, as there are triangle-free graphs of twin-width d that cannot be properly colored with fewer than $d+1$ colors [**14**]. It can also be extended to show that graphs G of twin-width d can be properly colored with at most $(d+2)^{\omega(G)-1}$ colors, where $\omega(G)$ is the *clique number* of G, i.e., the maximum cardinality of a clique of G. A class for which $\chi \sqsubseteq \omega$ is said *χ-bounded*.

Theorem 3.2 ([**21**])**.** *Graph classes of bounded twin-width are χ-bounded.*

As we will see in Chapter 5 that classes of bounded clique-width have bounded twin-width, this extends and simplifies the χ-boundedness of the former classes [**42**]. The function in ω bounding the chromatic number of graphs of bounded twin-width has been improved to quasipolynomial [**75**], and then to polynomial [**30**].

3. Twin-decompositions

There is a vertex-labeled rooted binary tree T that is naturally related to a d-sequence $\mathcal{S}$. It has one leaf per vertex of G, the initial n-vertex graph or binary structure, and one internal node per newly created vertex, having as children the two vertices that were contracted into it. From the mere tree T two elements are missing: one to fully describe $\mathcal{S}$, and one to fully describe G.

The order in which the contractions are done is important, as two distinct orders could very well result in a different value of the overall maximum degree in the corresponding red graphs. We thus augment T with a linear order $<$ on its internal nodes, specifying the order of the contractions. Note that for $<$ to indeed correspond to a contraction sequence, we expect it to be *compatible* with the tree, in the following sense: if y is a strict ancestor of x then $x < y$.

Now $(T, <)$ completely identifies $\mathcal{S}$, but the edges of G are missing. Instead of adding them at the leaves of T, we can condense the information of $E(G)$ into at most $(d+1)(n-1)$ edges added to T. This is better seen with the split sequences of the previous section.

After the split of a vertex z into two vertices u, v, at most $d+1$ black edges may *appear*: one between u and v, and at most one of uw, vw for every red neighbor w of z. Note that if both uw and vw are black edges, we do not count them as appearing, since zw was already a black edge. Every time a black edge appears in the split sequence, we link its endpoints in $V(T)$, which can be leaves or internal nodes, by a special edge, represented in blue in Fig. 3.3. We denote this set of edges added to T by B (for Biclique or Blue). One can observe that B partitions $E(G)$ into at most $(d+1)(n-1)$ bicliques of G, since there are $n-1$ splits. Note that if G was a binary structure with several binary relations, we would have one set B_j for each binary relation E_j.

A *twin-decomposition* is such a triple $(T, <, B)$; see Figure 3.3. Interestingly, a twin-decomposition $(T, <, B)$ of an n-vertex graph G of twin-width d can be a much sparser object than G itself. The graph $(V(T), E(T) \cup B)$ has $2n-1$ vertices and at most $(d+2)(n-1)$ edges, while G can have up to $\binom{n}{2}$ edges. This may allow for algorithms with sublinear complexity in the number of edges of the input, directly operating on twin-decompositions. In Chapters 7 and 10 we will see such examples for finding shortest paths and multiplying matrices. Notably, the sparsity of twin-decompositions will also prove useful in Chapter 8 to the characterization of classes of bounded twin-width as transductions of proper permutation classes.

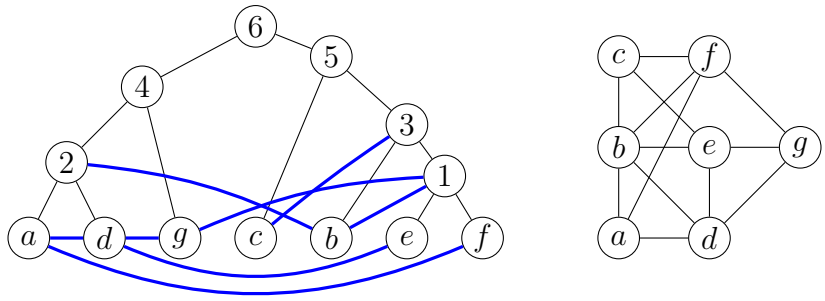


FIGURE 3.3. The twin-decomposition for the contraction sequence of Figure 2.2, next to the initial graph.

4. Technical lemmas

We use the twin-decomposition viewpoint to remark that the vertex set of a graph or binary structure can be ordered in such a way that all the contractions of some contraction sequence of minimum width are performed on "consecutive vertices."

Lemma 3.5. *For every binary structure G, there is a linear order $\prec$ on $V(G)$ such that G has a partition sequence $\mathcal{S}$ of width $tww(G)$ such that every part of every partition of $\mathcal{S}$ is on consecutive vertices along $\prec$.*

PROOF. Choose for $\prec$ the left-to-right order on the leaves of the tree of the twin-decomposition corresponding to $\mathcal{S}$; see Figure 3.3. □

The linear order $\prec$ is itself a binary relation. Note that the intended contraction sequence has the same width on G as it has on the augmented binary structure $(G, \prec)$, as by design no red edge would come from $\prec$. Thus Lemma 3.5 has the following consequence.

Lemma 3.6. *For every binary structure G, there is a linear order $\prec$ on $V(G)$ such that $tww(G) = tww(G, \prec)$.*

The next lemma bounds the width of partitions interpolating between an s-refinement of a partition $\mathcal{P}$ and $\mathcal{P}$ itself. It is more of an observation as any partial partition sequence between the latter two partitions works.

Lemma 3.7. *Let G be a binary structure and $\mathcal{P}, \mathcal{P}'$ be two partitions of $V(G)$, with width d and d', respectively, such that $\mathcal{P}'$ s-refines $\mathcal{P}$. There is a partial d''-sequence from $\mathcal{P}'$ to $\mathcal{P}$ with $d'' \leqslant \max(s(d+1) - 2, d')$.*

A typical use of Lemma 3.7 is when $s = 2$, and $d = d'$. It can then be informally phrased as follows. If, before and after contracting an arbitrary number of *disjoint* pairs of vertices, the width is at most d, then it does not get worse than $2d$ in between, regardless of the order in which the pairs are contracted. Successive applications of Lemma 3.7 can "piece together" point-wise refinements of bounded width of a partition sequence of unknown width.

Lemma 3.8. *Let $\mathcal{P}_n, \dots, \mathcal{P}_1$ be a partition sequence of a binary structure G. Let $\mathcal{P}'_n, \dots, \mathcal{P}'_1 = \{V(G)\}$ be possibly-equal partitions of $V(G)$ of width at most d such that for every $i \in [n]$: $\mathcal{P}'_i$ s-refines $\mathcal{P}_i$, and $i = 1$ or $\mathcal{P}'_i$ refines $\mathcal{P}'_{i-1}$. Then G admits a partition sequence of width at most $s(d+1)$ going through every partition of $\mathcal{P}'_n, \dots, \mathcal{P}'_1$.*

Lemma 3.8 is an essential element of the proof, given in Chapter 8, that first-order transductions preserve *bounded twin-width*. We also use it, in Chapter 4, for the characterization of *bounded twin-width* via adjacency matrices. We now consider how the twin-width may evolve when an *apex* is added, i.e., a new vertex linked arbitrarily to the rest of the graph.

Lemma 3.9. *Let G, H be two graphs such that G is isomorphic to $H - v$ for some $v \in V(H)$. Then $tww(H) \leqslant 2tww(G) + 1$.*

SKETCH. Let $A \subseteq V(H)$ be the set of neighbors of v in H, and set $B := V(H) \setminus (A \cup \{v\})$. For the partition sequence of H, we follow that of G but refuse to mix vertices in A with vertices in B: When the partition in G is $\mathcal{P}$, that in H is $\{\{v\}\} \cup \{P \cap A, P \cap B \ : \ P \in \mathcal{P}\}$ deprived of the possible empty sets. This way $\{v\}$ has no red neighbor, and every other part has at most $2\mathrm{tww}(G) + 1$ red neighbors, by the same argument as in Lemma 3.7. □

A similar proof to that of Lemma 3.9 shows that after adding a unary relation the twin-width at most essentially doubles.

Lemma 3.10. *Let G be a binary structure, and H be a binary structure obtained by adding a unary relation to G. Then $tww(H) \leqslant 2tww(G)+1$.*

Lemma 3.10 is why we will not worry too much about the unary relations of our binary structures. Repeated applications of Lemma 3.10 show that finite monadic lifts of classes of bounded twin-width have bounded twin-width.

CHAPTER 4

Characterization via Adjacency Matrices

We often refer to *the* adjacency matrix of a graph as if this matrix were unique. Figure 4.1 should remind us that a same graph may have very different adjacency matrices. In some way, this chapter quantifies

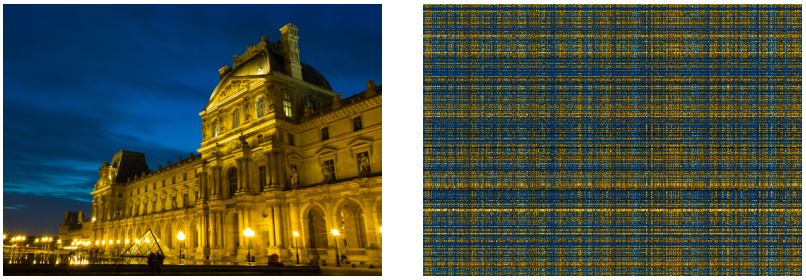


FIGURE 4.1. Not all biadjacency matrices are equal.

how much more structured the matrix on the left is than the one on the right.

1. Grid minors and the Marcus–Tardos theorem

We see the row set $\mathit{rows}(M)$ and the column set $\mathit{columns}(M)$ of a matrix M as naturally ordered. A *division* of a totally ordered set X is a partition of X into *intervals*, i.e., into parts made of consecutive elements. For two positive integers s and t, an (s,t)-division (or simply division) of a matrix M is a pair $(\mathcal{R},\mathcal{C})$ such that $\mathcal{R}$ is a division of $\mathrm{rows}(M)$ into s intervals, and $\mathcal{C}$ is a division of $\mathrm{columns}(M)$ into t intervals. A t-division is short for (t,t)-division. We define *matrix partitions* similarly but without the requirement that the parts must be intervals.

For $X \subseteq \mathrm{rows}(M)$ and $Y \subseteq \mathrm{columns}(M)$, $M[X,Y]$ denotes the submatrix of M obtained by removing the rows that are not in X, and the columns that are not in Y. A *zone* of a partition $(\mathcal{R},\mathcal{C})$ of M is a submatrix $M[R,C]$, with $R \in \mathcal{R}$ and $C \in \mathcal{C}$. We may alternatively use the word *cell* if the matrix partition is a matrix division. Note

that a t-division defines t^2 cells. Given a 0–1 matrix M and a division $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ of M, we define $M/\mathcal{D}$ as the 0–1 matrix with one row per $R \in \mathcal{R}$, one column per $C \in \mathcal{C}$, and a 0-entry at row R, column C if $M[R, C]$ is full 0, and a 1-entry otherwise.

A *t-grid minor* of a matrix M is a t-division $(\mathcal{R}, \mathcal{C})$ of M, every cell of which contains at least one nonzero entry; see left of Figure 4.2. We say that a matrix is *t-grid free* if it does not have a t-grid minor. The celebrated Marcus–Tardos theorem asserts that every 0–1 matrix

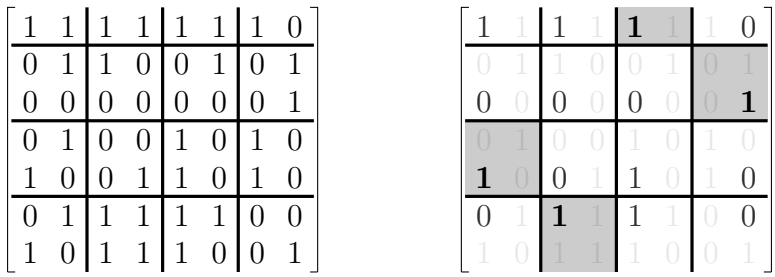


FIGURE 4.2. A 4-grid minor, and how to find in it any 4-element permutation as a pattern, here 2143.

with a large enough constant number of 1-entries in average per row and per column has a t-grid minor. Precisely:

Theorem 4.1 ([**69**]). *For every integer t, there is some number c_t such that every $n \times m$ 0–1 matrix M with at least $c_t \max(n, m)$ 1-entries has a t-grid minor.*

Marcus and Tardos established this theorem with $c_t = 2t^4\binom{t^2}{t}$. Fox [**46**] subsequently improved the bound to $3t2^{8t}$. He also showed that c_t must be exponential in t (at least $2^{\Omega(t^{1/4})}$). Cibulka and Kynčl [**34**] decreased c_t further down to $8/3(t+1)^2 2^{4t}$. We will simply remember that c_t is single-exponential in t.

Theorem 4.1 originated as a question posed by Füredi and Hajnal in 1992 [**47**], later promoted to the *Füredi–Hajnal conjecture*. Klazar [**63**] showed in 2000 that a positive solution to the Füredi–Hajnal conjecture would imply that of another somewhat older conjecture, independently made by Stanley and Wilf, that every proper permutation class has *single-exponential growth*, i.e., for some number c, at most c^n permutations of size n for every integer n.

Klazar's proof is by induction on n, and shows more generally that the Marcus–Tardos theorem implies that the set of square matrices of

any 0–1 matrix class avoiding a fixed permutation matrix as a pattern has single-exponential growth. By *matrix class* we mean a set of matrices closed under taking submatrices. And we say, for two 0–1 matrices M and N, that M *contains the pattern* N if N can be obtained by possibly turning some 1-entries into 0-entries in a submatrix of M; and *avoids* it, otherwise.

Let $\mathcal{S}$ be the set of every square 0–1 matrix avoiding a fixed $t \times t$ permutation matrix, and $n > 1$ be the smallest integer for which the single-exponential growth of $\mathcal{S}$ with basis $c := 15^{2c_t}$ has yet to be established (where c_t is as in Theorem 4.1). Set the $\lceil n/2 \rceil$-division $\mathcal{D}$ with every part of size 2 (but possibly the last one in row and column) on any $n \times n$ matrix M of $\mathcal{S}$. The matrix $M/\mathcal{D}$ (which is also in $\mathcal{S}$) has at most $c_t\lceil n/2 \rceil$ 1-entries since otherwise it would admit a t-grid minor, and thus would not avoid any $t \times t$ permutation matrix as a pattern; see right of Figure 4.3. Forgetting M, every 1-entry in $M/\mathcal{D}$ can be lifted to 15 distinct 0–1 matrices of size 2×2 that are not full 0, whereas every 0-entry in $M/\mathcal{D}$ corresponds to a full-0 matrix. Thus there are at most

$$15^{c_t\lceil n/2 \rceil} c^{\lceil n/2 \rceil} \leqslant c^{\lfloor n/2 \rfloor + \lceil n/2 \rceil} = c^n$$

$n \times n$ matrices in $\mathcal{S}$.

Despite being open for over a decade, the Füredi–Hajnal conjecture was proven by Marcus and Tardos with a short[1] and beautiful proof. We encourage the interested reader to enjoy it in the original paper [**69**]. We nevertheless list some hints that should progressively lead them to the solution:

(1) observe that the question reduces to dealing with square matrices;
(2) try and prove the converse by induction on the matrix size;
(3) divide again the matrix, this time in cells of size $t^2 \times t^2$;
(4) classify the cells into four categories: full-0 cells, *wide* cells with at least t nonzero columns, *tall* cells with at least t nonzero rows, and *light* cells, which are not full 0, nor wide, nor tall;
(5) how many wide (resp. tall) cells can a single column (resp. row) part at most contain without creating a t-grid minor?
(6) using the induction hypothesis, how many cells can be nonzero?
(7) how many 1-entries can at most be in a light cell?
(8) use all the above to bound the total number of 1-entries.

[1]It can be made to fit half a page.

2. Mixed minors and characterization via mixed number

A matrix M is said *vertical* (resp. *horizontal*) if all its rows (resp. columns) are equal. Observe that a 0–1 matrix which is both vertical and horizontal is full 0 or full 1. We say that M is *mixed* if it is not vertical nor horizontal. A *t-mixed minor* in M is a t-division, every cell of which is mixed; see Figure 4.3. A matrix without t-mixed minor is said *t-mixed free*. For instance, the $n \times n$ full-1 matrix is 1-mixed free but admits an n-grid minor.

$$\left[\begin{array}{cc|ccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

FIGURE 4.3. A 3-mixed minor.

The *mixed number*, $\mathrm{mn}(M)$, of a matrix M is the largest integer t such that M admits a t-mixed minor. And the mixed number, $\mathrm{mn}(G)$, of a graph G is the minimum mixed number among adjacency matrices of G. We define the *grid number*, gn, of matrices and graphs analogously with t-grid minor replacing t-mixed minor. These numbers readily extend to binary structures over finite relational signatures. For that, we shall just define the adjacency matrix $A_{\prec}(G)$ of a binary structure G along the order $\prec$, as having in row $u \in V(G)$, column $v \in V(G)$ the *binary atomic type* of the ordered pair (u, v), i.e., any bijective encoding of the set of binary relations R of G such that $(u, v) \in R$.

The following theorem is the cornerstone of twin-width theory.

Theorem 4.2 ([26]). *A class of binary structures has bounded twin-width if and only if it has bounded mixed number.*

With our notations of Chapter 2, it says that $\mathrm{tww} \equiv \mathrm{mn}$, and has two directions. The easy one:

Lemma 4.1. *$\mathrm{mn} \sqsubseteq \mathrm{tww}$.*

PROOF. Let G be any binary structure, and $\prec$ be a vertex ordering of $V(G)$ such that G has a *division sequence*, i.e., a partition sequence each partition of which is a division of $(V(G), \prec)$, $\mathcal{S}$ of width $\mathrm{tww}(G)$.

We saw in Chapter 3 (see Lemma 3.5) that this was always possible. We claim that $\mathrm{mn}(A_\prec(G)) \leqslant 2\mathrm{tww}(G) + 2$. Indeed, let $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ be a mixed minor of $A_\prec(G)$, and $\mathcal{D}'$ be the first division of $\mathcal{S}$ such that a part $P \in \mathcal{D}'$ contains a row part or a column part of $\mathcal{D}$. Say, without loss of generality, that P contains a part of $\mathcal{C}$.

A cell of $\mathcal{D}$ necessarily contains a non-constant row. For each $i \in [h]$ with $h = \lceil |\mathcal{D}|/2 \rceil$, let $P_i \in \mathcal{D}'$ contains a non-constant row in the $2i-1$-st row part of $\mathcal{D}$. By definition of P, all parts P_i are pairwise distinct but possibly one pair P_j, P_{j+1} both equal to P. Hence the red degree of P, thus the twin-width of G, is at least $\frac{1}{2}(\mathrm{mn}(A_\prec(G)) - 2) \geqslant \frac{1}{2}(\mathrm{mn}(G) - 2)$. □

And the harder and more useful one:

Lemma 4.2. *tww $\sqsubseteq$ mn.*

We prefer to give a more quantitative and unpacked statement.

Theorem 4.3 (**[26]**). *There is a function $f : \mathbb{N} \to \mathbb{N}$ with $f(t) = 2^{2^{O(t)}}$ such that every binary structure with a t-mixed free adjacency matrix has twin-width at most $f(t)$.*

SKETCH. Let G be any binary structure with an adjacency matrix $A_\prec(G)$ that is t-mixed free. Starting from the finest partition of $V(G)$, we build a division sequence $\mathcal{S}'$ (relative to $\prec$) by greedily merging pairs of consecutive parts along $\prec$, while no column (or row) part of the corresponding matrix division contains more than $10c_t$ mixed cells, with c_t as in Theorem 4.1. Here the Marcus–Tardos theorem is crucially used to show that this greedy process cannot be stopped (before reaching the coarsest partition) without implying the existence of a t-mixed minor. This is done via the auxiliary 0–1 matrix with 1-entries at positions of mixed cells, and 0-entries elsewhere.

The division sequence $\mathcal{S}'$, say $\mathcal{D}_n, \dots, \mathcal{D}_1$ may have large width, but allows to build a list of coarser and coarser partitions $\mathcal{P}_n, \dots, \mathcal{P}_1$ on $V(G)$, each of width $2^{O(c_t)}$, such that $\mathcal{P}_i$ $2^{O(c_t)}$-refines $\mathcal{D}_i$ for every $i \in [n]$. By Lemma 3.8, we thus get a partition sequence $\mathcal{S}$ of G with width $2^{O(c_t)} \cdot 2^{O(c_t)} = 2^{2^{O(t)}}$. □

A contraction sequence is a fairly complicated object, and it may be challenging to directly find contraction sequences of low width. Theorem 4.3 simplifies the task of bounding the twin-width of binary structures: One *only* needs to order their vertex set such that the

corresponding adjacency matrix is t-mixed free, ideally for some small value t.

A typical use to bound the twin-width of a class $\mathcal{C}$:

(1) find a good vertex-ordering scheme based on the properties of $\mathcal{C}$,
(2) assume that adjacency matrices in this order admit a t-mixed minor,
(3) use this t-mixed minor to contradict the membership to $\mathcal{C}$, and
(4) conclude with Theorem 4.3.

One can observe that if a square matrix has a t-mixed minor, then it also admits a $\lfloor \frac{t}{2} \rfloor$-mixed minor completely above or completely below its main diagonal. This turns out useful as the diagonal of adjacency matrices sometimes behave differently than its surrounding entries; think for instance the diagonal of 0-entries in the adjacency matrix of a complete graph.

Observation 4.3. *Let M be an adjacency matrix along an order $\prec$ that admits a t-mixed minor. Then there are two intervals X, Y along $\prec$, such that $X \prec Y$, and $M[X, Y]$ or $M[Y, X]$ admits a $\lfloor \frac{t}{2} \rfloor$-mixed minor.*

By $X \prec Y$ we simply mean that $x \prec y$ for every $x \in X$ and $y \in Y$. We notice that for symmetric matrices, like adjacency matrices of graphs, *both* $M[X, Y]$ and $M[Y, X]$ admit a $\lfloor \frac{t}{2} \rfloor$-mixed minor.

3. Applications of the characterization

In this section, we show that three classes have bounded twin-width: unit interval graphs, traceable graphs excluding K_t as a minor, and posets of bounded antichain size. These examples are chosen so as to yield simple uses of Theorem 4.3, of increasing difficulty. In Chapter 5, we will see better bounds and more general results. *Unit interval graphs* are the intersection graphs of intervals of length 1 on the real line.

Fact 4.4. *The class of unit interval graphs has bounded twin-width.*

Proof. Let G be a unit interval graph, and let $\prec$ totally order $V(G)$ following the left-to-right ordering of the interval left endpoints in some fixed geometric representation of G. We claim that $A_{\prec}(G)$ is 4-mixed free. By Observation 4.3, we shall just argue that $A_{\prec}(G)$ cannot have a 2-mixed minor above its diagonal.

Indeed, let $X_1 \prec X_2 \prec Y_1 \prec Y_2$ be such that $A_{\prec}(G)[X_1, Y_2]$ has a 1-entry (necessary condition for this cell to be mixed). Thus there is $x \in X_1$ and $y \in Y_2$ whose intervals intersect at least at the right endpoint, say p, of the interval of x. Then every vertex in $X_2 \cup Y_1$ is

represented by an interval containing p, so $X_2 \cup Y_1$ is a clique. This implies that $A_\prec(G)[X_2, Y_1]$ is full 1, thus not mixed. □

As we will see in Section 5, K_t-minor-free graphs have bounded twin-width. Further assuming the existence of a Hamiltonian path, this is a direct consequence of Theorem 4.3.

Fact 4.5. *K_t-minor-free traceable graphs have twin-width $2^{2^{O(t)}}$.*

Proof. We first observe that any spanning supergraph of the biclique $K_{t,t}$ contains the clique K_t as a minor, by performing the edge contractions of any perfect matching of $K_{t,t}$. Let G be a traceable graph. It has a Hamiltonian path $v_1 v_2 \dots v_n$, and we define the linear order $\prec$ as $v_1 \prec v_2 \prec \dots \prec v_n$. We again use Observation 4.3 to simply argue that there is no $X \prec Y$ such that $A_\prec(G)[X, Y]$ has a t-mixed minor. Indeed, let $X_1 \prec X_2 \prec \dots X_t$ partition X, and $Y_1 \prec Y_2 \prec \dots Y_t$ partition Y. Due to the Hamiltonian path, edge contractions can reduce each X_i and each Y_j to single vertices. At least one cell $A_\prec(G)[X_i, Y_j]$ has to be full 0, as otherwise a spanning supergraph of $K_{t,t}$ results from these edge contractions. □

Posets stand for partially ordered sets. Two elements x, y are *comparable* if $x \preceq_P y$ or $x \preceq_P y$, where $\preceq_P$ is the order of poset P. A *chain* in a poset is a set of pairwise comparable elements, whereas an *antichain*, is a set of pairwise incomparable elements. Our encoding of the adjacency matrix $A_\prec(P)$ of a poset P puts a 0-entry at row x, column y if $x = y$, a 1-entry if $x \preceq_P y$, a -1-entry if $y \preceq_P x$, and a 2-entry if x and y are incomparable, where $\prec$ is a linear order on $V(P)$ unrelated to $\preceq_P$.

