Stretch-Width

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

Julien Duron
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Abstract

We introduce a new parameter, called stretch-width, that we show sits strictly between clique-width and twin-width. Unlike the reduced parameters [BKW '22], planar graphs and polynomial subdivisions do not have bounded stretch-width. This leaves open the possibility of efficient algorithms for a broad fragment of problems within Monadic Second-Order (MSO) logic on graphs of bounded stretch-width. In this direction, we prove that graphs of bounded maximum degree and bounded stretch-width have at most logarithmic treewidth. As a consequence, in classes of bounded stretch-width, Maximum Independent Set can be solved in subexponential time $2^{\tilde{O}(n^{8/9})}$ on n-vertex graphs, and, if further the maximum degree is bounded, Existential Counting Modal Logic [Pilipczuk '11] can be model-checked in polynomial time. We also give a polynomial-time $O(\text{OPT}^2)$-approximation for the stretch-width of symmetric $0,1$-matrices or ordered graphs.

Somewhat unexpectedly, we prove that exponential subdivisions of bounded-degree graphs have bounded stretch-width. This allows to complement the logarithmic upper bound of treewidth with a matching lower bound. We leave as open the existence of an efficient approximation algorithm for the stretch-width of unordered graphs, if the exponential subdivisions of all graphs have bounded stretch-width, and if graphs of bounded stretch-width have logarithmic clique-width (or rank-width).

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Design and analysis of algorithms

Keywords and phrases Contraction sequences, twin-width, clique-width, algorithms, algorithmic metatheorems

Digital Object Identifier 10.4230/LIPIcs.IPEC.2023.8

Related Version *Full Version*: https://arxiv.org/abs/2305.12023

1 Introduction

Various graph classes have bounded *twin-width*[1] such as, for instance, bounded clique-width graphs, proper minor-closed classes, proper hereditary subclasses of permutation graphs, and some expander classes [11]. Low twin-width, together with the witnessing *contraction sequences*, enables parameterized algorithms (that are unlikely in general graphs) for testing if a graph satisfies a first-order sentence [11, 7], and improved approximation algorithms for highly inapproximable packing and coloring problems [7, 4].

However one should not expect a large gain, in the low twin-width regime, as far as (non-parameterized) exact algorithms are concerned. This is because every graph obtained by subdividing (at least) $2\lceil \log n \rceil$ times each edge of an n-vertex graph G has twin-width at most 4 [3]. It was already observed in the 70's that a problem like Maximum Independent Set (MIS, for short) remains NP-complete in $2t$-subdivisions [23]. Furthermore, known reductions [16] combined with the Sparsification Lemma [20], imply that unless the

[1] We refer the reader to Section 2 for the relevant definitions.

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström; Article No. 8; pp. 8:1–8:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Exponential-Time Hypothesis[2] (ETH) fails [19] solving MIS in subcubic graphs requires time $2^{\Omega(n)}$. The previous remarks entail that, unless the ETH fails, solving MIS in subcubic graphs of twin-width at most 4 requires time $2^{\Omega(n/\log n)}$.

In contrast, on the significantly less general classes of bounded clique-width not only can MIS be solved in polynomial-time, but a fixed-parameter algorithm solving MSO_1[3] model checking in time $f(w, |\varphi|)n^{O(1)}$ exists [14, 21], with w the clique-width of the input graph, φ the input sentence, and f some computable function.

In this paper, we start exploring the trade-off between class broadness and algorithmic generality in the zone delimited by bounded clique-width and bounded twin-width. It may seem like the *reduced parameters* [12], where a graph has *reduced* p at most k if it admits a contraction sequence in which all the red graphs have parameter p at most k, are exactly designed to tackle this endeavor. Indeed by definition, twin-width is *reduced* Δ, where Δ is the maximum degree, and it was shown that *reduced maximum connected component size* (under the name of *component twin-width*) is functionally equivalent to clique-width [10]. Between *maximum connected component size* and *maximum degree*, there are several parameters p, such as bandwidth, cutwidth, treewidth+Δ, whose reduced parameters give rise to a strict [12] hierarchy between bounded clique-width and bounded twin-width. Unfortunately, even reduced bandwidth –the closest to clique-width among the above-mentioned reduced parameters– turns out to be too general in the following sense: the n-subdivision of any n-vertex graph has reduced bandwidth at most 2 [12]. This means, by the arguments of the second paragraph of this introduction, that solving MIS on graphs of bounded reduced bandwidth requires time $2^{\Omega(\sqrt{n})}$, unless the ETH fails, even among graphs of bounded degree. Actually, another fact leading to the same conclusion is that planar graphs have bounded reduced bandwidth [12].