Theorem 4.4 ([**26**])**.** *Every class of finite posets with bounded antichain size has bounded twin-width.*

Proof. Let P be a poset with a maximum antichain of size t. By Dilworth's theorem (see Galvin's short proof [**50**]), $V(P)$ can be partitioned into t chains $P_1, P_2, \dots, P_t$. We define the linear order $\prec$ on $V(P)$ as $P_1 \prec P_2 \prec \dots \prec P_t$, and for every $x \neq y \in P_i$, $x \prec y$ whenever $x \preceq_P y$.

We show that $A_\prec(P)$ is $6t$-mixed free. By the pigeonhole principle and Observation 4.3, a $6t$-mixed minor would imply the existence of a 3-mixed minor in some $A_\prec(P)[P_i, P_j]$ with $i < j$. We now refute the possibility of such a 3-mixed minor.

Let $R_1 \prec R_2 \prec R_3$ be any partition of P_i, and $C_1 \prec C_2 \prec C_3$ be any partition of P_j. As $P_i \prec P_j$, for the cell $A_\prec(P)[R_2, C_2]$ to be mixed, there should be a comparable pair $x \in R_2, y \in C_2$. If $x \preceq_P y$, then by transitivity $R_1 \preceq_P C_3$, and the cell $A_\prec(P)[R_1, C_3]$ is full 1, hence not mixed. If instead $y \preceq_P x$, then by transitivity $C_1 \preceq_P R_3$, and the cell

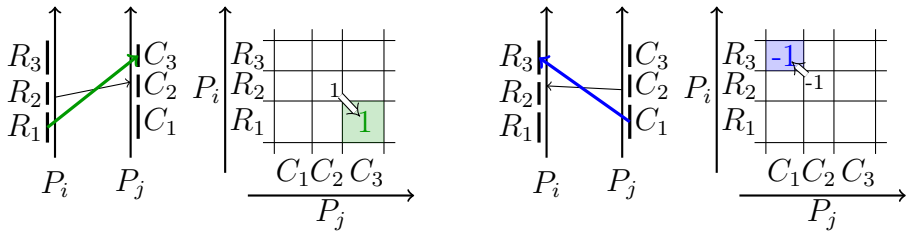


FIGURE 4.4. The two cases when $A_\prec(P)[R_2, C_2]$ is not full 2.

$A_\prec(P)[R_1, C_3]$ is full -1, hence not mixed; see Figure 4.4. □

This shows that posets with no antichain of size $t+1$ have twin-width at most $2^{2^{O(t)}}$. An upper bound of $8t$ has been alternatively achieved **[3]**.

4. Versatile twin-width and balanced sequences

We can extract more from the proof of Theorem 4.3. If the mixed number is low, one can not only find *one* contraction sequence of low width, but exponentially many, with a linear number of choices at each step. We formalize this with the notions of tree of contractions (not to be confused with a twin-decomposition) and versatile twin-width.

A *tree of contractions* T of a trigraph G is a rooted tree, whose root is labeled by G, whose leaves are all labeled by the 1-vertex trigraph, and such that one goes from a parent node to a child node by performing a single contraction operation. In particular, T has depth $|V(G)| - 1$, and a contraction sequence is the particular case when T is a path.

The *versatile twin-width* of a (tri)graph G is the least integer d such that G admits a tree of contractions in which every internal node labeled, say, F has maximum red degree at most d, and at least $\lfloor |V(F)|/d \rfloor$ children, obtained by contracting one of $\lfloor |V(F)|/d \rfloor$ pairwise disjoint pairs of vertices in F. Such a tree is then called a *versatile tree of contractions* of width d.

Theorem 4.5 (**[22]**). *Twin-width and versatile twin-width are functionally equivalent.*

In Chapter 9, we use Theorem 4.5 to design adjacency labeling schemes and bound the growth of classes of bounded twin-width. The latter is leveraged in Chapter 5 as a generic way to establish that some classes have unbounded twin-width.

A d-sequence is said *balanced* if its partitions $\mathcal{P}_n, \dots, \mathcal{P}_1$ are such that for every $i \in [n]$ and $P \in \mathcal{P}_i$, $|P| \leqslant d \cdot \frac{n}{i}$. Note that the partitions are not balanced in the sense that all of their parts are of roughly the same size: There might be small parts (even singletons) coexisting with parts of size $\lfloor d \cdot \frac{n}{i} \rfloor$ in $\mathcal{P}_i$. Rather, no part of $\mathcal{P}_i$ should be larger than the $\frac{n}{i}$ bound (of equitable partitions) by more than a d factor.

Another useful consequence of Theorem 4.5 is that classes of bounded twin-width admit balanced contraction sequences of bounded width. This will be used in Chapter 7 to approximate some combinatorial optimization problems.

Theorem 4.6 ([21]). *There is a function f and a polynomial-time algorithm that inputs a d-sequence of a graph G and outputs a balanced $f(d)$-sequence of G.*

SKETCH. Follow a branch of a versatile tree of contractions that always contracts parts of smallest combined size. □

5. Oriented twin-width

There are three ways vertex subsets X, Y can be inhomogeneous:

- with $y \in Y$ having a neighbor and a non-neighbor in X,
- with $x \in X$ having a neighbor and a non-neighbor in Y,
- with both of the above.

We can distinguish these cases by directing the red graph accordingly. Let us orient the red edge XY from X to Y in the first case, from Y to X in the second, and direct it in both ways in the third. This yields the *directed red graph* of a partition.

The *outdegree* (resp. *indegree*) of a vertex in a directed graph or *digraph* is its number of incident outgoing (resp. ingoing) arcs. We then define the *oriented twin-width* of a graph G, denoted by $\mathit{otww}(G)$, as the minimum, taken over every partition sequence $\mathcal{S}$, of the overall maximum outdegree of the directed red graphs of the partitions in $\mathcal{S}$. We may call *oriented width* the maximum outdegree of the directed red graph of a partition, or overall maximum outdegree of the directed red graphs of a partition sequence.

For every graph G, $\mathrm{otww}(G) \leqslant \mathrm{tww}(G)$ as the outdegree of a part in the directed red graph is at most its degree in the red graph. On the other hand, keeping the *oriented* width low is simpler, as only the just-contracted part can get new outgoing red arcs, possibly increasing the current bound. Note indeed that the other parts can only see their outdegree in the directed red graph decrease. It might even seem as though the oriented twin-width could be bounded on some classes on which the twin-width is not. Surprisingly this is not the case.

Theorem 4.7 ([25]). *Twin-width and oriented twin-width are functionally equivalent.*

Proof. Another surprise perhaps is that we already proved Theorem 4.7, that is, $\mathrm{tww} \sqsubseteq \mathrm{otww}$ (we observed that $\mathrm{otww} \sqsubseteq \mathrm{tww}$). In Lemma 4.1, when we proved $\mathrm{mn} \sqsubseteq \mathrm{tww}$, the reader can check that we actually showed that $\mathrm{mn} \sqsubseteq \mathrm{otww}$. We conclude since by Lemma 4.2, $\mathrm{tww} \sqsubseteq \mathrm{mn}$. □

More quantitatively, we have established that for every graph G, $\mathrm{tww}(G) \leqslant 2^{2^{O(\mathrm{otww}(G))}}$. Let us use this fact to bound the twin-width of K_t-minor-free graphs. In particular, this establishes that every proper minor-closed class has bounded twin-width, as every such class excludes a graph H, and hence the complete graph $K_{|V(H)|}$ as a minor. We simply need the following lemma.

Lemma 4.6 ([72]). *Every K_t-minor-free graph on at least two vertices admits two distinct vertices u, v both of degree at most $2^{O(t \log t)}$ such that u and v are either adjacent or false twins.*

To bound the oriented twin-width, we can think the *vertex contractions* in terms of graphs rather than trigraphs, or equivalently work with the total graphs rather than the red graphs. More specifically, to argue that $\mathrm{otww}(G) \leqslant d$, it is sufficient to find a contraction sequence of G such that every newly contracted vertex has degree at most d in the current total graph.

The proof of the next theorem iteratively contracts pairs u, v given by Lemma 4.6. Importantly, either u and v are adjacent and this vertex contraction is an edge contraction, or u and v are false twins and the vertex contraction can be thought as the deletion of v. In both cases then, the new total graph remains K_t-minor-free.

Theorem 4.8 ([25]). *K_t-minor-free graphs have twin-width $2^{2^{2^{\tilde{O}(t)}}}$.*

CHAPTER 5

Which Classes Have Bounded Twin-Width?

In the previous chapters, we have started to see examples of classes with bounded twin-width, namely those of cographs, unit interval graphs, graphs excluding a fixed minor, and posets with bounded maximum antichain. In this chapter, we continue this exploration more systematically. To bound the twin-width of a class, we have seen, in addition to directly providing contraction sequences, two arguably simpler alternatives: bounding the mixed number or the oriented twin-width. Anticipating Theorem 8.1 in Chapter 8, another option is to transduce the class from a class already known to have bounded twin-width.

How to show, on the contrary, that a class has *unbounded* twin-width? In some simple cases, a direct argument can be given. The easiest is when for every integer n, the class contains a graph G such that every pair $u, v \in V(G)$ satisfies $|(N(u) \setminus N[v]) \cup (N(v) \setminus N[u])| \geqslant n$; as it happens with rook graphs or Paley graphs. Then, any first contraction in G creates a vertex with at least n red neighbors. A very effective yet not completely satisfactory way of establishing that a class has unbounded twin-width is to show that its growth (or more conveniently the growth of its hereditary closure) is too fast. We will indeed see in Chapter 9 that classes of bounded twin-width are small, and even tiny; see Theorems 9.1 and 9.2. In Chapter 10 the converse is shown to hold for hereditary classes of ordered graphs. This too can be used to show that an *unordered* graph class has bounded twin-width [**43**].

1. Classical graph widths

In Chapter 2 the definition of several width measures was recalled: treewidth, clique-width, rank-width, and boolean-width. We remind the reader that boolw $\equiv$ rw $\equiv$ cw $\sqsubset$ tw; see for instance [**81**, Chapter 4]. We now prove that tww $\sqsubset$ boolw, making classes of bounded twin-width more general than those for which any of the latter widths (dubbed *classical*) is bounded.

Theorem 5.1 ([**26**]). *Every class of bounded boolean-width has bounded twin-width.*

PROOF. Let G be a graph of boolean-width k, and (T, γ) be a witnessing tree layout of G. We call *partitioned tree layout* on G a tree layout (T', γ') where the leaves are no longer in one-to-one correspondence with $V(G)$, but rather with the parts of a partition of it, i.e., $\{\gamma'(x) \ : \ x \in L(T')\}$ is a partition of $V(G)$.

We show that G has twin-width at most $2^{k+1} - 1$. For that, we construct a partition sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ of G in parallel with a sequence of partitioned tree layouts $(T_n, \gamma_n), \dots, (T_1, \gamma_1)$, where (T_n, γ_n) is the tree layout (T, γ) in which the label v of every leaf is turned to the singleton $\{v\}$.

For i going from n down to 2^{k+1}, let T_i' be a rooted subtree of T_i with at least $2^k + 1$ and at most 2^{k+1} leaves. Finding T_i' can be done by starting at the root of T_i, and iteratively moving to the child with larger subtree (or either child, in case of a tie), until the condition is met. By the pigeonhole principle and the definition of boolean-width, there are two parts $P, P' \in \mathcal{P}_i$ labeling two leaves of T_i' such that all vertices in $P \cup P'$ have, in G, the same neighborhood outside $\bigcup_{x \in L(T_i')} \gamma_i(x)$. We then go from $\mathcal{P}_i$ to $\mathcal{P}_{i-1}$ by contracting P with P'.

This affects the partitioned tree layout in the following way. We obtain (T_{i-1}, γ_{i-1}) from (T_i, γ_i) by replacing label P with label $P \cup P'$, deleting the leaf ℓ labeled by P', and smoothing out the parent of ℓ (i.e., deleting it and linking the sibling of ℓ directly to its grandparent). We may then arbitrarily finish the sequence $\mathcal{P}_{2^{k+1}-1}, \dots, \mathcal{P}_1$ (with the corresponding partitioned tree layouts). For every $i \in [n]$, every part $P = \gamma_i(\ell) \subseteq V(G)$ labeling a leaf ℓ of T_i, and the maximum rooted subtree $T_{i,P}$ of T_i that is of size at most 2^{k+1} and contains ℓ, it holds that the vertices of P have, in G, the same neighborhood outside $\bigcup_{x \in L(T_{i,P})} \gamma_i(x)$.

This implies that part P can have at most $2^{k+1}-1$ red neighbors. □

Therefore tww $\sqsubseteq$ boolw, but the converse does not hold as, for example, the class of all grids has twin-width at most 4 (see the proof by picture of Figure 5.1), but unbounded boolean-width. For every graph G, $\mathrm{boolw}(G) \leqslant \max(\mathrm{tw}(G)+1, \mathrm{cw}(G), \frac{1}{4}\mathrm{rw}(G)^2 + \gamma \mathrm{rw}(G))$ for some constant γ; see [**81**, Chapter 4] and [**32**]. Thus we proved that twin-width is at most single-exponential in tw, cw, boolw, and rw^2; see also [**59**] for a tighter upper bound of twin-width in terms of treewidth.

It happens that this exponential blow-up is required for all these parameters but clique-width [**17**], whereas $\mathrm{tww}(G) \leqslant 2(\mathrm{cw}(G)-1)$ for every graph G [**7**].

2. Subdivisions, grids, and expanders

In this section, we realize that classes of bounded twin-width extend much further than classes with bounded boolean-width, which we recall are the same as those with bounded clique-width or bounded rank-width. We analyze the behavior of twin-width on graph subdivisions, generalizations of grids, and expanders, all known to be of unbounded boolean-width.

It happens that any graph, if "subdivided enough," has low twin-width. Thus, unlike classical width parameters, twin-width is not a topological notion.

Theorem 5.2 ([**9**])**.** *Any* $(\geqslant 2\lceil \log n \rceil)$*-subdivision of any* n*-vertex graph has twin-width at most 4.*

The $\Omega(\log n)$ bound in the required length of the subdivision to make the twin-width low is essentially tight. The $o(\log n)$-subdivisions of the complete graph K_n have, on the contrary, unbounded twin-width. This can be shown by the counting argument (see Theorem 9.1), as their hereditary closure does not form a small class. However, a direct proof, with an explicit lower bound, is possible.

Theorem 5.3 ([**26**])**.** *Let* n, d *be integers larger than 2. For any positive integer* $\ell < \log_d(n-1) - 1$*, the* ℓ*-subdivision of* K_n *has twin-width at least* d*.*

SKETCH. Consider the first time in an hypothesized contraction sequence $\mathcal{S}$ of width $d-1$ when a branching vertex v is contracted with another part/vertex. As v has a set X of n neighbors such that for every other vertex u, $|N(u) \cap X| \leqslant 1$, all but at most one edge incident to v will turn red. This implies that at least n/d neighbors of v are, at this point of $\mathcal{S}$, in the same part. The same reasoning can be applied to this part, and its at least n/d neighbors outside v: n/d^2 of them shall be in the same part, and so on. This bottoms out when, after ℓ steps, we reach the other branching vertices, $n-2$ of which are, by definition, in singleton parts. □

A thorough analysis of the twin-width of subdivisions has been carried out [**1**]. In particular, the existence of n-vertex graphs whose $(\geqslant 2\lceil \log n \rceil)$-subdivisions have twin-width *exactly* 4 is shown.

That grids have bounded twin-width is, at this point, no surprise. We have seen in Chapter 4 the much more general result that any class excluding a minor has bounded twin-width; see Theorem 4.8. Indeed grids are K_5-minor-free. However, we now see that generalizations of grids to higher fixed dimensions—which do *not* exclude any minor—still have bounded twin-width, linear in the dimension. The class of *d-dimensional grid graphs* comprises any finite induced subgraph of the *infinite d-dimensional lattice*, with vertex set $\mathbb{Z}^d$ and edges linking every pair of vertices at Euclidean distance exactly 1. The *d-dimensional grid* $\Gamma(n_1, n_2, \ldots, n_d)$ is the subgraph of this lattice induced by $[n_1] \times [n_2] \times \ldots \times [n_d]$.

Theorem 5.4 ([**26**])**.** *The class of d-dimensional grid graphs has twin-width at most $3d$.*

SKETCH. We give a short argument for the worse upper bound of $4d$. By Observation 3.2, it is enough to show the bound for $\Gamma(n, n, \ldots, n)$ for every natural n (with d coordinates). By Observation 3.3, we can further assume that all its edges are already red. We denote this graph of maximum red degree $2d$, by $\Gamma^{red}(n, n, \ldots, n)$.

We show our claim by induction on d. The base case holds since the d-dimensional grid with $d = 0$ has a single vertex. If $d > 0$, let M be the perfect matching between the first and second slices of $\Gamma^{\mathrm{red}}(n, n, \ldots, n)$ in the direction orthogonal to the first axis. By Lemma 3.7, contracting the (disjoint) pairs of endpoints of M can be done in a partial $4d$-sequence, and results in $\Gamma^{\mathrm{red}}(n-1, n, \ldots, n)$. After $n-2$ successive applications of Lemma 3.7, we obtain $\Gamma^{\mathrm{red}}(n, \ldots, n)$ with only $d-1$ coordinates, and conclude. □

There is a 4-sequence for (2-dimensional) grids, as shown in Figure 5.1. This sequence is optimal for the 7×7 grid (and above) [**1**].

While the known upper bounds for the twin-width of K_t-minor-free graphs are quite loose, the twin-width of planar graphs, and more generally of graphs of bounded genus (two families excluding some fixed minors) is better understood. The class of planar graphs has twin-width at most 8 [**57**] and at least 7 [**64**], and the class of graphs with Euler genus g has twin-width $\Theta(\sqrt{g})$ [**65**].

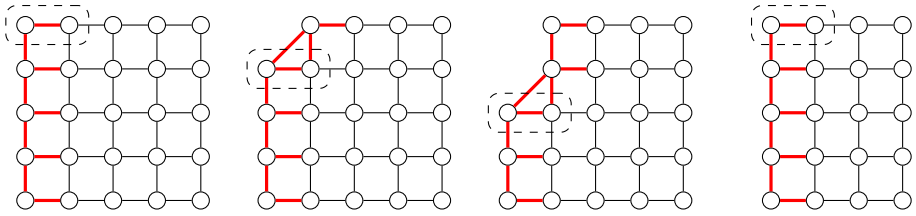


FIGURE 5.1. A 4-sequence for planar grids.

Even some classes of *expanders*, which we will informally and approximately define here as bounded-degree n-vertex graphs with treewidth $\Theta(n)$, may have bounded twin-width. This is somewhat surprising since these classes are thought as "maximally unstructured" among sparse graphs.

A *2-lift* of a graph G is a graph G' on twice as many vertices, built by duplicating every vertex $v \in V(G)$ into two copies v_1 and v_2, and for every edge $vw \in E(G)$, adding to $E(G')$ either the *parallel* edges v_1w_1 and v_2w_2 or the *crossing* edges v_1w_2 and v_2w_1. The choice, for each edge e of G, of yielding parallel edges or crossing edges in G' is called the *signing* of e. See Figure 5.2 for an example of a 2-lift.

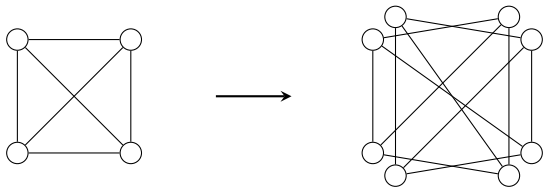


FIGURE 5.2. An example of a 2-lift of K_4.

For n a power of 2, performing a sequence of $\log n - 2$ randomly-signed 2-lifts starting on K_4 almost surely yields an n-vertex expander [**12**]. The next result shows that cubic expanders can have bounded twin-width.

Fact 5.1. *Every graph obtained from K_4 by performing a sequence of 2-lifts has twin-width at most 6.*

SKETCH. Simply rewind the sequence of 2-lifts (it can be seen as a split sequence if all edges are turned red) and apply Lemma 3.7. □

The Margulis–Gabber–Galil expander family [**48**], in which the n-th graph has vertex set $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ and every vertex (x, y) is linked to

$$(x \pm 2y, y), (x \pm (2y+1), y), (x, y \pm 2x), (x, y \pm (2x+1)),$$

also has bounded twin-width. This can be shown by repeatedly applying Lemma 3.7 to the partition of $\mathbb{Z}/n_i\mathbb{Z} \times \mathbb{Z}/n_i\mathbb{Z}$ into "squares" $\{(x,y),(x+1,y),(x,y+1),(x+1,y+1)\}$ with $n_1 = n$, and $n_{i+1} = \lceil n_i/2 \rceil$ for i going from 1 to $\lfloor \log n \rfloor - 1$.

By Theorem 9.1, a graph family built by picking, for every positive integer n, an n-vertex subcubic graph uniformly at random, has almost surely unbounded twin-width. However, we do not know of an explicit construction of bounded-degree expanders with unbounded twin-width; see Section 4.2.

3. Intersection graphs

We recall that the *intersection graph* G of a collection $\mathcal{O}$ of sets has vertex set $\mathcal{O}$ and an edge between two elements of $\mathcal{O}$ whenever they intersect. The collections $\mathcal{O}$ is called the *representation* of G, or geometric representation if the sets of $\mathcal{O}$ have some geometric nature.

We now need the notion of half-graphs. For us,[1] a *half-graph* is a graph on vertex set $A \uplus B$ with $A = \{a_1, \dots, a_h\}$ and $B = \{b_1, \dots, b_h\}$ such that $a_i b_j$ is an edge whenever $i \leqslant j$, and each of A and B is a clique or an independent set; see left of Figure 5.3.

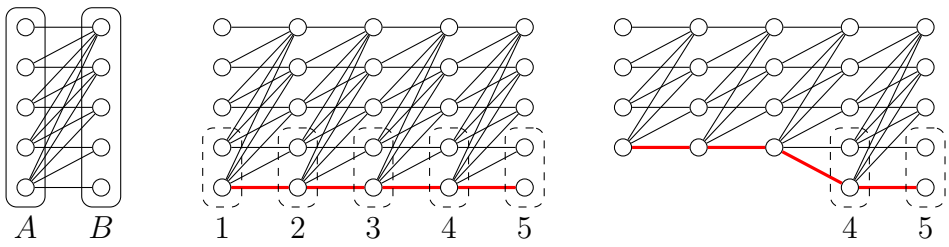


FIGURE 5.3. Half-graph and 2-sequence for a path of half-graphs.

A *path of half-graphs* of height h and length ℓ is a graph on vertex set $A_1 \uplus \ldots \uplus A_\ell$ with $A_k = \{a_1^k, \dots, a_h^k\}$ for every $k \in [\ell]$, such that for every $k \in [\ell - 1]$, $a_i^k a_j^{k+1}$ is an edge whenever $i \leqslant j$. The path of half-graphs has no other edge, except that possibly some of the sets A_k are cliques. See middle of Figure 5.3 for an illustration, disregarding the color of the bottom path of red edges. We can now refine the implicit upper bound on the twin-width of unit interval graphs, seen on Chapter 4, to the exact value.

[1] There are many variations on the definition of half-graphs.

Theorem 5.5 ([21]). *The class of unit interval graphs has twin-width 2.*

SKETCH. It can be shown that every n-vertex unit interval graph is an induced subgraph of the path of half-graphs H_n of height and length n where $A_1, \dots, A_n$ all are cliques. Figure 5.3 is a proof by picture that the latter graphs have twin-width at most 2. One can then argue that H_n has twin-width *exactly* 2, and is itself an unit interval graph. □

Similarly introducing *cycles of half-graphs*, one can prove that the class of *unit circular-arc graphs*, i.e., intersection graphs of unit-length arcs on a circle, has twin-width 3.

The *symmetric difference* of a graph G, denoted by $\mathrm{sd}(G)$, is the maximum for every induced subgraph G' of G, of the minimum for every pair $u \neq v \in V(G')$ of

$$\left|\{w \in V(G') \setminus \{u, v\} \ : \ w \text{ is adjacent to exactly one of } u, v\}\right|.$$

We already noticed that the first contraction in a sequence (for the induced subgraph G') implies the following.

Observation 5.2. *For every graph G, $\mathrm{tww}(G) \geqslant \mathrm{sd}(G)$.*

Like the *rook graphs*[2] and *Paley graphs*[3], permutation graphs can be shown to have unbounded symmetric difference, hence unbounded twin-width. The *grid permutation* on n^2 elements is the permutation σ such that $\sigma((a-1)n + b) = (b-1)n + a$, for every $a, b \in [n]$.

Theorem 5.6 ([22]). *The class of all permutation graphs has unbounded twin-width.*

PROOF. For every positive integer n, consider the permutation graph G for the grid permutation on n^2 elements, deprived of the last element n^2. We conclude by Observation 5.2 since $\mathrm{sd}(G) = n - 1$. □

A *proper permutation graph class* is any hereditary subclass of permutation graphs that is *not* the class of all permutation graphs. The following theorem is the graphic counterpart of the fact that any

[2]With vertex set $E(K_{t,t})$ and an edge between every pair of incident edges.

[3]With vertex set $\mathbb{F}_q$ for q a prime power with $q \equiv 1$ [4], and an edge between every pair whose difference is a nonzero square of $\mathbb{F}_q$.

proper permutation class has a bounded width as defined by Guillemot and Marx [**51**]; see Section 2 of Chapter 1.

Theorem 5.7 ([**26**]). *Any proper permutation graph class has bounded twin-width.*

For every permutation σ on n-elements, let $G(\sigma)$ be the $3n$-vertex graph with vertex set $A \uplus B \uplus C$, $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_n\}$, $C = \{c_1, \dots, c_n\}$, and edge set

$$\{a_i b_j \ : \ j < i\} \cup \{b_i c_j \ : \ j < \sigma(i)\}.$$

A *representation of* $G(\sigma)$ by geometric objects $\mathcal{O}$ is such that the intersection graph of $\mathcal{O}$ is isomorphic to any graph obtained from $G(\sigma)$ by possibly turning some of A, B, C into cliques.

Interval graphs are the intersection graphs of intervals on the real line. *Unit disk graphs* (*segment graphs*, *unit segment graphs*, respectively) are intersection graphs of disks of equal radius (segments, segments of equal length, respectively) in the plane. We prepend the adjective *axis-parallel* if all the segments can be drawn parallel to the axes.

Theorem 5.8 ([**22**]). *The classes of interval graphs, unit disk graphs, and axis-parallel unit segment graphs all have unbounded twin-width.*

SKETCH. All these classes can represent $G(\sigma)$ for any permutation σ, as shown in Figure 5.4. Then, one can finish the proof in two dif-

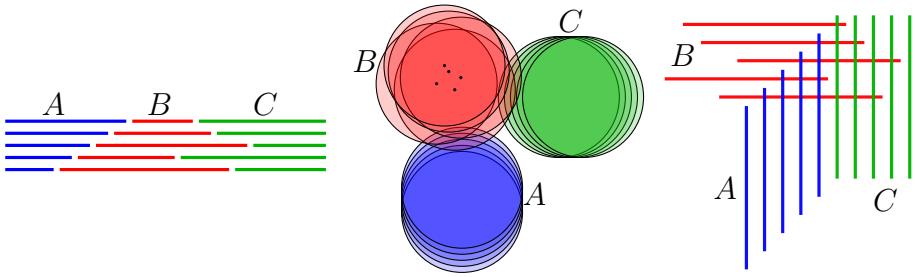


FIGURE 5.4. Representations of $G(\sigma)$ for $\sigma = 41532$.

ferent ways. One can show that the class $\{G(\sigma) \ : \ \sigma \text{ is a permutation}\}$ is not small, and invoke Theorem 9.1.

Alternatively, one can lower bound the twin-width of $G(\sigma)$ for the grid permutation on n^2 elements. Considering the first moment in the

contraction sequence where a vertex of B is contracted with another vertex/part, one can see that $\mathrm{tww}(G(\sigma)) = \Omega(n)$. □

The large cliques in the representation of $G(\sigma)$ with unit disks, and the large bicliques in the one with segments are both necessary. It can indeed be established that unit disk graphs without K_t **[26]**, and segment graphs without $K_{t,t}$ subgraph **[16]** have twin-width at most some function of t. The former is shown by exhibiting a partial contraction sequence to a red grid with diagonals, whose twin-width can be bounded similarly to the grid. The latter has a more involved proof whose base strategy is to transduce the class from the class of planar graphs, and invoke Theorem 8.1.

A more detailed investigation of the twin-width of classes of intersection or visibility graphs can be found in the following papers **[16, 4, 58]**.

4. Sparse classes

We use the adjective *sparse* to qualify a class as an informal umbrella term encompassing classes of graphs with subquadratically many edges. Mainly, sparse classes will refer to either of four families of increasing sparsity: weakly sparse classes, classes of bounded degeneracy, classes of bounded expansion, and classes of bounded degree.

4.1. Weakly sparse classes. Classes of bounded twin-width that are additionally weakly sparse can still be "complex" and seemingly "unstructured," as long subdivisions of every graph, and some classes of expanders fit in this category; recall Section 2. More generally[4] than the $\Omega(\log n)$-subdivisions of every n-vertex graph, classes of bounded queue number and of bounded stack number both have bounded twin-width. The *queue and stack numbers* of a graph G are defined as the minimum number of parts to partition $E(G)$ in, so that there is a linear order $\prec$ on $V(G)$ that makes each part be a queue (for queue number) or a stack (for stack number). A *queue* (resp. *stack*) is such that there is no $uv, xy \in E(G)$ with $u \prec x \prec y \prec v$ (resp. $u \prec x \prec v \prec y$).

Theorem 5.9 ([22]). *Every class with bounded queue number or with bounded stack number has bounded twin-width.*

SKETCH. The definition of queue and stack numbers readily implies that the corresponding classes have bounded mixed number (and even, grid number). □

[4]Indeed, the queue and stack numbers of long subdivisions are also bounded.

Notwithstanding the persistent complexity of weakly sparse classes of bounded twin-width, we will now see an interesting collapse down to classes of bounded degeneracy, and even of bounded expansion. This is also the occasion for a resurgence of the grid number, introduced in Chapter 4, as the sparse counterpart of the mixed number.

The *degeneracy* of a graph G is the maximum for every (induced) subgraph G' of G of the minimum degree among vertices in G'. A graph class $\mathcal{C}$ has *bounded expansion* if for every natural ℓ, the class of graphs whose ℓ-subdivision is a subgraph of a member of $\mathcal{C}$ has bounded degeneracy [**71**].

Theorem 5.10 ([**22**])**.** *Let $\mathcal{C}$ be a graph class of bounded twin-width. The following are equivalent:*

(1) $\mathcal{C}$ is weakly sparse;
(2) $\mathcal{C}$ has bounded grid number;
(3) $\mathcal{C}$ has bounded degeneracy;
(4) $\mathcal{C}$ has bounded expansion;
(5) The subgraph closure $\mathrm{Sub}(\mathcal{C})$ has bounded twin-width.

SKETCH. (1) $\Rightarrow$ (2). By Theorem 4.2, every graph $G \in \mathcal{C}$ has an adjacency matrix with low mixed number. If this matrix were of high grid number, it would admit a large division with nonzero cells, most of which not mixed, and hence horizontal or vertical. From them, one could extract a large biclique, contradicting that $\mathcal{C}$ is weakly sparse.

(2) $\Rightarrow$ (3). This is a reformulation of the Marcus–Tardos theorem.

(3) $\Rightarrow$ (1). This holds since large bicliques have large degeneracy.

(2) $\Rightarrow$ (5). Indeed, taking subgraphs can only decrease the grid number, which eventually ensures bounded twin-width.

(5) $\Rightarrow$ (1). This is because the subgraph closure of a large biclique contains all bipartite graphs of the same size, which is a family of large twin-width.

(4) $\Rightarrow$ (3). By definition of bounded expansion for $\ell = 0$.

At this point, we know that $(1), (2), (3), (5)$ are all equivalent, and implied by (4). We finally show that $(3) \wedge (5) \Rightarrow (4)$. For every ℓ, the class $\mathcal{D}$ of ℓ-subdivisions of graphs in $\mathrm{Sub}(\mathcal{C})$ can be transduced from $\mathrm{Sub}(\mathcal{C})$; of bounded twin-width by (5). Thus by Theorem 8.1, $\mathcal{D}$ has bounded twin-width. Besides, $\mathcal{D}$ preserves the bounded degeneracy of $\mathcal{C}$. So $\mathcal{C}$ has bounded expansion. □

4.2. Bounded-degree classes. One can go one step further up the sparsity ladder, and exclude vertices of large degree. Still, bounded-degree classes of bounded twin-width can be fairly complex, as some expander classes and the long subdivisions of all subcubic graphs persist. One notable feature of bounded-degree graphs, nonetheless, is that a closely tied parameter to twin-width can be defined without the need for trigraphs or red edges. It is based on the fact that, when performing contractions, the number of black neighbors of a vertex never increases.

Observation 5.3. *Every graph G of maximum degree Δ and twin-width d admits a contraction sequence all total graphs of which have maximum degree at most $\Delta + d$.*

Therefore, one can equivalently work with the variant of twin-width based on vertex contractions in graphs. The new width is simply the overall maximum degree in a sequence of (graphic) vertex contractions.

While we know that there are classes of bounded-degree graphs with unbounded twin-width by a counting argument (see Theorem 9.1), no explicit such family has been found so far. By *explicit*, we mean to ask for an infinite family of graphs $H_1, H_2, \dots$ of bounded-degree, and a proof that for every $i \in \mathbb{N}$, $\mathrm{tww}(H_i) \geqslant i$ *not* resorting to Theorem 9.1.

CHAPTER 6

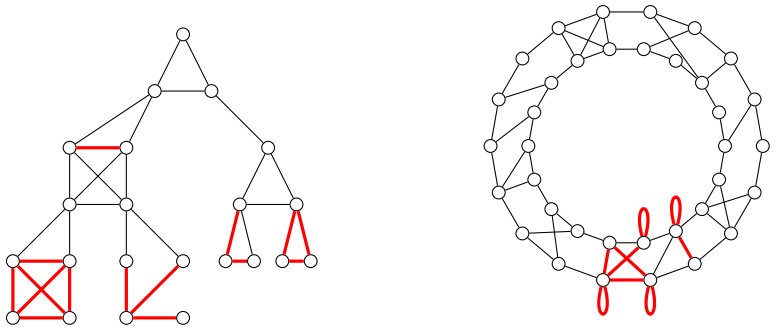
Other Parameters based on Contraction Sequences

What if we imposed different conditions on the red graphs of a contraction sequence other than bounding their maximum degree? One could be tempted,[1] for instance, to instead impose that all the red graphs have, say, bounded treewidth. This is unpromising as every graph admits a contraction sequence each red graph of which has treewidth at most 1. Indeed, if $v_1, \dots, v_n$ are its vertices, the partition sequence $\{\{v_1\}, \dots, \{v_n\}\}, \{\{v_1, v_2\}, \{v_3\}, \dots, \{v_n\}\}, \{\{v_1, v_2, v_3\}, \{v_4\}, \dots, \{v_n\}\}, \dots, \{\{v_1, \dots, v_n\}\}$ is such that each red graph is the disjoint union of a *star* (i.e., a tree with at most one internal node) and *isolated vertices* (i.e., vertices of degree 0).

However, for any parameter p that is *unbounded* on the class of stars, it may be interesting to consider the *reduced parameter* $p^{\downarrow}$ defined on every graph G as the minimum $p^{\downarrow}(G)$ among every contraction sequence $G = G_n, \dots, G_1$ of $\max_{i \in [n]} p(\mathcal{R}(G_i))$. In particular, $\Delta^{\downarrow}$ is another notation for tww. In fact we have already encountered a similar kind of alternative parameter to twin-width in the oriented twin-width. The only small difference is that the oriented twin-width was defined via the directed red graphs instead of the red graphs. We will now stick to the red graphs, and consider several reduced parameters. As stars are very simple graphs, we will exclusively consider parameters $p^{\downarrow}$ such that tww $\sqsubseteq p^{\downarrow}$, with p growing at least linearly in Δ.

The motivation then is not, as hinted by the footnote, to generalize twin-width but on the contrary to particularize it. In Section 1 we realize that clique-width and its linear counterpart are tied to some reduced parameters. Section 2 introduces a hierarchy of novel reduced

[1] Arguably, at this point, the only motivation of twin-width we have seen outside the overview of Chapter 1 is that classes of bounded twin-width are fairly general (Chapter 5) while χ-bounded (Theorem 3.2). In the next chapters, we will see many more applications, for efficient algorithms (Chapter 7), logical characterizations and properties (Chapter 8), adjacency labeling schemes (Chapter 9), enumerative combinatorics (Chapter 10). Anticipating these applications, one may legitimately want to extend them to other families of graph classes based on contraction sequences.



(a) Component twin-width. (b) Total twin-width.

FIGURE 6.1. Trigraphs of low component twin-width and low total twin-width, in seemingly sparse graphs, where these reduced parameters are tied to treewidth and pathwidth, respectively.

parameters ranging from clique-width up to twin-width, gives some motivations for their study, and briefly surveys some of their properties. In Section 3 we show that this hierarchy is strict.

1. Characterization of classical width parameters

In this section, we recast the classical width invariants defined and developed in the last fifty years, in the language of contraction sequences. Let $\star(G)$ be the maximum size among connected components of G. Note that $\star(G) \geqslant \Delta(G)+1$. We call *component twin-width* the reduced parameter $\star^{\downarrow}$. Remarkably, component twin-width is functionally equivalent to boolean-width, hence to clique-width and rank-width, and within weakly sparse classes, to treewidth; see Figure 6.1a.

Theorem 6.1 ([25]). *Component twin-width and boolean-width are functionally equivalent.*

PROOF. The direction $\star^{\downarrow} \sqsubseteq$ boolw is actually proven in Theorem 5.1. Indeed, the reader can observe that we bound the maximum degree of the red graphs by arguing they have no connected component larger than a given (exponential) bound in the boolean-width.

We thus only need to argue that boolw $\sqsubseteq \star^{\downarrow}$. Let, for any graph G, $\mathcal{P}_n, \dots, \mathcal{P}_1$ be a partition sequence of G such that $\star(\mathcal{R}(\mathcal{P}_i)) \leqslant \star^{\downarrow}(G)$ for each $i \in [n]$. We initialize an eventual tree layout as the star T_n, rooted at its center, whose n leaves are bijectively labeled by $V(G)$. Thus trivially, for every edge $e \in E(T_n)$, there are only few distinct

neighborhoods from one side of the bipartition (X_e, Y_e) to the other. However, T_n is not a binary tree so does not qualify as a tree layout. Following the partition sequence, we define a sequence of trees $T_n, T_{n-1}, \dots, T_1$ such that T_1 (deprived of its root) is a tree layout.

Let P, P' be the two merged parts when going from $\mathcal{P}_i$ to $\mathcal{P}_{i-1}$, and C be the connected component of $P \cup P'$ in $\mathcal{R}(\mathcal{P}_{i-1})$. Let C_P (resp. $C_{P'}$) be the connected component of P (resp. P') in $\mathcal{R}(\mathcal{P}_i)$. Note that C is made of $C_P, C_{P'}$ (which can be equal) together with at most $\star^{\downarrow}(G) - 1$ other connected components of $\mathcal{R}(\mathcal{P}_i)$. By induction, these $h \leqslant \star^{\downarrow}(G) + 1$ connected components each labels the leaves of a binary subtree U_j (with $j \in [h]$) rooted at a child of the root of T_i. We take any rooted binary tree U with h leaves, for instance a balanced one, identify each leaf with the root of a distinct U_j, and make the root of U a child of T_i. We finally remove the previous occurrence of each U_j (i.e., the one that is a child of the root), and obtain T_{i-1}. In T_1, we delete the initial root, and call the resulting rooted tree T.

As $\{V(G)\}$ is the unique vertex of $\mathcal{R}(\mathcal{P}_1)$, T is indeed binary, hence a tree layout of G. By induction, for every edge $e \in E(T)$, the bipartition (X_e, Y_e) is such that there are at most $2^{\star^{\downarrow}(G)}$ distinct neighborhoods of subsets of X_e (resp. Y_e) in Y_e (resp. X_e). □

For what comes next we have to very slightly amend our red graphs: Every non-singleton part has a red self-loop on it. This does not change much the twin-width nor the other parameters based on contraction sequences seen so far (oriented twin-width and component twin-width). If we decide to disregard the red loops when counting the red degree, this does not change the twin-width at all, and offsets it by at most 1, otherwise.

If G is a graph possibly having self-loops, let $e(G)$ be its total number of edges. Let $e^{\Downarrow}(G)$ be the minimum, among every contraction sequence $G = G_n, \dots, G_1$, of $\max_{i \in [n]} e(\mathcal{R}^{\circ}(G_i))$, where $\mathcal{R}^{\circ}(G_i)$ represents the amended red graph with self-loops. We use a different arrow symbol to be consistent with our generic definition of $p^{\downarrow}$. The reduced parameter $e^{\Downarrow}$, which we call *total twin-width*, is tied to the linear variants of boolean-width, clique-width, and rank-width, and within weakly sparse classes, to pathwidth; see Figure 6.1b.

Theorem 6.2 ([25]). *Total twin-width and linear boolean-width are functionally equivalent.*

SKETCH. To upper bound total twin-width as a function of linear boolean-width, follow Theorem 5.1 in the particular case when the tree layout is a comb. This yields a contraction sequence where, at any given point, all the red edges (including the self-loops) lie in a single subtree of bounded size. The converse is obtained by revisiting the proof of Theorem 6.1. If the contraction sequence witnesses low total twin-width, then the built tree layout can be made a comb. □

2. New parameters between clique-width and twin-width

The reduced parameters of the previous section can be thought as the bottom of a hierarchy interpolating between clique-width and twin-width. There are indeed many parameters p such that $\Delta \sqsubseteq p \sqsubseteq \star$. The corresponding reduced parameter $p^{\downarrow}$ then sits between clique-width and twin-width.

The *bandwidth* of an n-vertex graph G, denoted by $\mathrm{bandw}(G)$, is the minimum of $\max_{uv\in E(G)} |f(u)-f(v)|$ taken among every bijection f from $V(G)$ to $[n]$. Its *cutwidth*, denoted by $\mathrm{cutw}(G)$, is the minimum of $\max_{i\in[n]} |\{uv \in E(G) \ : \ f(u) \leqslant i < f(v)\}|$ also taken among all the bijections f from $V(G)$ to $[n]$. More intuitively, bandwidth is about injectively labeling the vertices of a graph with integers so as to minimize the maximum *stretch* of an edge, i.e., difference of its two endpoints, whereas cutwidth is about linearly ordering its vertex set, say on a horizontal line, so as to minimize the maximum number of edges crossing any vertical line.

For every graph G, $\lceil \Delta(G)/2 \rceil \leqslant \mathrm{bandw}(G) \leqslant \star(G) - 1$, whereas $\lceil \Delta(G)/2 \rceil \leqslant \mathrm{cutw}(G) \leqslant \Delta(G) \cdot \mathrm{bandw}(G) \leqslant 2\,\mathrm{bandw}(G)^2$. Therefore,

$$\Delta \sqsubseteq (\mathrm{tw} + \Delta) \sqsubseteq \mathrm{cutw} \sqsubseteq \mathrm{bandw} \sqsubseteq \star,$$

where $(\mathrm{tw}+\Delta)(G)$ is simply the sum of the treewidth of G with its maximum degree. It is indeed known that $\mathrm{cutw} \equiv (\mathrm{pw} + \Delta)$. Actually these four inclusions are strict, so

$$\Delta \sqsubset (\mathrm{tw} + \Delta) \sqsubset \mathrm{cutw} \sqsubset \mathrm{bandw} \sqsubset \star,$$

as evidenced by grids, full binary trees, subdivisions of the combs, and paths.[2] We obtain the three new parameters sandwiched between clique-width and twin-width

$$\mathrm{tww} = \Delta^{\downarrow} \sqsubseteq (\mathrm{tw} + \Delta)^{\downarrow} \sqsubseteq \mathrm{cutw}^{\downarrow} \sqsubseteq \mathrm{bandw}^{\downarrow} \sqsubseteq \star^{\downarrow} \equiv \mathrm{cw}.$$

We will see in Section 3 that these inclusions too are strict.

[2]The second and third separating examples require more thought.

A motivation for studying this hierarchy of reduced parameters is to better understand and locate "phase transitions" between the more structured but less general classes of bounded clique-width and the more widespread but less tamed classes of bounded twin-width. For example we have seen, on the one hand, that proper minor-closed classes have bounded twin-width (see Theorem 4.8). On the other hand, the class of all grids, which is K_5-minor-free, has unbounded clique-width. Which is the largest reduced parameter still bounded on grids? Among the presented ones, we know the answer: it is $\mathrm{bandw}^{\downarrow}$.

Theorem 6.3 ([27]). *The K_t-minor-free graphs have bounded reduced bandwidth.*

This shows how structured we can have the red graphs in contraction sequences of bounded width of K_t-minor-free graphs. They can be made subgraphs of the constant power[3] of paths, which characterizes bounded bandwidth.

Let us take another example. We know that there are classes of expanders of bounded twin-width, but none of bounded clique-width. Where exactly do we lose every expander class? So far, we only have a partial answer.

Theorem 6.4 ([27]). *Every expander class has unbounded reduced bandwidth.*

It is likely that Theorem 6.4 holds for $(\mathrm{tw}+\Delta)^{\downarrow}$, but this remains conjectural. The presence of expanders in a class $\mathcal{C}$ is often an obstacle to fast exact algorithms or good approximation algorithms designed on $\mathcal{C}$. Determining a smallest parameter p such that classes of bounded $p^{\downarrow}$ exclude expanders would tell us how close to twin-width one can be while falling in a qualitatively simpler regime.

There is also an interesting interplay between reduced parameters and subdivisions. We saw in Chapter 5 that any $(\geqslant 2\lceil \log n \rceil)$-subdivision of any n-vertex graph has twin-width at most 4, making twin-width, unlike the classical width parameters, a non-topological invariant. When trying to solve a combinatorial problem faster on graphs of bounded twin-width than on general graphs, this comes in the way. Indeed many graph problems, like finding an independent set of maximum size or computing the diameter, are almost as hard to solve exactly on a graph G as they are on any even subdivision of G. Nonetheless, as we will see in Chapter 7, contraction sequences

[3] where one links vertices at distance at most a fixed constant

of low width allow fixed-parameter tractable algorithms on problems definable in first-order logic, and improved approximation algorithms.

Topological reduced parameters and their associated contraction sequences, despite being less general than twin-width, would have a wider range of algorithmic applications. But do they exist apart from clique-width? We wonder for each q among $(\mathrm{tw}+\Delta)^{\downarrow}$, $\mathrm{cutw}^{\downarrow}$, $\mathrm{bandw}^{\downarrow}$ if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that the $(\geqslant f(n))$-subdivisions of all n-vertex graphs have bounded parameter q. The answer is positive for all three parameters, with a logarithmic function f for $(\mathrm{tw}+\Delta)^{\downarrow}$ (in fact, the same as for twin-width), and a polynomial function f for $\mathrm{cutw}^{\downarrow}$ and $\mathrm{bandw}^{\downarrow}$ [**27**].

This motivates the introduction of *stretch-width*, denoted by *stw*, which further constrains reduced bandwidth by imposing that the upper bound on the bandwidth of every red graph is realized by a single linear arrangement of the original vertices (see [**19**] for the technical definition). This seemingly small difference between stretch-width and reduced bandwidth is key. Even if stretch-width is still not topological, only exponentially long subdivisions have bounded stretch-width [**19**], which no longer precludes polynomial-time algorithms when given witnesses of low stretch-width, on otherwise NP-hard graph problems. Furthermore planar graphs, and more specifically grids, on which problems beyond first-order model checking tend to be intractable, have unbounded stretch-width.

As $\mathrm{stw} \sqsubset \mathrm{cw}$ [**19**], this makes stretch-width a good candidate for a generalization of clique-width still allowing parameterized algorithms for a large fragment of monadic second-order model checking. And indeed, this was established within bounded-degree graphs [**19**].

3. Separation of the reduced parameters

In this section, we establish that none of our reduced parameters are functionally equivalent. We give a generic way of building families with bounded $p^{\downarrow}$ but unbounded $q^{\downarrow}$ from families with bounded p and unbounded q, when p and q are graph parameters respecting some mild properties. This works by forcing a particular red graph to appear in any contraction sequence of low width. We then set these particular red graphs as those with bounded p and unbounded q.

For any graph G, let $\mathit{red}(G)$ be the trigraph with red graph G, and no black edge.

Lemma 6.1 ([**27**], adapted from [**10**])**.** *For any connected graph H of maximum degree d with no pair of twins, there is a graph G such that*

- *G admits a partial d-sequence to red(H) such that all connected components of red graphs of the sequence are isomorphic to subgraphs of H, and*
- *every $|V(H)|$-sequence of G goes through a trigraph that admits red(H) as an induced subtrigraph.*

SKETCH. Assume that $V(H) = [|V(H)|]$. To build G, replace every vertex v of H by a clique $C^v = \{a_1^v, \dots, a_h^v\}$ with $h := |V(H)|^{|V(H)|} + |V(H)|$, and every edge $vw \in E(H)$ with $v < w$ by a half-graph between A^v and A^w consisting of all the edges $a_i^v a_j^w$ such that $i \leqslant j$.

The partial sequence from G to H is as follows. For i going from 1 to $h-1$, contract a_i^v and a_{i+1}^v for every $v \in V(H)$, and still denote by a_{i+1}^v the resulting vertex. It can be seen that the non-singleton connected components of the red graphs are isomorphic to subgraphs of H; actually, H itself after the first iteration is completed.

For the second item, we first notice that, while each A^v has more than $|V(H)|$ vertices, contractions have to be done within cliques A^v, since H has no pair of twins. If contractions have happened in each A^v, the current red graph does admit H as induced subtrigraph. We conclude since getting down to $|V(H)|$ vertices in one A^v requires, in a $|V(H)|$-sequence, to have contracted in every clique $A^{v'}$. This last bit is proven with the same argument as in Theorem 5.3, and requires H be connected. □

To show that the inclusions among our reduced parameters are strict, namely,

$$\Delta^{\downarrow} \sqsubset (\mathrm{tw} + \Delta)^{\downarrow} \sqsubset \mathrm{cutw}^{\downarrow} \sqsubset \mathrm{bandw}^{\downarrow} \sqsubset \star^{\downarrow},$$

we simply invoke Lemma 6.1 with the graph H ranging over the set of grids, full binary trees (which we 1-subdivide to remove twins), subdivisions of combs, and paths. The four obtained families of graphs G make the four separations. Indeed if $p^{\downarrow}$, $q^{\downarrow}$ is the pair to separate, with $p \sqsubset q$, the lemma ensures that $q^{\downarrow}$ is unbounded, and that we have a partial sequence with red graphs of bounded p from G to red(H). We only need to require that p can only decrease on subgraphs and when isolated vertices are added, and that q can only decrease on induced subgraphs, and grows with the maximum degree. We finally check that red(H) admits a contraction sequence all red graphs of which have

- bounded maximum degree for the grids;
- bounded $(\mathrm{tw} + \Delta)$ for the 1-subdivision of the full binary trees;
- bounded cutwidth for the comb subdivisions;
- bounded bandwidth for the paths.

CHAPTER 7

Algorithmic Applications

In this chapter we see how to solve problems faster, or approximate them better, on graphs of bounded twin-width as compared to general graphs. These algorithms require that a d-sequence of the input graph be given. For almost all[1] the classes that we have shown to be of bounded twin-width, we also know how to find a contraction sequence of constant width in polynomial time. Admittedly, this may worsen the running time of our linear algorithms.

But more problematically, it is still an open question—the most pressing one on this very topic—whether twin-width can be efficiently approximated. The humblest version of this question is as follows. Are there functions f, g and an algorithm that, on input graphs G of twin-width d, outputs an $f(d)$-sequence of G in time $|V(G)|^{g(d)}$? It is known, however, that deciding if the twin-width of a graph is at most 4 is NP-complete [**9**]. But the previous question with, say, $f(d) = 2d$ is wide open.

Section 1 presents fixed-parameter tractable algorithms for monadic second-order model checking in graphs of bounded component twin-width, and for first-order model checking in graphs of bounded twin-width. These algorithms are based on dynamic programming along the contraction sequences. In Section 2, we get approximation algorithms for some of the classic combinatorial optimization problems, Min Dominating Set, Max Independent Set, Min Coloring, with improved approximation factors over what can be done in general graphs. The common ingredient is the use of balanced contraction sequences. In Section 3, twin-decompositions are leveraged to compute shortest-path trees in n-vertex m-edge graphs of low twin-width within time $O(n \log n)$, possibly sublinear in m.

Chapter 10 contains more algorithms on binary structures of low twin-width, most of which do *not* require a given contraction sequence, as well as algorithms on classes of *unbounded* twin-width benefiting from twin-width theory.

[1]The only exception is the construction of expanders based on 2-lifts.

1. Parameterized algorithms

In this section, when claiming $O_d(n)$ running times, we assume that the d-sequence is given in some read-only input tape, and permits $g(d,r)$-time access to the subtrigraph H induced by the distance-r neighborhood in the red graph of any newly formed vertex/part, with $g(d,r) = O(|V(H)|^2)$. Alternatively, we could simply assume that only these induced subtrigraphs are part of the input, as their total size is $g(d,r) \cdot n$. If we are only given the sequence of pairs to contract, an extra $O(m)$ additive term is needed to actually compute these induced subtrigraphs, where m denotes the number of edges of the input.

1.1. In classes of bounded component twin-width. In this subsection and the next, the general result on model checking is preceded by a particular case. This serves two purposes: presenting the ideas in their simplest form, and showcasing that algorithms based on contraction sequences need not be slow. The latter is worth mentioning in light of the implicit tetrational running times of Theorems 7.2 and 7.4. The 3-Coloring problem asks if the input graph can be properly colored using at most three colors.

Theorem 7.1 ([25]). *There is an algorithm solving* 3-Coloring *in time $O(7^d d^2 n)$ in n-vertex graphs given with a partition sequence witnessing that their component twin-width is at most d.*

Proof. Let G be the input graph, and $\mathcal{P}_n, \dots, \mathcal{P}_1$ be a partition sequence of G. A *profile* of a connected component C of $\mathcal{R}(\mathcal{P}_i)$ is a function $\lambda : V(C) \to 2^{\{1,2,3\}} \setminus \{\emptyset\}$ such that there is a proper 3-coloring c of $G[\bigcup_{P \in V(C)} P]$ satisfying $c(P) = \lambda(P)$ for every $P \in V(C)$. Thus, G is 3-colorable if and only if the unique connected component of $\mathcal{R}(\mathcal{P}_1)$, $\{V(G)\}$, admits a profile.

We determine if the latter holds by maintaining, via dynamic programming, every profile of every connected component for each red graph ranging from $\mathcal{R}(\mathcal{P}_n)$ to $\mathcal{R}(\mathcal{P}_1)$. As $\mathcal{R}(\mathcal{P}_n)$ is edgeless, the vertex sets of its connected components are of the form $\{\{v\}\}$ for each $v \in V(G)$. Each has three profiles: $\lambda : \{v\} \mapsto \{1\}$, $\lambda : \{v\} \mapsto \{2\}$, and $\lambda : \{v\} \mapsto \{3\}$. Now we only need to describe how all the profiles of $\mathcal{R}(\mathcal{P}_i)$ can be computed, knowing all the profiles of $\mathcal{R}(\mathcal{P}_{i+1})$.

Say, $\mathcal{P}_i$ results from the fusion of $P, P' \in \mathcal{P}_{i+1}$, and let C be the connected component of $\mathcal{R}(\mathcal{P}_i)$ containing the part $P \cup P'$. Since $|V(C)| \leqslant d$, at most $d+1$ connected components of $\mathcal{R}(\mathcal{P}_{i+1})$, say,

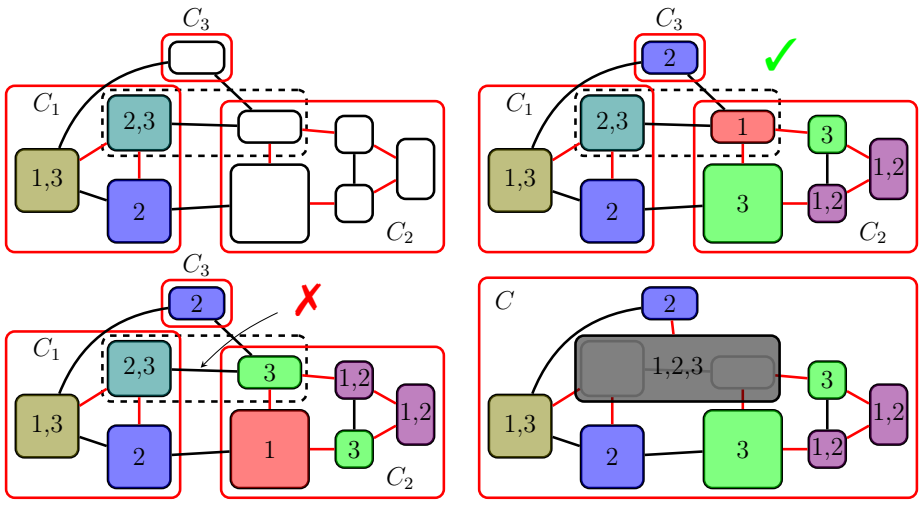


FIGURE 7.1. Example of a profile of connected component C_1 (top left), compatible profiles of C_1, C_2, C_3 (top right) yielding a profile of C (bottom right), and incompatible profiles (bottom left).

$C_1, \ldots, C_h$, gather to form C; see Figure 7.1. Let X_j be the computed set of profiles of C_j, for each $j \in [h]$.

For each $(\lambda_1, \ldots, \lambda_h) \in X_1 \times \ldots \times X_h$, let $\lambda' = \uplus_{j \in [h]} \lambda_j$. If no pair $P_1, P_2 \in \mathrm{dom}(\lambda')$ linked by a black edge is such that $\lambda'(P_1) \cap \lambda'(P_2) \neq \emptyset$, then we add λ to the profiles of C, where λ has domain $V(C)$, and is defined by $\lambda(Q) = \lambda'(Q)$ for every $Q \in \mathrm{dom}(\lambda') \setminus \{P, P'\}$, and $\lambda(P \cup P') = \lambda'(P) \cup \lambda'(P')$. Otherwise, we simply discard λ'.

In addition, we can store a witnessing proper 3-coloring c^λ for each profile λ, and eventually return c^λ for a profile λ of $\mathcal{R}(\mathcal{P}_1)$, if one exists. The algorithm is sound, since the way we filter out unions of profiles, the coloring $\uplus_{j \in [h]} c^{\lambda_j}$ cannot have a monochromatic edge. It is complete, as one can consider the eventual aggregation in a single part of the n profiles $\lambda^1 : \{v_1\} \mapsto \{c(v_1)\}, \ldots, \lambda^n : \{v_n\} \mapsto \{c(v_n)\}$ where $V(G) = \{v_1, \ldots, v_n\}$ and c is a proper 3-coloring of G.

At each contraction, we consider at most 7^{d+1} functions λ'. For each, checking the absence of black edges between intersecting sets of colors takes $O(d^2)$ time. Thus the overall running time is $O(7^d d^2 n)$, possibly much lower than the number of edges of G. □

We now extend Theorem 7.1 to the whole monadic second-order model checking. We fix some finite binary signature Σ, and denote by $MSO_k(\Sigma)$ the set of MSO sentences over Σ with quantifier rank

at most k. Here, we call *rank-k type* of a Σ-structure G the set of sentences

$$\text{mso-tp}_k(G) := \{\varphi \in \text{MSO}_k(\Sigma) \ : \ G \models \varphi\}.$$

We define $FO_k(\Sigma)$ and *fo-tp*$_k(G)$ analogously with first-order logic. We fix some integer d such that the component twin-width of G is at most d, witnessed by the partition sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$. Let Σ^{+d} be Σ augmented with d unary relation symbols. The *component rank-k type* of a Σ-structure H whose domain is partitioned by $U_1, \dots, U_d$ is the set of sentences

$$\text{ctp}_k(H, U_1, \dots, U_d) := \{\varphi \in \text{MSO}_k(\Sigma^{+d}) \ : \ H, U_1, \dots, U_d \models \varphi\}.$$

We revisit the scheme of Theorem 7.1 by maintaining component rank-k types instead of mere 3-colorings. Up to logical equivalence, the number of rank-k types over a finite signature is tetrational, and allowing repetitions they can be listed in tetrational time; see for instance [**67**, Proposition 7.5].

For $i \in [n-1]$, let D_i be the vertex set of the connected component in $\mathcal{R}(\mathcal{P}_i)$ containing the part stemming from the contraction in $\mathcal{P}_{i+1}$. Our goal is to compute the component rank-k type of $H_i := G[\bigcup_{P \in D_i} P]$ partitioned by D_i, for every $i \in [n-1]$. The component rank-k type of singletons is easy to compute, and that of D_1 coincides with mso-tp$_k(G)$. Thus our task boils down again to deriving the component rank-k types of H_i, D_i based on those previously computed.

Here and in the next subsection, we rely on the Ehrenfeucht–Fraïssé game characterizations of types. Let us start with the variant for first-order logic. In the *Ehrenfeucht–Fraïssé game*, or EF game, two players *Spoiler* and *Duplicator* confront each other over two Σ-structures $\mathcal{A}$ and $\mathcal{B}$, with domain A and B, respectively. They play a succession of rounds, where Spoiler wants to show that $\mathcal{A}$ and $\mathcal{B}$ are not isomorphic, whereas Duplicator tries to argue the opposite. At the i-th round, Spoiler chooses a structure $\mathcal{A}$ or $\mathcal{B}$, 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 k rounds, $a_i \mapsto b_i$ (for $i \in [q]$) is still an isomorphism between the induced substructures $\mathcal{A}[a_1, \dots, a_k]$ and $\mathcal{B}[b_1, \dots, b_q]$, Duplicator has *survived k rounds* of the EF game.

The *MSO-EF game* is similar to the EF game, but Spoiler can additionally decide to play a subset of A (resp. of B), to which Duplicator answers with a subset of B (resp. of A). Now after k rounds, two tuples of e elements have been played, say $(a_1, \dots, a_e) \in A^e$ and

$(b_1, \ldots, b_e) \in B^e$, as well as two tuples of s sets, say $(A_1, \ldots, A_s) \in (2^A)^s$ and $(B_1, \ldots, B_s) \in (2^B)^s$, with $k = e + s$. Duplicator has survived these k rounds if $a_i \mapsto b_i$ (for $i \in [e]$) is an isomorphism between $(\mathcal{A}, A_1, \ldots, A_s)[a_1, \ldots, a_e]$ and $(\mathcal{B}, B_1, \ldots, B_s)[b_1, \ldots, b_e]$.

We write $\mathcal{A} \equiv_k^{FO} \mathcal{B}$ (resp. $\mathcal{A} \equiv_k^{MSO} \mathcal{B}$) if Duplicator has a strategy in the EF game (resp. MSO-EF game) to survive k rounds. The Ehrenfeucht–Fraïssé theorem states that this is equivalent to having equal rank-k types.

Lemma 7.1 ([**67**], Theorem 3.9 and Corollary 7.8)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be two Σ-structures, and $\mathcal{L} \in \{FO, MSO\}$. Then, $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of $\mathcal{L}_k(\Sigma)$ if and only if $\mathcal{A} \equiv_k^{\mathcal{L}} \mathcal{B}$.*

We can now indicate how the following theorem is obtained.

Theorem 7.2 ([**25**])**.** *For any binary signature Σ, monadic second-order model checking over Σ can be solved in time $f(d, k) \cdot n$ on n-vertex Σ-structures given with a partition sequence witnessing component twin-width d, and sentences of quantifier rank k, for some tetrational function f.*

SKETCH. Let us recall that our goal is to determine the component rank-k type of H_i, D_i, for i going from $n-1$ down to 1. Let $P \cup P'$ be the newly created part in $\mathcal{P}_i$. We observe that D_i results from the gathering of up to $d+1$ connected components $C_1, \ldots, C_h$, where for every $p \in [h]$, $V(C_p)$ is some D_j with $j > i$. Let G' be the substructure of G induced by $\{v \ : \ p \in [h], v \in Q \in V(C_p)\}$; see Figure 7.2.

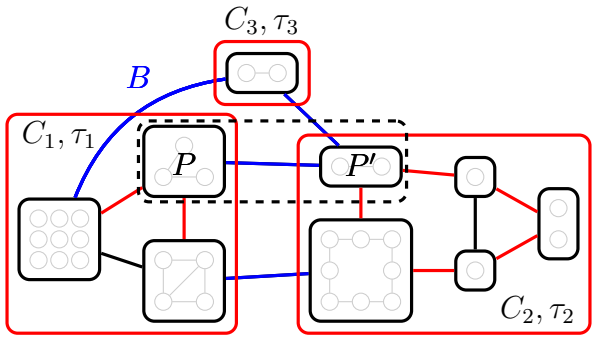


FIGURE 7.2. The Σ-structure G' partitioned by the parts of the connected components C_1, C_2, C_3 in $\mathcal{R}(\mathcal{P}_{i+1})$, unifying to component D_i after the merge of P, P'.

For each $p \in [h]$, we set

$$J_p := G\left[\bigcup_{Q \in V(C_p)} Q\right] \text{ and } \mathcal{P}^p := \{U_1^p, \dots, U_d^p\},$$

the partition in the parts of $V(C_p)$, with $U_z^p = \emptyset$ when $z > |V(C_p)|$. Assume that there is a Σ-structure H, a partition $\mathcal{P}$ of $V(H)$, and a trigraph isomorphism ψ from $G'/\{Q : p \in [h], Q \in V(C_p)\}$ to $H/\mathcal{P}$, such that for each $p \in [h]$, $J_p, \mathcal{P}^p$ has the same component rank-k type, say τ_p, as

$$H_p := H\left[\bigcup_{z \in [d]} \psi(U_z^p)\right] \text{ and } \mathcal{Q}^p := \{\psi(U_1^p), \dots, \psi(U_d^p)\},$$

extending ψ with $\psi(\emptyset) = \emptyset$.

We claim that the component rank-k type, say τ, of H_i, $D_i := \{P \cup P', P_1, \dots, P_s\}$, is determined by the data of $P, P', \tau_1, \dots, \tau_h$, and B, the black edges (or binary atomic types, if Σ has several binary relation symbols) of $G/\mathcal{P}_{i+1}$ linking parts from two distinct $V(C_p), V(C_{p'})$. We thus need to argue that H partitioned by $\{\psi(P) \cup \psi(P'), \psi(P_1), \dots, \psi(P_s)\}$ also has component rank-k type τ.

For this, we invoke Lemma 7.1 and use the *compositionality* of games. Duplicator wins by answering a vertex move in $V(J_p)$ (resp. $V(H_p)$) by its winning reply in $V(H_p)$ (resp. $V(J_p)$), which exists since $J_p, \mathcal{P}^p$ and $H_p, \mathcal{Q}^p$ have the same component rank-k type, and a subset move, by the union of the winning replies to its projections on each $V(H_p)$ (resp. $V(J_p)$). □

Theorem 7.2, when phrased with clique-width rather than component twin-width, is known as the Courcelle–Makowsky–Rotics theorem [**38**]. It generalizes the celebrated Courcelle's theorem: that monadic second-order model checking on incidence graphs of bounded treewidth can be solved in fixed-parameter linear time [**37**].

1.2. In classes of bounded twin-width. Again, we start with a simple problem, k-Independent Set, which seeks an independent set of size at least k. In general graphs, for any function f, an $f(k)n^{O(1)}$-time algorithm is unlikely to exist for this problem [**40**].

Theorem 7.3 ([**21**]). *k-Independent Set can be solved in time $O(d^{2k}k^2n)$ in n-vertex graphs given with a d-sequence.*

PROOF. Let $\mathcal{P}_n, \ldots, \mathcal{P}_1$ be a partition sequence of width d of the input graph G. Our plan is to store for each connected set X of size at most k in each $\mathcal{R}(\mathcal{P}_i)$, an independent set I_X of $G[\bigcup_{P \in X} P]$ with maximum cardinality among those that intersect every part $P \in X$. As soon as some computed I_X has size at least k, the algorithm outputs it and terminates.

As $\mathcal{R}(\mathcal{P}_n)$ is edgeless, its only connected sets are $\{\{v\}\}$ for $v \in V(G)$, and $I_{\{\{v\}\}} = \{v\}$. The only connected set of $\mathcal{R}(\mathcal{P}_1)$ is $\{V(G)\}$, and G is a no-instance if and only if $I_{\{V(G)\}}$ is computed and has size less than k. In which case, as a consolation prize, $I_{\{V(G)\}}$ is a maximum independent set of G.

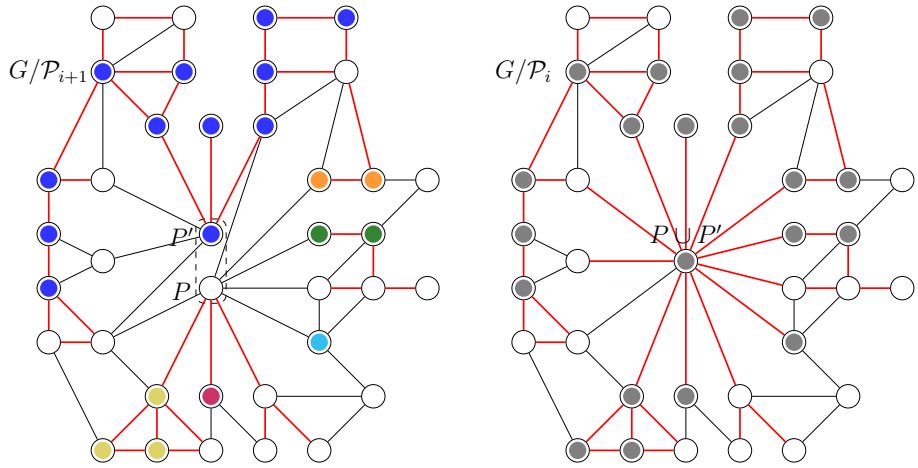


FIGURE 7.3. The set X (in gray) splits into subsets $X_1, \ldots, X_h$ (each in a different color). The option depicted is *intersecting* P' *but not* P, which is here the only one that may give rise to a partial solution.

Once again, the crux is to efficiently compute the partial solutions in $\mathcal{P}_i$ from those established at an earlier stage. Say P, P' are merged in $\mathcal{P}_{i+1}$ to make $\mathcal{P}_i$. The only *new* connected sets X of $\mathcal{R}(\mathcal{P}_i)$ are those containing $P \cup P'$.

When looking in the previous partition $\mathcal{P}_{i+1}$, there are three ways of intersecting $P \cup P'$: by intersecting P but not P', intersecting P' but not P, and intersecting both. For each option, X splits into at most $d+2$ connected sets $X_1, \ldots, X_h$ of $\mathcal{R}(\mathcal{P}_{i+1})$; see Figure 7.3. We set I_X to $I_{X_1} \cup \ldots \cup I_{X_h}$ for the option maximizing the cardinality of this set among those for which no pairs of parts in $\bigcup_{p \in [h]} X_p$ are linked by

a black edge in $G/\mathcal{P}_{i+1}$, and no I_{X_p} is the `nil` symbol. It may happen that none of the three options work, in which case we set $I_X := \texttt{nil}$.

The running time is as claimed since there are at most d^{2k} connected sets of size at most k containing a fixed vertex in a graph of maximum degree at most d. The algorithm is sound since every partial solution is the union of pairwise non-adjacent partial solutions. It is complete since any independent set of G of size k ends up in a connected set of size at most k in some $\mathcal{R}(\mathcal{P}_i)$. □

Similar parameterized algorithms can be designed for other problems such as k-Dominating Set, k-Subgraph Isomorphism or Induced k-Subgraph Isomorphism with only slightly worse running times [**21**]. These problems are defined, given an input graph G, as the search for a dominating set of G of size at most k, and with an additional k-vertex graph H, as whether H is isomorphic to a subgraph of G, or to an induced subgraph of G, respectively.

The *Exponential-Time Hypothesis* (or ETH) asserts that there is a $\lambda > 0$ such that solving n-variable 3-SAT requires time λ^n. Under this widely-believed assumption, Theorem 7.3 and the parameterized algorithms for k-Dominating Set, k-Subgraph Isomorphism, and Induced k-Subgraph Isomorphism have essentially-optimal running times. An algorithm in time $2^{o(k/\log k)}n^{O(1)}$, or in time $2^{o(n/\log n)}$ for that matter, for any of these problems would refute the ETH.

These parameterized algorithms can be lifted, admittedly at the cost of a much higher running time, to the whole first-order model checking. We fix the quantifier rank k of the first-order sentences we wish to model check, and the binary signature Σ. Given a partition $\mathcal{P}$ of the domain of a Σ-structure G, we say that a tuple $(P_1, \dots, P_q) \in \mathcal{P}^q$ with $q \leqslant k$ is *local* (at P_1) if for every $j \in [2, q]$, part P_j is at distance at most 2^{k-q} of some part of $\{P_1, \dots, P_{j-1}\}$ in $\mathcal{R}(G/\mathcal{P})$.

A *local sentence* at P_1 in $G, \mathcal{P}$ is a first-order sentence of the form $Q_1 x_1 \in P_1 \cdots Q_q x_q \in P_q\ \phi$, where $Q_j \in \{\exists, \forall\}$, ϕ is quantifier-free Σ^{+q}-formula, and $(P_1, \dots, P_q)$ is local. The *local rank-k type* of $G, \mathcal{P}$ at $P \in \mathcal{P}$, denoted by $ltp_k(G, \mathcal{P}, P)$, is the set of local sentences at P in $G, \mathcal{P}$ that are true in $G, \mathcal{P}$. We can now sketch a proof of the main algorithmic result related to twin-width, following [**49**].

Theorem 7.4 ([**26**]). *There is a tetrational function f, and an algorithm that inputs a d-sequence of an n-vertex binary structure over signature Σ, and a sentence $\varphi \in FO(\Sigma)$ of quantifier rank k, and decides $G \models \varphi$ in time $f(d,k) \cdot n$.*

SKETCH. The plan is as in Theorem 7.2 replacing component rank-k types with local rank-k types. Let $\mathcal{P}_n, \dots, \mathcal{P}_1$ be a partition sequence of G with width d. The reader can check that, again, computing the local rank-k types in $G, \mathcal{P}_n$ is easy, and that the local rank-k types in $G, \mathcal{P}_1$ coincide with fo-tp$_k(G)$.

So we mainly need to check that the local rank-k types in $G, \mathcal{P}_i$ can be computed from those already established. Intuitively, we are combining the proofs of Theorems 7.2 and 7.3, using the tailored notion of local rank-k type. The merge in $\mathcal{P}_{i+1}$ yielding $\mathcal{P}_i$ may bring closer in the red graph pairs of parts that were until then far apart. That the local rank-k types in $G, \mathcal{P}_{i+1}$ suffice to get the new local types can be shown with Ehrenfeucht–Fraïssé games, and Lemma 7.1.

Duplicator survives k rounds by splitting the game in a *disjoint union* of *local* games. Say that q moves were already played, and that the local games are $J_1, \dots, J_h$. If Spoiler plays *at distance* at most 2^{k-q} of some J_p, Duplicator follows her winning strategy in this local game, as computed in $G, \mathcal{P}_{i+1}$. Otherwise, Duplicator starts a new local game J_{h+1}. Crucially, as 2^{k-q} is divided by 2 after each move, Spoiler cannot confuse Duplicator by playing close to more than one local game. □

2. Approximation algorithms

The MIN DOMINATING SET problem seeks, given a graph G, a subset $D \subseteq V(G)$ of minimum cardinality such that $N_G[D] = V(G)$, called *dominating set*. On n-vertex graphs MIN DOMINATING SET admits polynomial-time $\ln n$-approximation algorithms [**60, 68**]. This is best possible, as $(1-\varepsilon)\ln n$-approximating this problem is NP-hard, for any $\varepsilon > 0$ [**39**].

We obtain a constant-factor polynomial-time approximation algorithm on graphs of bounded twin-width. This is done by extracting in a versatile tree of contractions a partial partition sequence maintaining low weight of its parts, after vertices are weighted by a fixed optimum solution of the linear relaxation of MIN DOMINATING SET.

Theorem 7.5 ([21]). MIN DOMINATING SET *has a polynomial-time $O_d(1)$-approximation on graphs given with a d-sequence.*

PROOF. Let G be a non-empty input graph. We compute w^* an optimum solution to the linear program

$$\text{minimize} \sum_{u \in V(G)} w(u) \text{ subject to}$$

$$\forall u \in V(G), \sum_{v \in N_G[u]} w(v) \geqslant 1, \text{ and } 0 \leqslant w(u) \leqslant 1,$$

and $\gamma^* := \sum_{u \in V(G)} w^*(u) \geqslant 1$. Theorem 4.5, the functional equivalence of twin-width and versatile twin-width, is effective. This yields a polynomial-time subroutine finding a maximal partial partition sequence $\mathcal{S}$ of width $d' = f(d)$, the versatile twin-width of G, from the finest partition on G to $\mathcal{P}$, that never creates a part P with w^*-*weight* $w^*(P) := \sum_{u \in P} w^*(u) \geqslant \frac{1}{d'+1}$. Hence parts of at least this w^*-weight have to be singletons.

Partition $\mathcal{P}$ has at most $2d'(d'+1)\gamma^*$ parts, since otherwise at least one pair of the $\lfloor |\mathcal{P}|/d' \rfloor$ disjoint ones given by the versatility would make a part of w^*-weight less than $\frac{1}{d'+1}$, thereby contradicting the maximality of $\mathcal{S}$.

We thus arbitrarily pick one vertex v_P per part $P \in \mathcal{P}$ to define D a dominating set of G within factor $2d'(d'+1)$ of the optimum solution. We shall just argue that D indeed dominates every vertex of G. Every vertex $u \in P$ such that PP' is a black edge of $G/\mathcal{P}$, for some $P' \in \mathcal{P}$, is a neighbor of $v_{P'} \in D$ in G. The other vertices $u \in P$ are such that P has at most d' neighbors, all red. At least one vertex u' of $N_G[u]$ has to be such that $\{u'\}$ is a singleton part of $N_{G/\mathcal{P}}[P]$, otherwise $\sum_{v \in N_G[u]} w^*(v) < |N_{G/\mathcal{P}}[P]| \cdot \frac{1}{d'+1} \leqslant 1$. Such a u' is necessarily picked in D, as the only representative of its part. □

The previous algorithm can be adapted to the task of finding a largest subset of vertices pairwise at distance at least 3. However, this method does not extend to MAX INDEPENDENT SET, for which *at least 3* is replaced by *at least 2*. Nevertheless, we achieve some improved approximation algorithms for this problem via a different route, based on properly coloring the red graph, and exploiting the modular decomposition on each color class.

The Max Independent Set problem simply asks for an independent set of maximum cardinality. This problem is about as inapproximable as it gets: for any $\varepsilon > 0$, it is NP-hard to $n^{1-\varepsilon}$-approximate on n-vertex graphs [**55, 82**]. Furthermore, for any $r \leqslant \sqrt{n}$ and $\varepsilon > 0$, an r-approximation algorithm running in time $2^{n^{1-\varepsilon}/r^{1+\varepsilon}}$ would refute the ETH [**33**]. Both barriers can be breached within graphs of bounded twin-width.

Proposition 7.2 ([**10**])**.** Max Independent Set *admits an $O_d(1)$-approximation algorithm running in time $2^{O_d(\sqrt{n})}$ on n-vertex graphs given with a d-sequence.*

Proof. By Theorem 4.6, we compute in polynomial time a balanced $f(d)$-sequence $\mathcal{P}_n, \dots, \mathcal{P}_1$ of the input graph G. We only make use of $\mathcal{P} := \mathcal{P}_{\lceil\sqrt{n}\rceil}$, a partition leveling its number of parts, $\lceil\sqrt{n}\rceil$, with the maximum size among its parts, which is between $\sqrt{n} - 1$ and $f(d)\sqrt{n}$. Let $I_1, \dots, I_{f(d)+1}$ be a partition of $\mathcal{P}$ into independent sets of $\mathcal{R}(\mathcal{P})$. This partition exists since every graph of maximum degree Δ can be properly $\Delta + 1$-colored by the first-fit coloring.

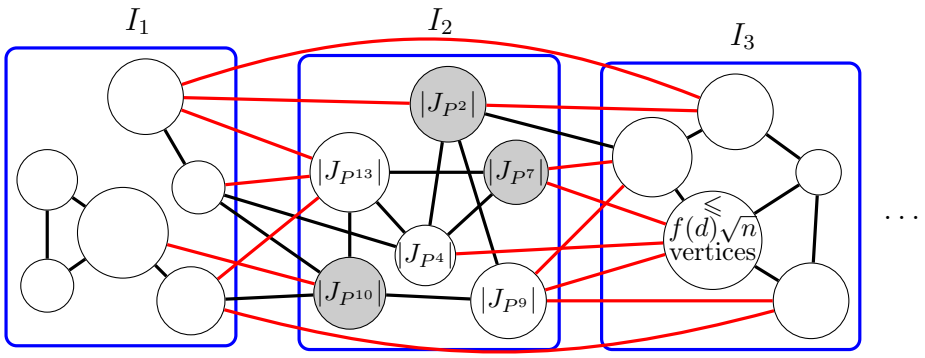


Figure 7.4. The trigraph $G/\mathcal{P}$. An independent set of maximum weight in $\mathcal{B}(G/\mathcal{P}[I_2])$ (shaded) yields a maximum independent set in G restricted to I_2.

For each $h \in [f(d) + 1]$, we can find a maximum independent set of $G[\bigcup_{P \in I_h} P]$ in time $2^{O_d(\sqrt{n})}$. Indeed for every $P \in I_h$, an exhaustive search solves Max Independent Set in $G[P]$ in time $2^{O(f(d)\sqrt{n})}$ and outputs, say, J_P. One then solves in time $2^{O(\sqrt{n})}$, again by exhaustive search, the weighted Max Independent Set on the black graph of $G/\mathcal{P}[I_h]$ (equivalently, its total graph since I_h induces no red edge, by design) where every vertex $P \in I_h$ has weight $|J_P|$, and outputs, say, J_h.

Then $S_h := \bigcup_{P \in J_h} J_P$ is the desired optimum solution for $G[\bigcup_{P \in I_h} P]$; see Figure 7.4. By the pigeonhole principle, at least one independent set S_h is at least as large as an $\frac{1}{f(d)+1}$ fraction of the optimum solution. □

Theorem 7.6 ([10]). *For any $\varepsilon > 0$,* Max Independent Set *admits a polynomial-time n^{ε}-approximation algorithm on n-vertex graphs given with an $O(1)$-sequence.*

sketch. In the previous algorithm, replace each exhaustive search by a recursive call. This is possible since these calls are on induced subgraphs. At depth q, this improves the running time to $\exp(O(n^{2^{-q}})) \cdot n^{O(1)}$ while degrading the approximation factor to $O(1)^{2^{q-1}}$. We obtain the desired result for some $q = O(\log(\varepsilon \log n))$. □

Variations on the algorithm of Proposition 7.2 lead to similar results for other problems such as Min Coloring and Max Induced Matching, both as inapproximable as Max Independent Set in general graphs [**33, 82**]. The Min Coloring problem asks for a proper coloring of the input graph using the minimum number of colors, while Max Induced Matching seeks the largest induced subgraph within which every vertex has degree exactly 1.

Theorem 7.7 ([10]). *For any $\varepsilon > 0$,* Min Coloring *and* Max Induced Matching *admit polynomial-time n^{ε}-approximation algorithms on n-vertex graphs given with an $O(1)$-sequence.*

In contrast with the results presented so far in this section, some optimization problems are (almost) as hard to approximate in graphs of bounded twin-width as they are in general graphs. This is the case of approximating the induced or non-induced longest paths, or finding as small as possible independent sets that are also dominating sets, called Min Independent Dominating Set. The reason for the former is that logarithmically-long subdivisions of general graphs have, by Theorem 5.2, bounded twin-width, while the latter uses that, by Lemma 3.4 and Theorem 4.8, so do substitutions of independent sets in planar graphs, on which a direct gap-introducing reduction from Planar 3-SAT to Min Independent Dominating Set can be designed [**10**].

It is interesting to note that the approximation factors of n^{ε} in Theorems 7.6 and 7.7 could be in principle improved. A polynomial-time approximation scheme (PTAS) for Max Independent Set is not

ruled out on graphs of bounded twin-width, and a known reduction based on self-substitutions would turn any constant-factor approximation algorithm into a PTAS [**21**].

3. Shortest paths

The Single-Source Shortest Paths problem inputs a graph G and a vertex $s \in V(G)$, and asks for a *shortest-path tree* rooted at s, that is, a spanning tree T of the connected component of G containing s, such that for every $t \in V(T)$, s and t are at the same distance from each other in T and in G.

Let us call *width* of a twin-decomposition the width of its corresponding contraction sequence. In particular we saw in Chapter 3 that, if a twin-decomposition $(T, <, B)$ of an n-vertex graph G has width d, then $|B| \leqslant (d+1)(n-1)$. Let us relabel the leaves of T such that the i-th leaf in the left-to-right order of some fixed planar embedding of T gets label $\{i\}$. From parents of the leaves to the root of T, let us label each parent with the union of the labels of its two children. This way, every node of T is labeled by a discrete interval that is the union of the labels at the leaves of its rooted subtree; see Figure 7.5.

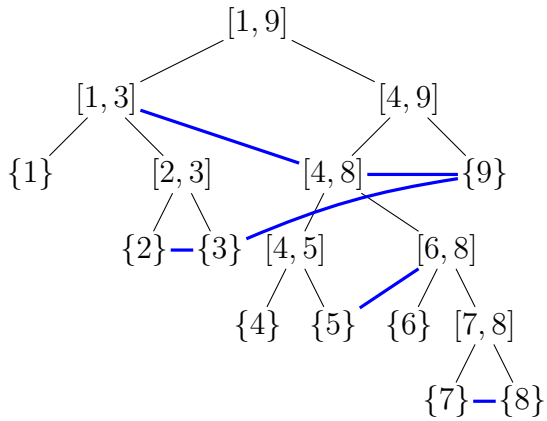


Figure 7.5. Example of a twin-decomposition labeled by discrete intervals, discarding the contraction order.

The original graph G can be described by at most $(d+1)(n-1)$ bicliques partitioning $E(G)$, which can be encoded as pairs of intervals. We call this an *interval biclique partition*. For instance, the interval biclique partition corresponding to the twin-decomposition of Figure 7.5 is $\{\{[1,3],[4,8]\},\{\{2\},\{3\}\},\{\{3\},\{9\}\},\{\{5\},[6,8]\},\{[4,8],\{9\}\},\{\{7\},\{8\}\}\}$. While every graph admits an edge partition into at most $n-1$

bicliques, not every n-vertex graph admits a vertex ordering and an edge partition into $O(n)$ bicliques whose sides are all intervals. We use interval biclique partitions to speed up the computation of shortest paths.

Theorem 7.8 ([21]). SINGLE-SOURCE SHORTEST PATHS *can be solved in time* $O(dn \log n)$ *in* n*-vertex graphs given with a twin-decomposition of width* d*.*

PROOF. We first compute B the corresponding interval biclique partition in $O(dn)$ time. We add the at most $2(d+1)(n-1)$ intervals that are sides of a biclique in an augmented tree T_B. For us, an *augmented tree* is a red-black tree, where nodes v are labeled by discrete intervals I_v, and possess two additional fields: the maximum value found within labels in the subtree rooted at v, and the list of intervals forming with I_v a biclique of B. We call *maximum* field the former, and *neighbor* field the latter. As a binary search tree, the sorting key is the label left endpoint; see Figure 7.6.

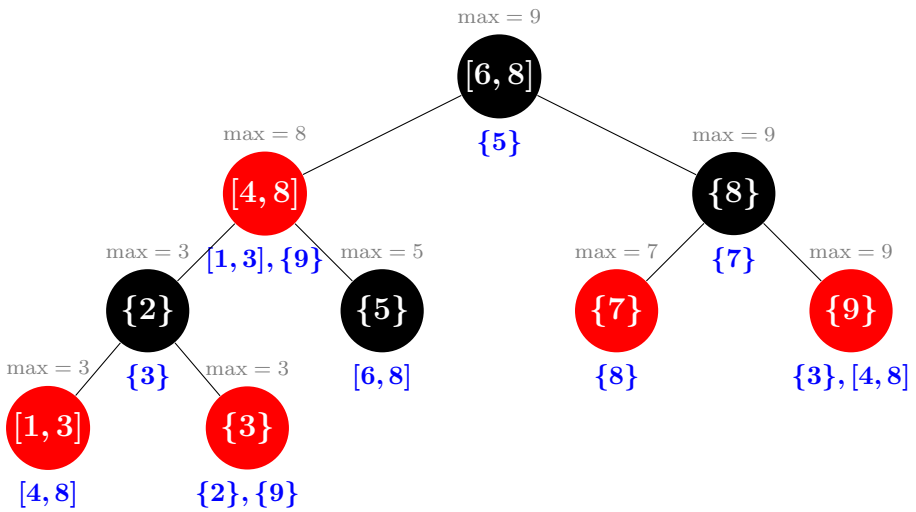


FIGURE 7.6. Augmented tree matching Figure 7.5. Each node has a label (inside the node), a *maximum* field (top, in gray), and a *neighbor* field (bottom, in blue).

The *maximum* field allows to find, given a query interval I, a label intersecting I (if one exists) in time $\log(O(dn)) = O(\log n)$. This is done, starting at the root, by reporting the current label if it intersects I,

moving to the left child v if the left endpoint of I is smaller or equal to the *maximum* field of v, and moving to the right child otherwise; see [**36**, Section 14.3: Interval trees].

Insertions and deletions are done like in plain red-black trees, except some *maximum* fields may need an update, and still takes $O(\log n)$ time. Thus reporting all the q labels intersecting a query interval I may be performed in $O(q \log n)$ time, by removing one by one the found intervals, and eventually adding them back. When making such queries, we will not add the intervals back, as we wish them removed. We observe in passing that this operation can be supported in optimal time $O(q + \log n)$ [**77**], but it would not improve the overall running time, up to a multiplicative constant factor.

Let $G, s \in V(G) = [n]$ be the input. We initialize a second augmented tree T_V whose nodes have labels $\{i\}$ for $i \in V(G)$, and solely have a *maximum* field. We also set a queue Q initially only containing s. We make a breadth-first search from s using the bicliques in B, rather than the individual edges of G.

While Q is non-empty, we do the following. We remove the head u of Q from T_V, set an empty list L_u of intervals, and query the list of intervals of T_B intersecting $\{u\}$. Each found interval is removed from T_B, after its *neighbor* field is appended to L_u. For every interval I of L_u, we query the list of singletons of T_V (open vertices) intersecting I. Each found singleton $\{v\}$ is removed from T_V, and v is added to the tail of Q. The parent of v in the eventual shortest-path tree is set to u; its distance to s is that of u plus 1.

The correctness relies on the fact that no shortest path takes more than two edges of a biclique. More precisely, if $\{X, Y\} \in B$ and $P = v_1, \ldots, v_h$ is a shortest path in G, then there is at most one i such that $v_i \in X$ and $v_{i+1} \in Y$. In other words, once the biclique is traversed from X to Y, the option $X \to Y$ can be erased, as we do when deleting the side X from T_B. It is however possible that the biclique is then traversed from Y to X, and we indeed leave the option $Y \to X$ open by not removing Y from T_B, nor shrinking its *neighbor* field.

Each node of T_V or T_B is responsible for at most $O(\log n)$ computation time, leading to its eventual deletion. Hence the overall running time is $O((n + 2|B|) \log n) = O(dn \log n)$. □

If we are only given a contraction sequence of the input m-edge graph, the corresponding twin-decomposition can be computed in $O(m)$ time. For Single-Source Shortest Paths this is too slow, as $O(m)$

is the time for computing a breadth-first search in any connected graph (of possibly large twin-width). However, this preprocessing is acceptable for All-Pairs Shortest Paths, which asks for the shortest-path distance between every two vertices. Hence we get the following algorithm, less demanding on its input representation.

Theorem 7.9 ([**21**])**.** All-Pairs Shortest Paths *can be solved in time* $O(dn^2 \log n)$ *in* n*-vertex graphs given with a* d*-sequence.*

For comparison, in general graphs the unweighted All-Pairs Shortest Paths essentially takes matrix-multiplication time. One may wonder if the diameter can be computed faster than calling All-Pairs Shortest Paths, when a d-sequence of the input graph is provided. Again, logarithmically-long subdivisions (which have bounded twin-width) entail that, like in general graphs, a truly subquadratic algorithm is unlikely, even one that only 1.499-approximates the diameter [**21**]. Nevertheless, if the diameter is constant, a d-sequence allows an $O_d(n)$-time algorithm by Theorem 7.4, as the problem is then definable by a first-order sentence.

CHAPTER 8

First-Order Logic and Twin-Width

Clique-width and its weakly sparse counterpart, treewidth, enjoy a deep connection to monadic second-order logic. As we recalled in the previous chapter, model checking this logic in graphs of bounded clique-width [**38**] or incidence graphs of bounded treewidth [**37**] is fixed-parameter tractable. Moreover the family of classes of bounded clique-width is closed under MSO transductions. In Section 1, we show that the same holds with twin-width and first-order transductions. This was already announced in Chapter 5 as a powerful way of establishing that a class has bounded twin-width. Besides, it is useful directly or indirectly in all the results of the current chapter.

Classes of bounded clique-width (resp. linear clique-width) coincide with MSO transductions of a very simple graph class, that of trees (resp. paths) [**35**]. No similar result can hold for twin-width since grids have bounded twin-width, their MSO transductions contain the class of all graphs, and transductions compose. However, the previous theorems can be expressed in terms of first-order transductions. Classes of bounded clique-width (resp. linear clique-width) are the transductions of tree orders (resp. linear orders) [**35**]. This can further be phrased in terms of permutation classes. Denoting by $Av(\sigma)$ the proper permutation class containing all the permutations except those having σ as a pattern, linear orders can replaced by Av(21), i.e, the class of all identity permutations, and tree orders, by Av(231). We will see in Section 2 that a class of binary structures has bounded twin-width if and only if a proper permutation class transduces it. This can be strengthened. There is a fixed permutation σ—not nearly as nice as 21 or 231 with the current proof—such that classes of bounded twin-width are the transductions of Av(σ); see Table 1.

The parallel between clique-width & MSO and twin-width & FO breaks (in disfavor of the latter duo) as the efficient algorithm for model checking in classes of bounded clique-width or bounded treewidth goes almost as far as possible [**66**]. On the contrary there are classes, like bounded-degree graphs, of unbounded twin-width yet admitting

Bounded	Fast model checking	FO transduction of
linear clique-width	MSO	linear orders, Av(21)
clique-width	MSO	tree orders, Av(231)
(effective) twin-width	FO	Av(σ)

TABLE 1. Parallel between clique-width and twin-width.

a fixed-parameter tractable first-order model checking. More generally *monadically stable* classes, i.e, those which cannot transduce the class of all linear orders, admit such an algorithm [**41**]. It is in fact expected that this extends to every monadically dependent class, and is known to not go any further [**41**]. It is challenging to name a natural monadically dependent class that is not of effectively bounded twin-width[1] nor monadically stable. Up to the naturalness requirement, interpretations of classes where the clique-width is bounded by a function of r in every distance-r neighborhood are such an example, and one for which an efficient FO model checking algorithm is known [**18**]. In Section 3 we explore on which classes $\mathcal{C}$, *bounded twin-width* and *monadic dependence* coincide for every subclass of $\mathcal{C}$. We call these classes $\mathcal{C}$ delineated.

1. First-order transductions preserve bounded twin-width

We get the following theorem essentially as a by-product of the parameterized algorithm for first-order model checking of Chapter 7.

Theorem 8.1 ([**26**])**.** *Let $\mathcal{C}, \mathcal{D}$ be two classes of binary structures such that $\mathcal{C}$ has bounded twin-width and transduces $\mathcal{D}$. Then $\mathcal{D}$ has bounded twin-width.*

PROOF. By Lemma 3.10, a finite monadic lift of a class of bounded twin-width had bounded twin-width. Thus we shall simply prove the theorem when $\mathcal{C}$ interprets $\mathcal{D}$, and $\mathcal{D}$ has only binary relations. Let Σ, Γ be the vocabularies of $\mathcal{C}, \mathcal{D}$, respectively.

Let $\varphi_1(x,y), \dots, \varphi_h(x,y)$ describe this simple interpretation I, that is, $\varphi_j(x,y)$ is a Σ-formula interpreting the j-th (binary) relation of Γ. We set $k := \max_{j\in[h]} \operatorname{qr}(\varphi_j)$. For any $G \in \mathcal{C}$, let $\mathcal{P}_n, \dots, \mathcal{P}_1$ be a partition sequence of G of width at most $d := \operatorname{tww}(\mathcal{C})$. We define *local rank-$k$ s-types* similarly to local rank-k types with first-order formulas having s free variables instead of sentences. For each $i \in [n]$, let $\mathcal{P}'_i$

[1]A class $\mathcal{C}$ has *effectively bounded twin-width* if there is a polynomial-time algorithm that inputs a graph $G \in \mathcal{C}$ and outputs an $O(1)$-sequence of G.

refine partition $\mathcal{P}_i$ by splitting every $P \in \mathcal{P}_i$ into equivalence classes of local rank-$(k+1)$ 1-types. The upper bound on the number of types implies that $\mathcal{P}'_i$ is an $f(d,k)$-refinement of $\mathcal{P}_i$ for some tetrational function f. Thus for every $i \in [n]$, the partition $\mathcal{P}'_i$ over $\mathsf{I}(G)$ has width at most $f(d,k) \cdot d^{2^{k+1}}$. We conclude by Lemma 3.8. □

Theorem 8.1 is handy to extend the family of classes with known bounded twin-width. For instance, one can combine Theorem 8.1 and Theorem 4.8 to derive that squares of planar graphs have bounded twin-width. Bounded-degree segment graphs is another example of a class for which the only known upper bound on the twin-width uses Theorem 8.1.

Proposition 8.1. *Every subclass of segment graphs with bounded maximum degree has bounded twin-width.*

Proof. From a geometric representation of the segment graph, we build a (bipartite) planar graph H, with one vertex per segment intersection, one vertex per truncated segment, defined as path-component once the intersections are removed, and every edge between an intersection and incident truncated segments.

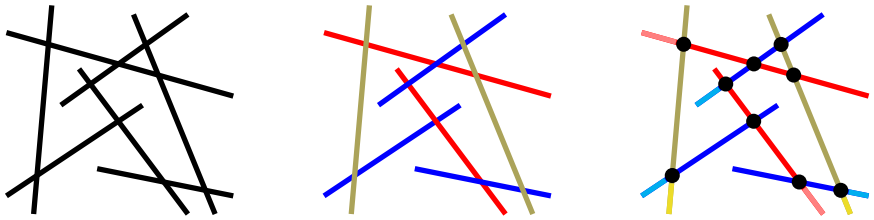


Figure 8.1. Geometric representation (left), proper coloring of the segments (center), and intersection vertices and appropriate coloring of the unary expansion (right).

As planar graphs have bounded twin-width, we simply need to present a transduction that *reconstructs* any segment graph of maximum degree at most Δ, from *one* corresponding planar graph H. For the monadic lift of the transduction, we use $2(\Delta+1)+1$ unary relations. We get a proper $\Delta+1$-coloring c of the segments. The first unary relation X is interpreted as the set of vertices of H corresponding to intersections. Each other vertex of H (truncated segment) gets the color of its including segment assigned by c, but one extremity (between the segment endpoint and its first intersection) gets a lighter version of this color; see right of Figure 8.1.

The simple interpretation sets the new domain as the light-colored vertices. We define the following formula for every pair of colors, here (blue, red), and every pair $i, j \in [\Delta]$:

$$\varphi_{i,j}^{\bullet,\bullet}(x,y) = \text{light-blue}(x) \wedge \text{light-red}(y) \wedge$$
$$\exists x_1 \exists x'_1 \ldots \exists x_{i-1} \exists x'_{i-1} \exists x_i \ \exists y_1 \exists y'_1 \ldots \exists y_{j-1} \exists y'_{j-1} \exists y_j \ E(x,x_1) \wedge E(y,y_1)$$
$$\wedge\, x_i = y_j \wedge \bigwedge_{h\in[i]} X(x_h) \wedge \bigwedge_{h\in[i-1]} \text{blue}(x'_h) \wedge \bigwedge_{h\in[j]} X(y_h) \wedge \bigwedge_{h\in[j-1]} \text{red}(y'_h)$$
$$\wedge \bigwedge_{h\in[i-1]} E(x_h,x'_h) \wedge E(x'_h,x_{h+1}) \wedge \bigwedge_{h\in[j-1]} E(y_h,y'_h) \wedge E(y'_h,y_{h+1}).$$

Note that this formula is satisfied by a pair of light-colored truncated segments such that the i-th intersection of the first including segment coincides with the j-th intersection of the second including segment. The proper coloring of the segments ensure that the two sequences of truncated segments do not deviate from their original including segment. For the edge set of the segment graph, we take the disjunction $\varphi(x,y)$ of the above formula for every pair of colors and every $i, j \in [\Delta]$. The latter range is sufficient as every segment is intersected at most Δ times. □

With a more elaborate implementation of the same general strategy, one can show that any weakly sparse subclass of segment graphs have bounded twin-width [**16**]. Theorem 8.1 is also an important pillar in the development of the theory. All the results that come next use, at some point, Theorem 8.1. Besides, it mirrors the fact that monadic second-order transductions of classes of bounded clique-width have bounded clique-width [**35**].

2. Permutations strike back

As detailed in Chapter 1, twin-width originates from some beautiful work on permutations [**51**]. In this section we outline the proof of the following characterization, linking twin-width back to permutations.

Theorem 8.2 ([**29**])**.** *A class of binary structures has bounded twin-width if and only if it is a first-order transduction of a proper permutation class.*

SKETCH. We know that proper permutation classes have bounded twin-width. So Theorem 8.1 implies the *if* statement. Our task is then

to build, from a class $\mathcal{C}$ of bounded twin-width, a permutation class excluding a pattern that transduces $\mathcal{C}$.

We work with twin-models, which are twin-decompositions $(T, <, B)$ deprived of the linear order $<$ on the internal nodes of T. We distinguish three encodings of twin-models on $V(T)$:

- the plain *twin-model* defined by the *tree* edges $E_T := E(T)$, and the *biclique* edges $E_B := B$;
- the *full twin-model* defined by the tree order $\prec_T$ of T, and E_B;
- the *ordered twin-model* defined by the pre-order transversal $<_T$ of T, E_T, and E_B;

Let us recall that the *pre-order* tree transversal visits the current node, recursively explores the left subtree, and recursively explores the right subtree. The full twin-model, unlike the mere twin-model, is expressive enough to reinterpret the original graph G encoded by (T, B). Indeed $uv \in E(G)$ if and only if $\exists u' \exists v'\ u' \preceq_T u \ \wedge\ v' \preceq_T v \ \wedge\ E_B(u', v')$, since a biclique edge $u'v'$ expresses that the sets of leaves of the two subtrees rooted at u' and v' are fully adjacent in G.

This motivates full twin-models. The justification for introducing ordered twin-models is that they contain a linear order. This brings us closer to permutations as those are nothing but two linear orders on a same set. It is a nice (and in the present case, useful) exercise to check that $\prec_T$ can transduce $E_T, <_T$, and vice versa. Each variant of twin-model, as a binary structure itself, has a well-defined twin-width. Figure 8.2 summarizes the current situation.

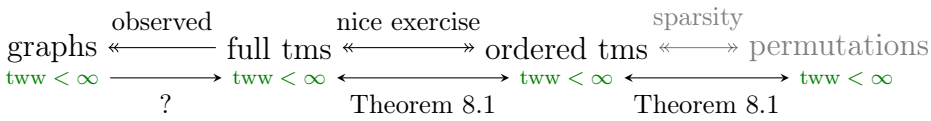


FIGURE 8.2. Proof outline. Single-headed arrows are logical implications, and double-headed arrows are transductions (*tms* stands for *twin-models*).

Let us only say a few words on the two-way transductions between ordered twin-models and permutations. The twin-models having both bounded twin-width and bounded degeneracy, they have, by Theorem 5.10, bounded expansion. One thus benefits from the toolkit of sparsity theory [**71**], and in particular, twin-models can be properly colored with a constant number of colors such that any two color classes induce a disjoint union of stars. This coloring is used to define two

linear orders on a same set (hence a permutation), both built around the pre-order traversal $<_T$, such that the permutation can be uniquely decoded. By Theorem 8.1, the constructed permutations have bounded twin-width, and thus their pattern closure is a proper permutation class.

We would be done, except we never justified why the different twin-models have bounded twin-width. This is the question mark of Figure 8.2. As there is no transduction from graphs to their full twin-models, we cannot simply invoke Theorem 8.1. One possible argument is that the in-order traversal witnesses low mixed number. □

Looking at the full proof in **[29]**, it appears that n-vertex graphs of $\mathcal{C}$ are images by a transduction T of permutations on $O(n)$ elements of a proper permutation class $\mathcal{P}$. Thus $\mathcal{C}$ has, up to isomorphism, at most $(2^{c_\mathsf{T}})^{O(n)} c_\mathcal{P}^{O(n)} = 2^{O(n)}$ graphs on n vertices, where c_T is the number of unary relations of T, and $c_\mathcal{P}$, an upper bound on the basis of the single-exponential growth of $\mathcal{P}$ guaranteed by the Marcus–Tardos–Klazar theorem **[63, 69]**. In the language of the next chapter, classes of bounded twin-width are tiny.

In Theorem 8.2 the transduction *and* the permutation class depend on the graph class of bounded twin-width. We can eliminate the dependence in the permutation class. There is a single permutation class that *captures* all classes of bounded twin-width in the following sense.

Theorem 8.3 ([15]). *There is a permutation class $\mathcal{P}$ such that for every class $\mathcal{C}$ of binary structures, $\mathcal{C}$ has bounded twin-width if and only if $\mathcal{C}$ is a first-order transduction of $\mathcal{P}$.*

The class $\mathcal{P}$ can be chosen as $\mathrm{Av}(\sigma)$ for some permutation σ, which could be made explicit. However, the current proof does not yield a particularly insightful permutation σ, unlike the similar characterizations for clique-width and linear clique-width.

3. Delineation

We say that a class $\mathcal{C}$ of binary structures is *(effectively) delineated* if for every hereditary closure $\mathcal{D}$ of a subclass of $\mathcal{C}$ it holds that $\mathcal{D}$ has (effectively) bounded twin-width if and only if $\mathcal{D}$ is monadically dependent. Note that, as every class of bounded twin-width is monadically dependent, we can equivalently drop "*and only if*" in the previous definition. On the one hand, first-order model checking is fixed-parameter tractable on classes of effectively bounded twin-width **[26]**. On the

other hand, this problem is intractable on any monadically independent class [**41**]. Hence the question of efficient first-order model checking is completely understood in every effectively delineated class.

Every class of (effectively) bounded twin-width trivially is (effectively) delineated, but improperly so. We are interested here in identifying effectively delineated classes of unbounded twin-width.

Theorem 8.4. *These classes are effectively delineated:*

- *permutation graphs* [**26**]*, and more generally,*
- *intersection graphs of chords of a circle* [**58**]*,*
- *interval graphs* [**16, 58**]*, and more generally,*
- *intersection graphs of vertical paths in a tree* [**16**]*, and*
- *ordered graphs* [**23**]*.*

In the previous statement, a *vertical path* in a rooted tree is any subpath of a root-to-leaf path. The last item will be central to Chapter 10. It suggests another characterization of classes of bounded twin-width.

Theorem 8.5 ([**23**])**.** *A graph class has bounded twin-width if and only if it is the reduct of a monadically dependent class of ordered graphs.*

To show that a class is effectively delineated, the main strategy is to come up with an ordering process of the domain. We want that the order either witnesses that the binary structure has low mixed number, hence low twin-width, or reveals an obstruction to monadic dependence. Let us see this in action on interval graphs.

Proposition 8.2 ([**16, 58**])**.** *Interval graphs are effectively delineated.*

SKETCH. Let $\mathcal{D}$ be any hereditary subclass of interval graphs. We order the vertices of graphs in $\mathcal{D}$ by increasing left endpoints, which we can assume all distinct, in a fixed geometric representation. Let us denote by $\prec$ this order. If the resulting adjacency matrices have bounded mixed number, we conclude by Theorem 4.2 that $\mathcal{D}$ has bounded twin-width. We thus assume that we find increasingly large mixed minors among these adjacency matrices.

For every integer n there is an adjacency matrix $M := A_{\prec}(G)$ with $G \in \mathcal{D}$ that has a $3n$-mixed minor, which we can suppose to be a symmetric division, i.e., of the form $(\mathcal{P}, \mathcal{P})$ for a division $\mathcal{P}$ of $(V(G), \prec)$. Let A, B, C be the n first, n middle, and n last parts of $\mathcal{P}$, respectively. We claim that $\mathcal{D}$ transduces the class of all permutations (encoded by two linear orders), known to be monadically independent.

The n-mixed minor of $M[\bigcup B, \bigcup C]$ yields n segments of $\bigcup C$ setting any linear order on n segments of $\bigcup B$, reading $b_1, \dots, b_n$ by increasing left endpoints; see right of Figure 4.2. The segments of $\bigcup A$, whose adjacencies with $\bigcup B$ only depend on the left endpoints of segments in $\bigcup B$, can then set the linear order $b_1 \prec' \dots \prec' b_n$. □

We now turn to classes that are not delineated. Of course, any monadically dependent class containing all cubic graphs is not delineated, as the equivalence between *bounded twin-width* and *monadic dependence* fails on the class itself. The following facts contrast with Theorem 8.4.

Theorem 8.6 ([16]). *These classes are not delineated:*

- *axis-parallel segment graphs,*
- *intersection graphs of paths in a tree,*
- *visibility graphs of simple polygons.*

A *visibility graph* of points in a geometric configuration puts an edge between pairs of points whenever they are visible from each other. Here we consider the most natural visibility notion: two points see each other if the line segment they define does not cross the rest of the geometric configuration. A *simple polygon* is a polygon that is not self-crossing and has no hole. Visibility graphs of simple polygons have as vertex set the geometric vertices of a simple polygon, and as edges every pair of vertices seeing each other.

The way we show that the classes $\mathcal{C}$ of Theorem 8.6 are not delineated is by exhibiting a subclass $\mathcal{D} \subseteq \mathcal{C}$ *transduction equivalent* to the class $\mathcal{B}_{\leqslant 3}$ of all bipartite subcubic graphs, i.e., such that (i) $\mathcal{D}$ transduces $\mathcal{B}_{\leqslant 3}$, and (ii) $\mathcal{B}_{\leqslant 3}$ transduces $\mathcal{D}$. Indeed, item (i) implies by Theorem 8.1 that $\mathcal{D}$ has unbounded twin-width since $\mathcal{B}_{\leqslant 3}$ is not small and so has unbounded twin-width, while item (ii) implies that $\mathcal{D}$ is monadically dependent, as a transduction of a monadically dependent class.

We construct an appropriate subclass $\mathcal{D}$ to show, as an illustrative example, the third item of Theorem 8.6. Subclass $\mathcal{D}$ consists of the image by the following transformation Π of each bipartite subcubic graph G, with bipartition $(A := \{a_1, \dots, a_s\}, B := \{b_1, \dots, b_t\})$. For each vertex $a_i \in A$, we add the $2d_G(a_i) + 1 \in \{1, 3, 5, 7\}$ first vertices of $d_i, p_i, d'_i, p'_i, d''_i, p''_i, d'''_i$. For each vertex $b_j \in B$, we add a vertex q_j. Let D be the set of vertices of the form d_i, d'_i, d''_i, d'''_i, let P be the set of vertices of the form p_i, p'_i, p''_i, and $Q := \{q_j \ : \ b_j \in B\}$.

We make p_i (p'_i, p''_i, respectively) adjacent to q_j such that b_j is the neighbor of a_i with largest (second largest, third largest, respectively)

index. We make d_i adjacent to p_i, d'_i adjacent to p_i and p'_i, d''_i adjacent to p'_i and p''_i, and d'''_i adjacent to p''_i. Finally we turn $D \cup Q$ into a clique. This ends the construction of $\Pi(G)$. Figure 8.3 shows that, for every $G \in \mathcal{B}_{\leqslant 3}$, $\Pi(G)$ is the visibility graph of a simple polygon.

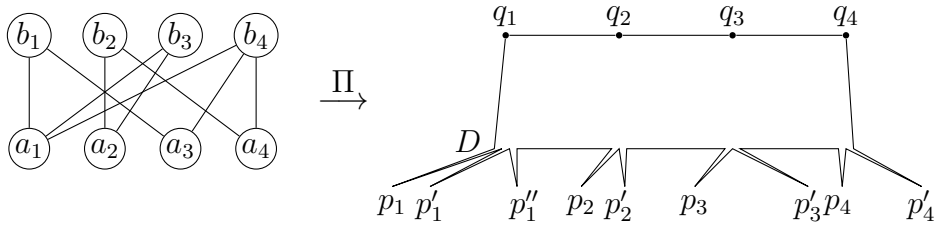


FIGURE 8.3. Representation as visibility graph of a simple polygon for Π applied to the graph on the left.

Proposition 8.3 (**[16]**). *$\mathcal{D}$ and $\mathcal{B}_{\leqslant 3}$ are transduction equivalent.*

SKETCH. To get a transduction from $\mathcal{B}_{\leqslant 3}$ to $\mathcal{D}$, simply observe that $\Pi(G)$ deprived of the edges of the clique $D \cup Q$ is itself a bipartite subcubic graph with bipartition $(P, D \cup Q)$. These edges can be added with a single unary relation interpreted as $D \cup Q$. For the converse transduction, consider adding three unary relations, interpreted as in Figure 8.4.

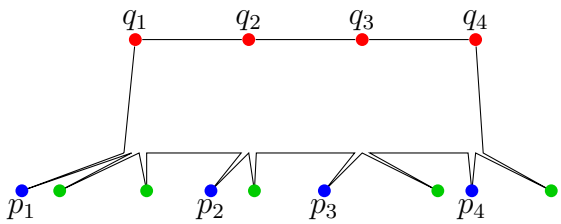


FIGURE 8.4. The desired interpretation of the three unary relations.

Now we recover the initial bipartite subcubic graph, by redefining the domain as the set of blue or red vertices, and edges as linking a blue vertex x to a red vertex y, whenever they are already adjacent, or there is a green vertex w adjacent to y, such that there is a path between x and w on three or five vertices that goes blue–uncolored–green(–uncolored–green). □

CHAPTER 9

Growth of Classes and Labeling Schemes

How many graphs are there in my class $\mathcal{C}$? This is a natural inquiry, if not for the fact that the answer will almost always be unrevealing: *an infinity*. Instead, what we want to ask is how the number of graphs of $\mathcal{C}$ on n vertices scales up with n. This is still not formal enough. Indeed, we defined a graph class as a collection of graphs closed under isomorphism. Thus every single graph $G \in \mathcal{C}$ entails that an infinity of graphs (on various vertex sets) isomorphic to G are also in $\mathcal{C}$. There are two ways to fix the question: counting up to isomorphism or imposing a canonical vertex set. So we either ask for the number of isomorphism equivalence classes among n-vertex graphs in $\mathcal{C}$, that is, the number of n-vertex *unlabeled graphs*—which have indistinguishable vertices. Or we count up to equality the number of n-vertex *labeled graphs* in $\mathcal{C}$, whose vertex set is imposed to be $[n]$; see Figure 9.1.

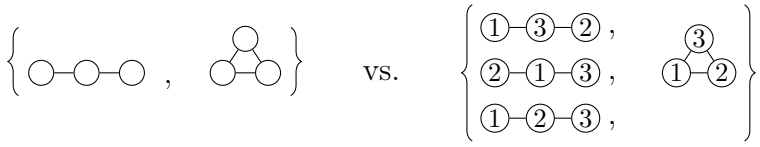


FIGURE 9.1. A simple class with two unlabeled graphs, and four labeled graphs. The 3-vertex path has 2 automorphisms, so gives rise to $3!/2 = 3$ labeled graphs.

A class $\mathcal{C}$ has *growth* (resp. *unlabeled growth*) $f\colon \mathbb{N} \to \mathbb{N}$ if for every natural number n, class $\mathcal{C}$ has at most $f(n)$ labeled (resp. unlabeled) n-vertex members. We may denote by $\mathcal{C}_n$ the set of the n-vertex labeled members of $\mathcal{C}$. Note that if a class has unlabeled growth $f(n)$, then it has growth $n!\, f(n)$. Most natural hereditary graph classes fall into one of the following three categories:

- growth $n!\, 2^{O(n)}$ such as planar graphs, and more generally K_t-minor-free graphs, or graphs with bounded treewidth or clique-width;
- growth $2^{\Theta(n \log n)}$ such as interval graphs, graphs with bounded degeneracy, and more generally with bounded symmetric difference;

- growth $2^{\Theta(n^2)}$ such as bipartite graphs, or the class of all graphs.

There is a formal categorization of the growth of hereditary classes: Every hereditary graph class has constant, polynomial, single-exponential, or at least factorial growth **[79]**. While these "jumps" are noteworthy, they do not reflect very relevant dividing lines. Indeed note that the growth of the very simple class of all paths is $n!/2$, hence paths, as most interesting graph classes, fall in the last category.

On the other hand, most classes with growth $2^{\Theta(n^2)}$, or simply growth $2^{\Omega(n^\beta)}$ and $\beta > 1$, are too complex to lend themselves to exploitable structural decompositions. We thus focus on *factorial* classes, i.e., hereditary classes with growth $2^{O(n \log n)}$. We ask for connections between the growth of a class and its structural or algorithmic simplicity. Among factorial classes, we are particularly interested in the so-called small classes.

1. Small and tiny classes

A hereditary graph class $\mathcal{C}$ is *small* if there is a number c such that $\mathcal{C}$ has growth $n!\, c^n$, and *tiny* if $\mathcal{C}$ satisfies the stronger condition that its unlabeled growth is at most c^n. We unify and generalize the smallness shown on individual classes of bounded twin-width **[8, 63, 69, 72, 61]**.

Theorem 9.1 (**[22]**). *Every class of bounded twin-width is small.*

Proof. For the sake of simplicity, we give the argument in the case of a *graph* class $\mathcal{C}$. By Theorem 4.5, $\mathcal{C}$ has versatile twin-width at most $d < \infty$. We show the stronger fact that the class $\mathcal{V}^d$ of all trigraphs with versatile twin-width at most d is small. Let $\mathcal{L}^d_n \subseteq \mathcal{V}^d_n$ consist of the labeled trigraphs of $\mathcal{V}^d_n$ wherein vertex n can be contracted with another vertex such that the resulting labeled trigraph is in $\mathcal{V}^d_{n-1}$.

By definition of versatile twin-width, $|\mathcal{L}^d_n| \geqslant |\mathcal{V}^d_n|/d$. Indeed, in at least n/d circular rotations of the vertex set of a member of $\mathcal{V}^d_n$, the last vertex is in a pair that can be contracted. We can now upper bound $|\mathcal{L}^d_n|$ in terms of $|\mathcal{V}^d_{n-1}|$. Indeed from any labeled trigraph H of $\mathcal{V}^d_{n-1}$ can only come at most $(n-1) \cdot 3 \cdot 7^d$ labeled trigraphs of $\mathcal{L}^d_n$, where $n-1$ upper bounds the number of vertices $j \in V(H)$ that can be split into j, n, the number 3 accounts for the existence of an edge, red edge, or non-edge between j and n, and $7^d = (3^2-2)^d$ upper bounds the possible relations between j, n and the at most d red neighbors of j

in H. Therefore

$$|\mathcal{V}_n^d| \leqslant d \cdot |\mathcal{L}_n^d| \leqslant d \cdot (n-1) \cdot 3 \cdot 7^d \cdot |\mathcal{V}_{n-1}^d|,$$

and we conclude by induction that $|\mathcal{V}_n^d| \leqslant n! \cdot c^n$ with $c := 3d \cdot 7^d$. □

The previous proof was a pretext to play with versatile twin-width, and revisit the scheme of [**72**]. We already know a stronger result. Indeed, as we have observed in Chapter 8, the linear transduction of Theorem 8.2 transfers the tinyness of proper permutation classes [**63, 69**] to any class of bounded twin-width.

Theorem 9.2 ([**29**])**.** *Every class of bounded twin-width is tiny.*

This prompts two questions:

- Is every hereditary small class tiny?
- Is every hereditary tiny class of bounded twin-width?

Without the *hereditary* condition, these questions have easy negative answers. While the first question remains open, Section 3 answers the second one negatively. Not only *bounded twin-width* fails to characterize smallness within hereditary classes, but so does any countable family of graph parameters.

2. Labeling schemes and universal graphs

The unlabeled growth of a class $\mathcal{C}$ gives an information-theoretic lower bound on how compactly its n-vertex members can be represented, up to isomorphism. One indeed needs at least $\log |\mathcal{C}_n^u|$ bits to do so, where $\mathcal{C}_n^u$ denotes the set of n-vertex unlabeled graphs of $\mathcal{C}$. As the unlabeled growth of classes of bounded twin-width is single-exponential, an $O(n)$-bit representation is theoretically possible. The interval biclique partition already guarantees a representation with at most $4(d+1)n \log n$ bits, for n-vertex unlabeled graphs of twin-width at most d. Some ingenious recursive data structure on matrices with low mixed number indeed meets the information-theoretic bound of $O_d(n)$ [**76**], albeit with a worst dependence in d, but improved $O_d(\log \log n)$ time per adjacency query.

A related task, primarily motivated by network representation in a distributed setting, is that of finding short adjacency labeling schemes. A graph class $\mathcal{C}$ has an *$f(n)$-bit adjacency labeling scheme*, or labeling scheme for short, if there is a decoding function $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ such that for every n-vertex graph $G \in \mathcal{C}$ there is an injective labeling function $\ell : V(G) \to \{0,1\}^*$, satisfying $|\ell(u)| \leqslant f(n)$ for every

$u \in V(G)$, and $A(\ell(u), \ell(v)) = 1$ if and only if $uv \in E(G)$. Intuitively, we want to assign to each vertex a bit string as short as possible so that the adjacency between u and v can be determined solely based on their labels. For example, trees have a $\log n + O(1)$-bit adjacency labeling scheme **[2]**.

Class $\mathcal{C}$ has an $f(n)$-bit adjacency labeling scheme if and only if, for every integer n, there is a *universal graph* U_n, not necessarily in $\mathcal{C}$, on at most $2^{f(n)}$ vertices such that every n-vertex graph of $\mathcal{C}$ is an induced subgraph of U_n; to see this, consider the possible labels as the vertex set of the universal graph. Numerous classes, such as interval graphs and K_t-minor-free graphs, have $O(\log n)$-bit labeling schemes. By counting, only factorial classes may possibly admit $O(\log n)$-bit labeling schemes. Indeed the number κ of distinct labels is $2^{O(\log n)} = n^{O(1)}$. Hence the number of n-vertex graphs that can be induced subgraphs of the universal graph U_n is upper bounded by $\binom{\kappa}{n} = n^{O(n)}$. The *implicit graph conjecture* originally simply asks whether every factorial hereditary class has an $O(\log n)$-bit labeling scheme **[62]**. It was recently refuted **[56]** but holds in classes of bounded twin-width.

Theorem 9.3 (**[22]**). *For every integer d, the class of graphs with twin-width at most d admits a $2^{2^{O(d)}} \log n$-bit labeling scheme.*

And equivalently, every class of bounded twin-width has universal graphs of polynomial size. We give a simple $O_d(\log^2 n)$-bit labeling scheme directly based on twin-decompositions $(T, <, B)$ where T has depth $O_d(\log n)$, whose existence is implied by Theorem 4.5.

The graph $(V(T), B)$ has degeneracy at most $f(d) := d'+1$, where d' is the versatile twin-width of the class of graphs of twin-width at most d. Thus there is an orientation of $(V(T), B)$ with maximum outdegree at most $f(d)$, i.e., such that every vertex has at most $f(d)$ *out-neighbors*. We first give an *identifier* to every vertex in $V(T)$, in the form of a nonzero bit string of length $\lceil \log 2n \rceil$. The label $\ell(u)$ of a leaf u of T is the concatenation of the identifiers of u and its ancestors (first half), followed by the identifiers of the out-neighbors of u and its ancestors (second half). We can pad the first half of the label with zeros, so that each has exactly length $\lceil \log 2n \rceil \cdot h$ with $h = O_d(\log n)$ being the depth of T. This way, one can distinguish the first half of the label from its second half. As every identifier has length exactly $\lceil \log 2n \rceil$, breaking $\ell(u)$ down to its identifiers is also easy. The decoding A simply outputs 1 whenever an identifier in the first half of one label is equal to an identifier in the second half of the other label.

This produces labels of size at most $\lceil \log 2n \rceil \cdot h \cdot (f(d)+1) = O_d(\log^2 n)$. The correctness of the decoding relies on a property of twin-decompositions that we already observed, for instance, in the proof of Theorem 8.2. The proof of Theorem 9.3 also uses twin-decompositions of logarithmic depth, but is less wasteful in the label size by describing what locally happens when a contraction occurs.

It is an open question to design *effective* labeling schemes with (poly)logarithmic label size, where by *effective* we require that the labeling function can be computed in polynomial-time. An orthogonal quest is to reduce the *label* length. As far as we currently know, the information-theoretic lower bound of $(1+o_d(1))\log n$-bit labeling scheme could be reachable.

3. Complex tiny classes

We are going to see that small hereditary classes can have unbounded twin-width, and actually that, for every number c, there are tiny monotone classes with both unbounded twin-width and no $c\log n$-bit labeling scheme—far from the information-theoretic lower bound of $(1+o(1))\log n$. This testifies that even in this growth regime, classes can be particularly untamed. By analyzing the behavior of the Erdős–Rényi random graphs $G(n, \frac{d}{n})$, it can be shown **[20]** that a polynomial fraction of n-vertex graphs of average degree at most d satisfy the following properties for some γ depending only on d:

(1) each of their k-vertex connected subgraphs with $k \leqslant \log^2 n$ has at most $k(1+\frac{1}{\log k})$ edges, and for every $k' \in [n]$
(2) they have at most $\gamma^{k'}$ subgraphs on k' vertices up to isomorphism.

Let us denote by $\mathcal{D}(d)$ the class of graphs of average degree at most d satisfying the above properties. By the previous claim shown in **[20]**,

$$|\mathcal{D}(d)^u_n| \geqslant 2^{(1-o(1))(\frac{d}{2}-1)n\log n}.$$

The class $\mathcal{Y}$ of all connected n-vertex graphs with at most $n(1+\frac{1}{\log n})$ edges, for n ranging over the natural numbers, is tiny. Indeed any such graph can be described by a spanning tree and fewer than $n/\log n$ extra edges. And, as the number of unlabeled n-vertex trees is at most $n\cdot 4^n$ **[74]**:

$$|\mathcal{Y}^u_n| \leqslant n\cdot 4^n \cdot \sum_{0\leqslant i\leqslant \frac{n}{\log n}} \binom{\binom{n}{2}}{i} \leqslant \frac{n^2}{\log n}\cdot 4^n \cdot n^{\frac{2n}{\log n}} = \frac{n^2}{\log n}\cdot 16^n.$$

As the *partition function*[1] is subexponential (it is upper bounded by $\exp(\pi\sqrt{2n/3})$ [**44**]), the class $\mathcal{Z}$ of all disjoint unions of graphs of $\mathcal{Y}$ is also tiny, with $|\mathcal{Z}^u_n| \leqslant 50^n$ for every n.

Let us define the sequence $\ell_0 := 1$ and $\ell_{i+1} := \lceil 2^{\sqrt{\ell_i}} \rceil$ for every $i \in \mathbb{N}$. We now have all the ingredients to build a complex tiny monotone class. For each ℓ_i, we can select up to $2^{O(\log^2 \ell_i)}$ many ℓ_i-vertex graphs of $\mathcal{D}(d)$, and take the subgraph closure of our collection of graphs. Figure 9.2 illustrates why such a class is indeed tiny.

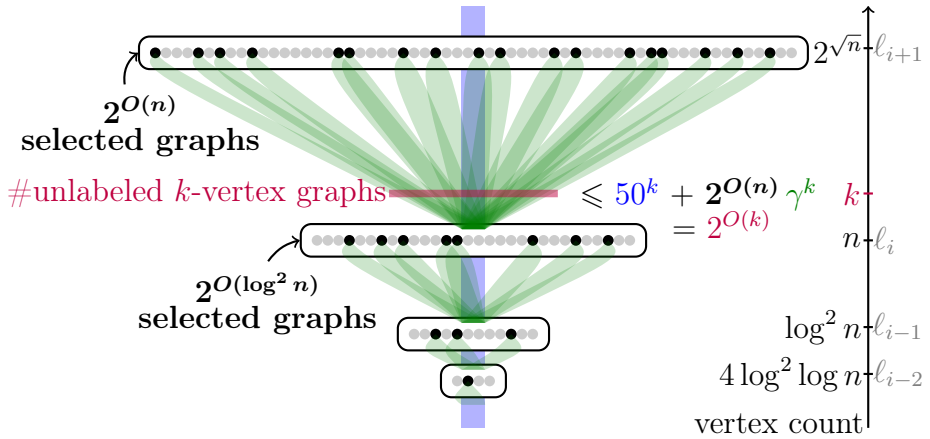


FIGURE 9.2. Why the built class is tiny. The blue strip represents the class $\mathcal{Z}$. The gray dots are the graphs of $\mathcal{D}(d)$, and the black dots, those selected. The green "cones" represent their subgraphs.

Indeed, on k vertices with $n := \ell_i < k \leqslant \ell_{i+1}$, we find

- subgraphs of selected graphs on ℓ_j vertices with $j > i+1$, but these are in $\mathcal{Z}$ by Property (1), so up to isomorphism fewer than 50^k,
- subgraphs of the at most $2^{O(n)}$, hence $2^{O(k)}$, selected graphs on ℓ_{i+1} vertices, each of which has at most γ^k non-isomorphic subgraphs on k vertices by Property (2).

As $\mathcal{D}(d)$ is not a small class for any $d \geqslant 5$, it has unbounded twin-width, and picking a single ℓ_i-vertex graph $H_i \in \mathcal{D}(d)$ with $\lim_{i\to\infty} \mathrm{tww}(H_i) = \infty$, builds a tiny monotone class of unbounded twin-width. We could decide not to select any graph on ℓ_i vertices, for some i. We only need to pick an infinite number of graphs H_i. Let us say that a graph parameter p is *small* (resp. *tiny*) if every class

[1]This is perhaps a misnomer for a function that counts the number of multisets of positive integers summing up to n, and is *not* the sequence of Bell numbers.

with bounded p is small (resp. tiny). With the previous observation, and since $\mathbb{N}^2$ and $\mathbb{N}$ are equipotent, one can construct a tiny monotone class simultaneously with unbounded p for a countable number of small parameters p.

We can use a "second-order" counting argument devised to refute the implicit graph conjecture [**56**], in order to show, for every c, the existence of tiny monotone classes without $c \log n$-bit labeling scheme or equivalently n^c-vertex universal graph. A "first-order" counting argument would simply be that an n^c-vertex graph has at most $\binom{n^c}{n}$ induced subgraphs on n vertices. The "second-order" counting argument upper bounds the number of *collections of κ_n graphs on n vertices* that can simultaneously be induced subgraphs of a graph on $u_n := n^c$ vertices by

$$2^{\binom{u_n}{2}} \binom{\binom{u_n}{n}}{\kappa_n} \leqslant 2^{n^{2c} + c\kappa_n n \log n}.$$

If we set $\kappa_n := n^{2c-1}$, this upper bound becomes $2^{(1+c\log n)n^{2c}}$. On the other hand, the number of families of κ_n graphs of $\mathcal{D}(d)_n^u$ is at least

$$\binom{2^{(1-o(1))(\frac{d}{2}-1)n\log n}}{\kappa_n} > 2^{\frac{d}{3}\kappa_n n \log n} = 2^{\frac{d}{3}\log n \cdot n^{2c}},$$

with the first inequality holding for sufficiently large n. If we initially chose $d := 3c + 4$, then $2^{\frac{d}{3}\log n \cdot n^{2c}} > 2^{(1+c\log n)n^{2c}}$. Thus at least one choice (in fact most choices) to select the $\kappa_n = 2^{O(\log^2 n)}$ graphs rules out a universal graph of size n^c. We deduce:

Theorem 9.4 ([**20**]). *For every family $(p_i)_{i\in\mathbb{N}}$ of small parameters and every number c, there is a tiny monotone graph class that has unbounded p_i for every $i \in \mathbb{N}$, and no $c \log n$-bit labeling scheme.*

The *girth* of a graph is the length of its shortest cycles. Another way to design small classes of unbounded twin-width uses a random group construction [**73**], which isometrically embeds any infinite family of bounded-degree graphs with rapidly increasing girth and bounded *diameter* to *girth* ratio, within a single Cayley graph of a finitely generated group. Interestingly both this method and the one we presented rely on a counting argument. As in Section 4.2 of Chapter 5, so far no small or tiny hereditary class of unbounded twin-width has been *explicitly* built.

CHAPTER 10

Ordered Graphs and Matrices

An *ordered graph* is a relational structure made of two binary relations: one interpreted as a simple graph, and the other, as a linear order on its vertices. This final chapter is devoted to characterizing *bounded twin-width* within ordered graphs or matrices over finite alphabets whose rows and columns carry their natural ordering. Looking back at the previous chapters, a list of shortcomings of twin-width—or of *us*, for the first item—could read as follows:

- We do not know how to efficiently approximate $O(1)$-sequences;
- Twin-width does not characterize monadic dependence, nor classes with a fixed-parameter tractable first-order model checking;
- Twin-width comes close but fails to characterize small classes.

All these complaints dissipate within ordered graphs. A possible conclusion would be that twin-width really is about linearly ordered binary structures. Nevertheless, this should be nuanced as we also saw some interesting applications of twin-width on unordered graphs, some for which an extra linear order would not be particularly meaningful.

1. Rank minors, rank number, and rich divisions

Now that our graph G comes equipped with a linear order $\prec$, it would be tempting to think that $A_{\prec}(G, \prec)$ has low mixed number if and only if $(G, \prec)$ has low twin-width. This is not quite true. Consider the ordered graph $(G, \prec)$ with $V(G) = [n]$ such that $\prec$ is the natural order on $[n]$, and $ij \in E(G)$ whenever $i + j$ is odd. The matrix $A_{\prec}(G, \prec)$ has a $\lfloor n/2 \rfloor$-mixed minor (see left of Figure 10.1), but the linear order $\prec'$ such that $i \prec' j$ if i is odd and j is even, or if i and j have the same parity and $i < j$, verifies that $A_{\prec'}(G, \prec)$ is 4-mixed free (see right of Figure 10.1).

A t-mixed minor, we recall, is a t-division every cell of which has at least two distinct rows and at least two distinct columns. Let us instead require that every cell has at least t distinct rows and at least t distinct columns. We call this new t-division a *t-rank minor*. We can now define the *rank number* of a matrix as the largest t such that it

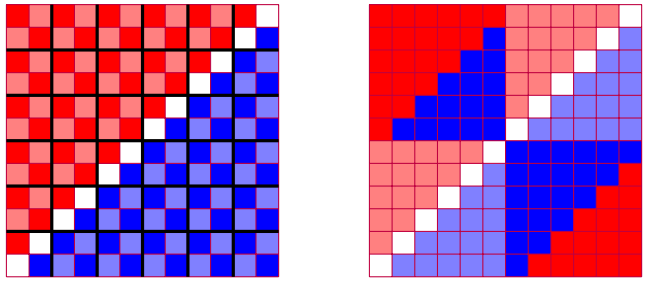


FIGURE 10.1. $A_{\prec}(G, \prec)$ (left) has a large mixed minor, while $A_{\prec'}(G, \prec)$ (right) is 4-mixed free. In the color coding of entry (i, j), *red* means $i > j$, *blue* means $i < j$, and edges of G make the color darker.

admits a t-rank minor, and the *rank number* of a binary structure as the minimum rank number among its adjacency matrices. Promisingly, $A_{\prec}(G, \prec)$ has no 3-rank minor. This new definition, as we shall see, spares us the task of finding the "good" ordering $\prec'$. Given an *ordered binary structure* $(G, \prec)$, i.e., a binary structure G and a linear order $\prec$ on the same domain, we call *canonical adjacency matrix* of $(G, \prec)$ the matrix $A_{\prec}(G, \prec)$.

Theorem 10.1 ([**23**]). *A class of ordered binary structures has bounded twin-width if and only if its canonical adjacency matrices have bounded rank number.*

Theorem 10.1 is established by introducing yet another kind of matrix divisions. A *t-rich division* of a matrix M is a division $(\mathcal{R}, \mathcal{C})$ of M such that

- for every $R \in \mathcal{R}$ and $X \in \binom{\mathcal{C}}{t}$, the submatrix $M[R, \mathcal{C} \setminus \bigcup X]$ has at least t distinct row vectors, and symmetrically
- for every $C \in \mathcal{C}$ and $X \in \binom{\mathcal{R}}{t}$, the submatrix $M[\mathcal{R} \setminus \bigcup X, C]$ has at least t distinct column vectors.

Intuitively, the diversity of vectors in every part should be robust under any removal of a few "orthogonal" parts. Observe that a $t + 1$-rank minor is a t-rich division. Indeed after deleting, say, any t row parts of the rank minor, every column part still forms with the non-deleted row part a submatrix with at least $t + 1$ distinct columns. Conversely, the existence of huge rich divisions implies that of large rank minors.

Lemma 10.1 ([**23**]). *There is a function f, such that for every integer t and finite alphabet A, every matrix M over A with an $f(t, |A|)$-rich division admits a t-rank minor.*

SKETCH. Let $(\mathcal{R}, \mathcal{C})$ be an $f(t, |A|)$-rich division of M, for some large function f. For every $R \in \mathcal{R}$ (and symmetrically *in columns*),

- the number of cells $M[R, C]$ with $C \in \mathcal{C}$ that are the first, from left to right, to contain a particular *column* vector among the cells with fewer than t distinct column vectors
- plus the number of cells $M[R, C]$ with $C \in \mathcal{C}$ having at least t distinct column vectors and at least t distinct row vectors

has to be large. Otherwise R can be shown to contradict that $(\mathcal{R}, \mathcal{C})$ is a rich division. We then find a t-rank minor by applying the Marcus–Tardos theorem (see Theorem 4.1) to the auxiliary matrix with a 1-entry for each cell fitting the above description. □

Therefore Theorem 10.1 may be obtained by relating twin-width and rich divisions. Let us see a first direction of this functional equivalence.

Lemma 10.2. *Let $(G, \prec)$ be an ordered binary structure of twin-width at most d. Then $A_{\prec}(G, \prec)$ has no $2(d+2)^2$-rich division.*

PROOF. Let $\mathcal{D}$ be any division of $M := A_{\prec}(G, \prec)$, and let us show that $\mathcal{D}$ is not a $2(d+2)^2$-rich division. Consider the first partition $\mathcal{P}$ of a d-sequence of $(G, \prec)$ such that a part $P \in \mathcal{P}$ intersects three different parts of $\mathcal{D}$, say the column parts C_1, C_2, C_3; see Figure 10.2.

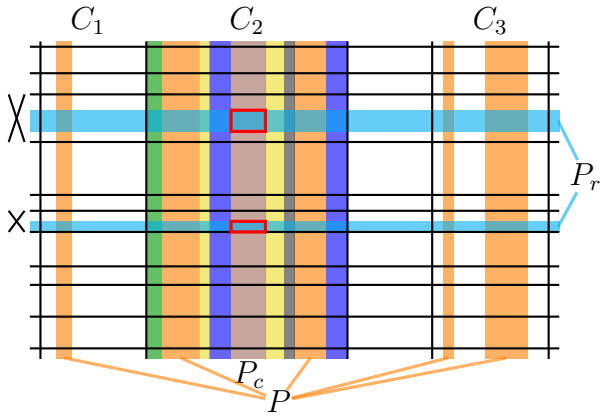


FIGURE 10.2. The division $\mathcal{D}$ in black, the part $P \in \mathcal{P}$ in orange. A part P_c intersecting C_2 and one of its red neighbors P_r make for at most two parts of $\mathcal{D}$ to remove.

There are at most d other parts P' of $\mathcal{P}$ intersecting C_2 since such a part P' is a red neighbor of P. Indeed any $(x, y) \in (P \cap C_1) \times (P' \cap C_2)$ and $(x', y') \in (P \cap C_3) \times (P' \cap C_2)$ have distinct binary atomic types due to $\prec$. For each $P_c \in \mathcal{P}$ (seen as a column part) intersecting C_2, there are at most $d+1$ parts $P_r \in \mathcal{P}$ (seen as a row part) such that $M[P_r, P_c]$ is not constant. Each part P_r intersects, by design, at most two row parts of $\mathcal{D}$. Let us remove these at most $2(d+1)$ row parts of $\mathcal{D}$ for each such P_c; including P, for which the upper bound is $2(d+1)+1$. That is a total of $2(d+1)^2 + 1 < 2(d+2)^2$ removed parts. In the remaining matrix, C_2 has at most $d+1$ distinct columns; at most one per part of $\mathcal{P}$ intersecting C_2. □

The other direction comes in some algorithmic form, important in its own right. There is a fixed-parameter tractable algorithm that, on the canonical adjacency matrix of an ordered binary structure, either finds a rich division witnessing large twin-width, or a contraction sequence of relatively low width.

Theorem 10.2 ([**23, 24**])**.** *There are functions f, g and an algorithm that, given an n-vertex ordered binary structure $(G, \prec)$ over Σ and an integer d, either returns*

- *a $2(d+2)^2$-rich division of $A_{\prec}(G, \prec)$, thus $tww(G, \prec) > d$, or*
- *a $g(d, |\Sigma|)$-sequence of $(G, \prec)$,*

in time $f(d, |\Sigma|)\, n^2 \log n$.

SKETCH. The idea is to first greedily compute a partition sequence such that every part P of every partition $\mathcal{P}$ contradicts that $\mathcal{P}$ is a rich division. If this process is stopped, we can exhibit a $2(d+2)^2$-rich division of $A_{\prec}(G, \prec)$. Otherwise, the partition sequence can be turned into one with low width, similarly to Lemma 4.2. □

Theorem 10.2 is the sort of parameterized approximation algorithm that is still missing for unordered binary structures.

2. Equivalences

Unless the parameterized complexity classes FPT and AW[$*$] coincide, which is very unlikely, there is no fixed-parameter tractable algorithm for first-order model checking on general graphs. This problem is indeed complete for the class AW[$*$]. Our next milestone is the following theorem, which equates *bounded twin-width* with algorithmic, model-theoretic, and enumerative properties.

Theorem 10.3 ([**23**]). *Let $\mathcal{C}$ be a hereditary class of ordered binary structures. The following are equivalent:*

(1) $\mathcal{C}$ has bounded twin-width;
(2) $\mathcal{C}$ is monadically dependent;
(3) $\mathcal{C}$ admits a fixed-parameter tractable first-order model checking;
(this item only implies the other items under FPT $\neq$ AW[$*$].)
(4) $\mathcal{C}$ is tiny.

From the previous chapters, we know that the first item implies the other ones; actually, (1) $\Rightarrow$ (3) also requires Theorem 10.2. We now need to understand ordered binary structures with unbounded twin-width. From Section 1 we learned that the canonical adjacency matrices of these structures have unbounded rank number.

We call *disguised n-grid permutation* any $2n^2 \times 2n^2$ matrix over an alphabet of size 2 obtained from the matrix of the grid permutation on n^2 elements, by turning 1-entries into non-constant 2×2 matrices, and 0-entries into constant 2×2 matrices. Figure 10.3 illustrates a disguised 3-grid permutation.

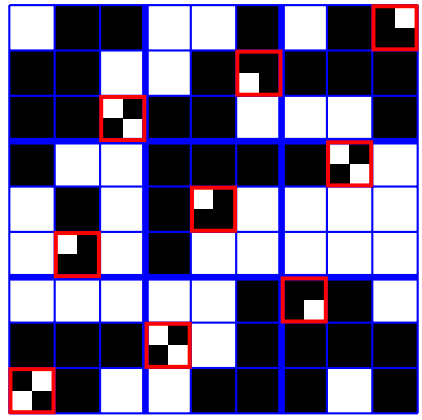


FIGURE 10.3. A disguised 3-grid permutation. The red squares show the 1-entries of the grid permutation.

There is a function f such that every matrix with rank number at least $f(n)$ contains a disguised n-grid permutation as a submatrix. This is shown by iterating over non-constant cells C forming the grid permutation, and restricting the current matrix to a submatrix where C is only aligned with constant cells. This establishes Theorem 10.3 since there is a transparent transduction from any class with arbitrarily large

disguised grid permutations to the class of all (ordered) permutations. The latter class is monadically independent, has an AW[∗]-hard first-order model checking, and is factorial. Indeed any graph can be encoded as an ordered matching, for example, by allocating intervals along the order $\prec$ for each vertex, with edges of length 1 along $\prec$ to delimit the intervals, and one edge bridging two intervals if they correspond to adjacent vertices. Permutations and ordered matchings can be seen transduction equivalent, the same way bipartite graphs and general graphs are.

We can deepen our understanding of hereditary classes of ordered binary structures over a finite signature with unbounded twin-width. Some further applications of Ramsey's theorem on disguised grid permutations show that any such class includes the class of all permutations in at least one of the six forms of Figure 10.4.

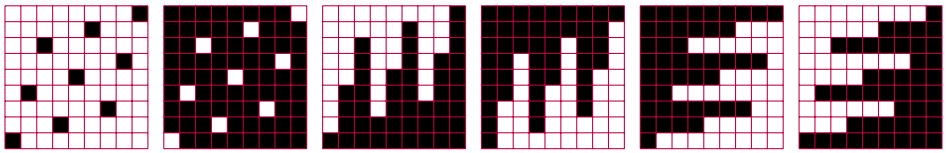


FIGURE 10.4. Grid permutation in each of the six families.

The first family is simply the class of all permutations. The second one flips 0- and 1-entries. The other four propagate 1-entries of a permutation matrix downward, upward, leftward, and rightward, respectively. Note that this description is tailored for 0–1 matrices. In the general case of finite alphabets A, the 0- and 1-entries are realized by two distinct elements of A, and the six families thus collapse to only three.

In any hereditary class of ordered graphs with unbounded twin-width, we either find the class of permutation graphs with their natural order, or arbitrarily large sets X, Y such that $X \prec Y$ and their bi-adjacency matrix is the grid permutation in one of the six forms of Figure 10.4. In the latter case, we can further obtain that X, Y are one the four combinations of cliques and independent sets. This defines *25 families* $(1 + 6 \cdot 4)$ with unbounded twin-width. These families are all minimal: all their proper hereditary subclasses have bounded twin-width. An ordered graph, due to its linear order, has no automorphism other than the identity. Thus the growth and unlabeled growth of any class of ordered graphs are exactly an $n!$ multiplicative

factor apart. One can check that the growth of each of the 25 families is appropriately fast, to confirm a conjecture that the growth of hereditary classes of ordered graphs jumps from single-exponential to factorial [**6**].

Theorem 10.4 ([**23**]). *Every hereditary class of ordered graphs has growth at most single-exponential or at least* $\sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} k!$.

PROOF. Either the class has bounded twin-width and hence is tiny by Theorem 9.2, or it includes one the 25 described families. □

3. Unconditional algorithms

We conclude with some algorithms free of the caveat of Chapter 7.

3.1. Matrices of bounded twin-width. Due to Theorem 10.2, we no longer require contraction sequences to be given in order to efficiently compute on matrices of bounded twin-width. We will exclusively deal with matrices M over a fixed finite alphabet, say $A = \{a_0, \dots, a_{q-1}\}$, and consider them as binary structures whose signature is $\{E_0, \dots, E_{q-1}, \prec, R\}$ and domain is the set of row and column indices, with for every $i \in [0, q-1]$,

- E_i has arity 2, and is interpreted as $E_i(x, y) \Leftrightarrow M[x, y] = a_i$,
- $\prec$ is interpreted as the natural linear order on the row indices followed by the column indices, and
- R is a unary symbol interpreted as $R(x) \Leftrightarrow x$ is a row index.

We will rely on an augmented logic. First-order logic with *modulo counting*, denoted by *FO+MOD*, introduces a new type of quantifiers: $\exists^{i[p]}$ for some fixed integer p, and for any $i \in [0, p-1]$. For any structure $\mathcal{M}$ with domain D,

$$\mathcal{M} \models \exists^{i[p]} x\, \varphi(x) \text{ holds whenever } |\{a \in D \,:\, \mathcal{M} \models \varphi(a)\}| \equiv i \mod p.$$

One can now express that the number of witnesses for a first-order formula $\varphi(x)$ is equal to i modulo p. Despite this seemingly increased power, both the model checking algorithm (Theorem 7.4) and the closure of the family of classes of bounded twin-width by transductions (Theorem 8.1) readily generalize to this logic.

As $\left(\begin{smallmatrix} 0 & A \\ B & 0 \end{smallmatrix}\right)^2 = \left(\begin{smallmatrix} AB & 0 \\ 0 & BA \end{smallmatrix}\right)$, multiplying two matrices or simply squaring one is essentially the same. Interestingly, squaring a matrix M over a finite field $\mathbb{F}_q$ can be written as an FO+MOD-interpretation. One

can keep the relations R and $\prec$ as in M, and set for every $i \in [0, q-1]$,

$$E_i^{M^2}(x,y) := \bigvee_{\substack{a:[q-1]^2 \to [0,q-1] \\ \sum_{j,k \in [q-1]^2} a(j,k)\cdot(\tilde{j}\tilde{k}) = \tilde{i}}} \bigwedge_{j,k \in [q-1]} \exists^{a(j,k)[q]} z\, E_j^M(x,z) \wedge E_k^M(z,y).$$

In the previous formula, we write $\tilde{i}$ for the element of $\mathbb{F}_q$ corresponding to relation E_i, and assume that $\tilde{0}$ is the absorbing element. The expression $\tilde{j}\tilde{k}$ is a product in $\mathbb{F}_q$, while $a(j,k) \cdot (\tilde{j}\tilde{k})$ is meant as the sum of $a(j,k)$ occurrences of $\tilde{j}\tilde{k}$ in $\mathbb{F}_q$. As every element of $(\mathbb{F}_q, +)$ has an order dividing q, it is enough to count the number of pairs $(\tilde{j}, \tilde{k}) = (M[x,z], M[z,y])$ modulo q, which the formula does.

Given a collection $\mathcal{F}$ of matrices, let us denote by $\mathcal{F}^2$ the set of all products MN such that M, N are in $\mathcal{F}$, with the right dimension for MN to be defined. By the claimed extension of Theorem 8.1 and the above interpretation, we get the following fact.

Theorem 10.5 ([**24**]). *Let $\mathcal{F}$ be a family of matrices of bounded twin-width over $\mathbb{F}_q$. Then $\mathcal{F}^2$ has bounded twin-width.*

Moreover we can square matrices, hence multiply them, in time $O_d(n^2 \log n)$ within $n \times n$ matrices of twin-width at most d. One way is to use an extension of Theorem 7.4 [**49**]: Given a Σ-structure $\mathcal{M}$ on domain D, a d-sequence of $\mathcal{M}$, and a first-order Σ-formula $\varphi(x_1, \ldots, x_h)$ of quantifier rank k, a data structure can be computed in time $O_{d,h,k}(n)$ that answers whether $\mathcal{M} \models \varphi(v_1, \ldots, v_h)$ holds for any query $v_1, \ldots, v_h \in D$ in time $O_{d,h,k}(\log \log n)$. The same is possible with construction time $O_{d,h,k}(n^{1+\varepsilon})$ for any $\varepsilon > 0$, and query time $O_{d,h,k}(1/\varepsilon)$ [**49**]. Again, this result readily applies to FO+MOD. Combined with Theorem 10.2 and the FO+MOD-interpretation for squaring matrices, we obtain the following theorem.

Theorem 10.6 ([**24**]). *The square of $n \times n$ matrices over $\mathbb{F}_q$ with twin-width at most d can be computed in time $O_d(n^2 \log n)$.*

The current bottleneck is to compute an $O_d(1)$-sequence. Should this step be done in $O_d(n^2)$, we would get an optimal quadratic-time algorithm for multiplying two matrices of bounded twin-width. Alternatively to using the data structure in [**49**], one can convert the contraction sequence into a twin-decomposition in time $O_d(n^2)$. Then an *ad hoc* squaring algorithm can be designed operating at the level of twin-decompositions.

Theorem 10.7 ([24]). *Given a twin-decomposition with width d of an $n \times n$ matrix M over $\mathbb{F}_q$, one can compute a twin-decomposition of M^2 with width $2^{O_q(d)}$ in time $2^{O_q(d)}\, n$.*

The linear time in n, sublinear in the number of matrix entries, renders twin-decompositions a good encoding of matrices of bounded twin-width to perform multiplications. The represented matrix can be recovered in $O(n^2)$ time, and on twin-decompositions of width d and depth h, individual entry queries can be answered in $O(dh)$ time.

Theorem 10.2 also leads to parameterized algorithms on matrices. Interestingly, while only a fixed-parameter tractable *approximation* algorithm is known for the twin-width of matrices, we can *exactly* compute their grid, mixed, and rank numbers.

Theorem 10.8 ([24]). *Computing the grid number, mixed number, or rank number of a matrix is fixed-parameter tractable.*

SKETCH. We show the statement for the grid number as the other numbers can be computed analogously. Let M be the input $n \times m$ matrix that we assume, for simplicity, to be a 0–1 matrix. The argument would work the same on any finite alphabet. By Theorem 10.2, we find a d-sequence for M. By Theorem 4.2, we know that d is upper bounded by a fixed function of $\mathrm{gn}(M)$; hence this first step is fixed-parameter tractable (in $\mathrm{gn}(M)$). We finally express the existence of a t-grid minor in M by a first-order sentence, and conclude with Theorem 8.1.

For an integer t going from 1 to $\min(n, m)$, supposed to *guess* the grid number of M, we test if M has a t-grid minor with the sentence

$$\exists x_0 \exists x_1 \cdots \exists x_{t-1}\ \exists y_0 \exists y_1 \cdots \exists y_{t-1}$$

$$R(x_0) \wedge \neg R(y_0) \wedge \forall x'(R(x') \to x_0 \preceq x') \wedge \forall y'(\neg R(y') \to y_0 \preceq y') \wedge$$

$$\bigwedge_{i\in[t-1]} R(x_i) \wedge \bigwedge_{i\in[t-1]} \neg R(y_i) \wedge \bigwedge_{i\in[t-2]} (x_i \prec x_{i+1} \wedge y_i \prec y_{i+1}) \wedge$$

$$\bigwedge_{0\leqslant i,j\leqslant t-2} (\exists x\ \exists y\ x_i \preceq x \prec x_{i+1} \ \wedge\ y_j \preceq y \prec y_{j+1} \ \wedge\ E_1(x,y)) \ \wedge$$

$$\bigwedge_{0\leqslant i\leqslant t-2} (\exists x\ \exists y\ x_i \preceq x \prec x_{i+1} \ \wedge\ y_{t-1} \preceq y \ \wedge\ E_1(x,y)) \ \wedge$$

$$\bigwedge_{0\leqslant j\leqslant t-2} (\exists x\ \exists y\ x_{t-1} \preceq x \ \wedge\ y_j \preceq y \prec y_{j+1} \ \wedge\ E_1(x,y)) \ \wedge$$

$$\exists x\ \exists y\ x_{t-1} \preceq x \ \wedge\ y_{t-1} \preceq y \ \wedge\ E_1(x,y),$$

where E_1 interprets the 1-entries of M. At the first value of t for which the model checking algorithm reports a no-instance, we output $t-1$. □

3.2. Algorithms on classes of unbounded twin-width. Algorithms made possible by the theory on twin-width, but operating on classes of unbounded twin-width, use a win-win argument, and follow in the footsteps of [**51**]. The argument distinguishes two covering cases: the twin-width is either above or below a certain threshold. In the former case, the existence of a large pattern as depicted in Figure 10.4 provides enough information to efficiently conclude. In the latter case, we resort to the algorithm of Theorem 7.4. This of course requires that the problem at hand is definable in first-order logic, possibly with modulo counting. We finish with an example contrasting with that, in the exact same setting, such an algorithm for k-Dominating Set would refute the ETH [**28**].

Theorem 10.9 ([**16**]). *k-Independent Set in visibility graphs of simple polygons given with a representation is fixed-parameter tractable.*

Sketch. Let $\mathcal{P}$ be a simple polygon, and G, its visibility graph. We identify vertices of G with the corresponding vertices of $\mathcal{P}$. Let $\prec$ be a linear order whose successor relation is a Hamiltonian path of the boundary of $\mathcal{P}$. We run Theorem 10.2 on $(G, \prec)$ with d being a large (but fixed) function of k. If the algorithm outputs a contraction sequence, we solve k-Independent Set by Theorem 7.3. If instead we obtain a large rich division, the algorithm simply answers yes. Let us justify this decision.

Visibility graphs of simple polygons satisfy that: If $b' \prec a \prec b \prec c \prec d \prec c'$ and $\{ac, bd, ac', b'd\} \subseteq E(G)$, then $ad \in E(G)$; see Figure 10.5.

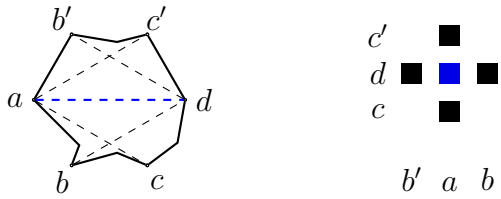


Figure 10.5. Above or below the diagonal of $A_\prec(G)$ no 0-entry can have four "surrounding" 1-entries.

This precludes the second pattern of Figure 10.4 in $A_\prec(G)$.

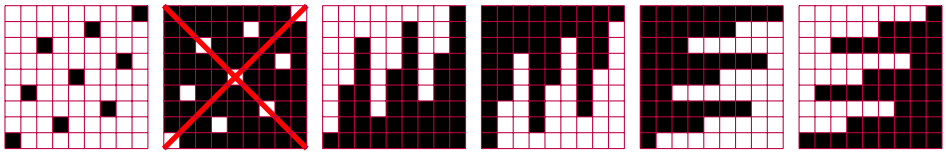


FIGURE 10.6. The pattern ruled out by Figure 10.5.

That is 4 of the 25 families of ordered graphs that cannot occur. The algorithm is correct to output yes if we get a representative of the family of permutation graphs with their natural order, or of any family where one of the sides X, Y of the biadjacency matrix is an independent set. This leaves us with the five families of Figure 10.6 when both sides X, Y are cliques. Each would imply the existence of $x_1 \prec x_2 \prec x_3 \prec x_4 \in X$, and $y_4 \prec y_3 \prec y_2 \prec y_1 \in Y$ such that $x_i y_j \in E(G)$ if and only if $i = j$ ($i \leqslant j$, $i \geqslant j$, respectively). Let us write $X' := \{x_1, x_2, x_3, x_4\}$ and $Y' := \{y_1, y_2, y_3, y_4\}$.

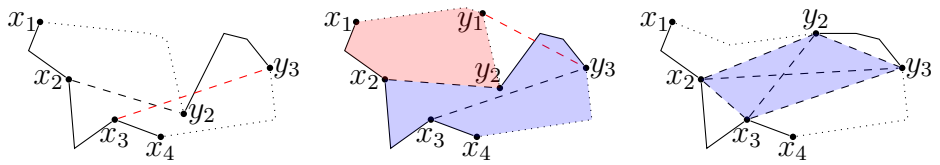


FIGURE 10.7. If quadrangle $q := x_2x_3y_3y_2$ self-intersects, one edge of x_2y_2, x_3y_3 would be missing (left), if q is not convex, one of X', Y' would not be a clique (middle), and the edges $x_2x_3, x_2y_2, x_3y_3, y_2y_3$ force q to be included in $\mathcal{P}$, thus x_2y_3 and x_3y_2 are edges (right).

Figure 10.7 is a proof by picture that the edges x_2y_2, x_3y_3 and the fact that X', Y' are cliques force both x_2y_3 and x_3y_2 to be in $E(G)$, thus not fitting the pattern. □

Bibliography

[1] Jungho Ahn, Debsoumya Chakraborti, Kevin Hendrey, and Sang-il Oum. Twin-width of subdivisions of multigraphs. *CoRR*, abs/2306.05334, 2023.

[2] Stephen Alstrup, Søren Dahlgaard, and Mathias Bæk Tejs Knudsen. Optimal induced universal graphs and adjacency labeling for trees. *J. ACM*, 64(4):27:1–27:22, 2017.

[3] Jakub Balabán and Petr Hlinený. Twin-width is linear in the poset width. In Petr A. Golovach and Meirav Zehavi, editors, *16th International Symposium on Parameterized and Exact Computation, IPEC 2021, September 8-10, 2021, Lisbon, Portugal*, volume 214 of *LIPIcs*, pages 6:1–6:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[4] Jakub Balabán, Petr Hlinený, and Jan Jedelský. Twin-width and transductions of proper k-mixed-thin graphs. *CoRR*, abs/2202.12536, 2022.

[5] John T. Baldwin and Saharon Shelah. Second-order quantifiers and the complexity of theories. *Notre Dame Journal of Formal Logic*, 26(3):229–303, 1985.

[6] József Balogh, Béla Bollobás, and Robert Morris. Hereditary properties of partitions, ordered graphs and ordered hypergraphs. *Eur. J. Comb.*, 27(8):1263–1281, 2006.

[7] Ambroise Baril, Miguel Couceiro, and Victor Lagerkvist. Linear bounds between component twin-width and clique-width with algorithmic applications to counting graph colorings. 2023.

[8] Lowell W Beineke and Raymond E Pippert. The number of labeled k-dimensional trees. *Journal of Combinatorial Theory*, 6(2):200–205, 1969.

[9] Pierre Bergé, Édouard Bonnet, and Hugues Déprés. Deciding twin-width at most 4 is NP-complete. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[10] Pierre Bergé, Édouard Bonnet, Hugues Déprés, and Rémi Watrigant. Approximating highly inapproximable problems on graphs of bounded twin-width. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 10:1–10:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[11] Umberto Bertele and Francesco Brioschi. On non-serial dynamic programming. *J. Comb. Theory, Ser. A*, 14(2):137–148, 1973.

[12] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap*. *Combinatorica*, 26(5):495–519, 2006.

[13] Robin L. Blankenship. *Book embeddings of graphs*. Louisiana State University and Agricultural & Mechanical College, 2003.

[14] Édouard Bonnet, Romain Bourneuf, Julien Duron, Colin Geniet, Stéphan Thomassé, and Nicolas Trotignon. A tamed family of triangle-free graphs with unbounded chromatic number. *arXiv preprint arXiv:2304.04296*, 2023.

[15] Édouard Bonnet, Romain Bourneuf, Colin Geniet, and Stéphan Thomassé. Factoring pattern-free permutations into separable ones. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 752–779. SIAM, 2024.

[16] Édouard Bonnet, Dibyayan Chakraborty, Eun Jung Kim, Noleen Köhler, Raul Lopes, and Stéphan Thomassé. Twin-width VIII: delineation and win-wins. In Holger Dell and Jesper Nederlof, editors, *17th International Symposium on Parameterized and Exact Computation, IPEC 2022, September 7-9, 2022, Potsdam, Germany*, volume 249 of *LIPIcs*, pages 9:1–9:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[17] Édouard Bonnet and Hugues Déprés. Twin-width can be exponential in treewidth. *J. Comb. Theory, Ser. B*, 161:1–14, 2023.

[18] Édouard Bonnet, Jan Dreier, Jakub Gajarský, Stephan Kreutzer, Nikolas Mählmann, Pierre Simon, and Szymon Torunczyk. Model checking on interpretations of classes of bounded local cliquewidth. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 54:1–54:13. ACM, 2022.

[19] Édouard Bonnet and Julien Duron. Stretch-width. In Neeldhara Misra and Magnus Wahlström, editors, *18th International Symposium on Parameterized and Exact Computation, IPEC 2023, September 6-8, 2023, Amsterdam, The Netherlands*, volume 285 of *LIPIcs*, pages 8:1–8:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[20] Édouard Bonnet, Julien Duron, John Sylvester, Viktor Zamaraev, and Maksim Zhukovskii. Small but unwieldy: A lower bound on adjacency labels for small classes. In *SODA 2024*, 2024.

[21] 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.

[22] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. *Combinatorial Theory*, 2(2), 2022.

[23] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Torunczyk. Twin-width IV: ordered graphs and matrices. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 924–937. ACM, 2022.

[24] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, and Stéphan Thomassé. Twin-width V: linear minors, modular counting, and matrix multiplication. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 15:1–15:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[25] Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, and Stéphan Thomassé. Twin-width VI: the lens of contraction sequences. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1036–1056. SIAM, 2022.

[26] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022.

[27] Édouard Bonnet, O-joung Kwon, and David R. Wood. Reduced bandwidth: a qualitative strengthening of twin-width in minor-closed classes (and beyond). *CoRR*, abs/2202.11858, 2022.

[28] Édouard Bonnet and Tillmann Miltzow. Parameterized hardness of art gallery problems. *ACM Trans. Algorithms*, 16(4):42:1–42:23, 2020.

[29] Édouard Bonnet, Jaroslav Nesetril, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations. *CoRR*, abs/2102.06880, 2021.

[30] Romain Bourneuf and Stéphan Thomassé. Bounded twin-width graphs are polynomially χ-bounded. *CoRR*, abs/2303.11231, 2023.

[31] Samuel Braunfeld and Michael C. Laskowski. Characterizations of monadic NIP, 2021.

[32] Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Boolean-width of graphs. *Theor. Comput. Sci.*, 412(39):5187–5204, 2011.

[33] Parinya Chalermsook, Bundit Laekhanukit, and Danupon Nanongkai. Graph products revisited: Tight approximation hardness of induced matching, poset dimension and more. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1557–1576. SIAM, 2013.

[34] Josef Cibulka and Jan Kyncl. Füredi-Hajnal limits are typically subexponential. *CoRR*, abs/1607.07491, 2016.

[35] Thomas Colcombet. A combinatorial theorem for trees. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 901–912. Springer, 2007.

[36] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.

[37] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12 – 75, 1990.

[38] 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.

[39] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633. ACM, 2014.

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

[41] Jan Dreier, Ioannis Eleftheriadis, Nikolas Mählmann, Rose McCarty, Michal Pilipczuk, and Szymon Torunczyk. First-order model checking on monadically stable graph classes. *CoRR*, abs/2311.18740, 2023.

[42] Zdenek Dvorák and Daniel Král'. Classes of graphs with small rank decompositions are x-bounded. *Eur. J. Comb.*, 33(4):679–683, 2012.

[43] David Eppstein. The widths of strict outerconfluent graphs. *arXiv preprint arXiv:2308.03967*, 2023.

[44] Pául Erdos. On an elementary proof of some asymptotic formulas in the theory of partitions. *Annals of Mathematics*, pages 437–450, 1942.

[45] Fedor V. Fomin, Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensionality. In *Encyclopedia of Algorithms*, pages 203–207. 2016.

[46] Jacob Fox. Stanley-Wilf limits are typically exponential. *CoRR*, abs/1310.8378, 2013.

[47] Zoltán Füredi and Péter Hajnal. Davenport-schinzel theory of matrices. *Discret. Math.*, 103(3):233–251, 1992.

[48] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *J. Comput. Syst. Sci.*, 22(3):407–420, 1981.

[49] Jakub Gajarský, Michal Pilipczuk, Wojciech Przybyszewski, and Szymon Torunczyk. Twin-width and types. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 123:1–123:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[50] Fred Galvin. A proof of Dilworth's chain decomposition theorem. *The American Mathematical Monthly*, 101(4):352–353, 1994.

[51] Sylvain Guillemot and Dániel Marx. Finding small patterns in permutations in linear time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 82–101, 2014.

[52] 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.

[53] Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. *Comput. Sci. Rev.*, 4(1):41–59, 2010.

[54] Rudolf Halin. S-functions for graphs. *Journal of geometry*, 8:171–186, 1976.

[55] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 627–636, 1996.

[56] Hamed Hatami and Pooya Hatami. The implicit graph conjecture is false. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1134–1137. IEEE, 2022.

[57] Petr Hlinený and Jan Jedelský. Twin-width of planar graphs is at most 8, and at most 6 when bipartite planar. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 75:1–75:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[58] Petr Hlinený and Filip Pokrývka. Twin-width and limits of tractability of FO model checking on geometric graphs. *CoRR*, abs/2204.13742, 2022.

[59] Hugo Jacob and Marcin Pilipczuk. Bounding twin-width for bounded-treewidth graphs, planar graphs, and bipartite graphs. In Michael A. Bekos and Michael Kaufmann, editors, *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers*, volume 13453 of *Lecture Notes in Computer Science*, pages 287–299. Springer, 2022.

[60] David S. Johnson. Approximation algorithms for combinatorial problems. In Alfred V. Aho, Allan Borodin, Robert L. Constable, Robert W. Floyd, Michael A. Harrison, Richard M. Karp, and H. Raymond Strong, editors, *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1973, Austin, Texas, USA*, pages 38–49. ACM, 1973.

[61] Shahin Kamali. Compact representation of graphs of small clique-width. *Algorithmica*, 80(7):2106–2131, 2018.

[62] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM J. Discret. Math.*, 5(4):596–603, 1992.

[63] Martin Klazar. The Füredi-Hajnal conjecture implies the Stanley-Wilf conjecture. In *Formal power series and algebraic combinatorics*, pages 250–255. Springer, 2000.

[64] Daniel Král and Ander Lamaison. Planar graph with twin-width seven. *CoRR*, abs/2209.11537, 2022.

[65] Daniel Král, Kristýna Pekárková, and Kenny Storgel. Twin-width of graphs on surfaces. *CoRR*, abs/2307.05811, 2023.

[66] Stephan Kreutzer and Siamak Tazari. Lower bounds for the complexity of monadic second-order logic. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 189–198. IEEE Computer Society, 2010.

[67] Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[68] László Lovász. On the ratio of optimal integral and fractional covers. *Discret. Math.*, 13(4):383–390, 1975.

[69] Adam Marcus and Gábor Tardos. Excluded permutation matrices and the Stanley-Wilf conjecture. *J. Comb. Theory, Ser. A*, 107(1):153–160, 2004.

[70] Jan Mycielski. Sur le coloriage des graphs. In *Colloquium Mathematicae*, volume 3, pages 161–162, 1955.

[71] Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.

[72] 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.

[73] Damian Osajda. Small cancellation labellings of some infinite graphs and applications. 2020.

[74] Richard Otter. The number of trees. *Annals of Mathematics*, pages 583–599, 1948.

[75] Michal Pilipczuk and Marek Sokolowski. Graphs of bounded twin-width are quasi-polynomially χ-bounded. *J. Comb. Theory, Ser. B*, 161:382–406, 2023.

[76] Michal Pilipczuk, Marek Sokolowski, and Anna Zych-Pawlewicz. Compact representation for matrices of bounded twin-width. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 52:1–52:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[77] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry - An Introduction*. Texts and Monographs in Computer Science. Springer, 1985.

[78] Neil Robertson and Paul D. Seymour. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.

[79] Edward R. Scheinerman and Jennifer S. Zito. On the size of hereditary classes of graphs. *J. Comb. Theory, Ser. B*, 61(1):16–39, 1994.

[80] Detlef Seese. The structure of models of decidable monadic theories of graphs. *Ann. Pure Appl. Log.*, 53(2):169–195, 1991.

[81] Martin Vatshelle. New width parameters of graphs. 2012.

[82] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.