We therefore introduce another parameter, that we call stretch-width[4] and denote by stw, which, while inspired by reduced parameters, does not fully fit that framework. To a first approximation, *stretch-width* can be thought as *reduced bandwidth* where the bandwidth upper bound on the red graphs have to be witnessed by a single (and fixed) order on the vertex set. Observe indeed that the linear orders witnessing that all the red graphs of the sequence have low bandwidth can, in reduced bandwidth, be very different one from the other. We first show that the family of bounded stretch-width classes strictly contains the family of bounded clique-width classes. Using an upper bound of component twin-width by clique-width [2], we prove that:

▶ **Theorem 1** ($\star$). *The stretch-width of any graph is at most twice its clique-width.*

All the statements marked with a $\star$ are only proved in the long version, in appendix.

Then we provide a separating class of bounded stretch-width and unbounded clique-width.

▶ **Theorem 2** ($\star$). *There is an infinite family of graphs G with bounded stretch-width and clique-width $\Omega(\log |V(G)|)$.*

As was done for twin-width [9], we give an effective characterization of bounded stretch-width for symmetric $0,1$-matrices (or ordered graphs).

▶ **Theorem 3** ($\star$). *A class $\mathcal{C}$ of symmetric $0,1$-matrices has bounded stretch-width if and only if there is an integer k such that no matrix of $\mathcal{C}$ has a k-wide division.*

---

[2] the assumption that there is a $\lambda > 1$ such that n-variable 3-SAT cannot be solved in time $\lambda^n n^{O(1)}$
[3] Monadic Second-Order logic, when the second-order variables can only be vertex subsets
[4] We refer a reader who would already want a formal definition to the start of Section 3.

The *k-wide division* (see definition in Section 4) is a scaled-down version of the *k-rich division* that analogously characterizes matrices of bounded twin-width [9]. Theorem 3 yields a polynomial-time approximation algorithm for the stretch-width of symmetric $0,1$-matrices. More precisely:

▶ **Theorem 4** ($\star$). *Given an integer k and a symmetric $n \times n$ $0,1$-matrix M, there is an $n^{O(1)}$-time algorithm that outputs a sequence witnessing that $stw(M) = O(k^3)$ or correctly reports that $stw(M) > k$.*

Compared to the approximation algorithm for the twin-width of a matrix, this is better both in terms of running time (polynomial vs fixed-parameter tractable) and approximation factor (quadratic vs exponential).

Conveniently for the sought algorithmic applications, planar graphs and n^c-subdivisions of n-vertex graphs (for any constant c) both have *unbounded* stretch-width (whereas they have bounded reduced bandwidth if $c \geqslant 1$). We indeed establish the following upper bound on treewidth, implying that graphs of bounded maximum degree and bounded stretch-width have at most logarithmic treewidth.

▶ **Theorem 5.** *There is a c such that for every graph G, $tw(G) \leqslant c\Delta(G)^2 stw(G)^4 \log |V(G)|$.*

We match Theorem 5 with a lower bound. There are graphs with bounded $\Delta + \text{stw}$ and treewidth growing as a logarithm of their number of vertices. This is because, as we prove, very long subdivisions of bounded-degree graphs have bounded stretch-width.

▶ **Theorem 6.** *Every $(\geqslant n2^m)$-subdivision of every n-vertex m-edge graph G of maximum degree d has stretch-width at most $32(4d + 5)^3$.*

By $(\geqslant s)$*-subdivision* of G, we mean every graph obtained by subdividing each edge of G at least s times. In particular, for every natural k, the n-vertex $k^2 2^{2k(k-1)}$-subdivision of the $k \times k$ grid has bounded maximum degree (by 4) and stretch-width (by 296352), whereas it has treewidth $k = \Omega(\sqrt{\log n})$. A more careful argument and reexamination of Theorem 6 show that, for some constant c, the n-vertex 2^{ck}-subdivision of the $k \times k$ grid has bounded $\Delta + \text{stw}$, and treewidth $k = \Omega(\log n)$ matching the upper bound of Theorem 5.

The proofs of Theorems 5 and 6 involve the notion of *overlap graph* of a graph G whose vertex set is totally ordered by $\prec$, with one vertex per edge of G, and an edge between two "overlapping edges" of G, that is, two edges ab and cd such that $a \prec c \prec b \prec d$. Using Theorem 3, we show that finding a vertex ordering such that the overlap graph has no large biclique allows to bound the stretch-width.

▶ **Lemma 7.** *For every ordered graph $(G, \prec)$ and every integer t, if the overlap graph of $(G, \prec)$ has no $K_{t,t}$ subgraph then $stw(G) < 32(2t + 1)^3$.*

Theorem 6 is then derived by designing a long subdivision process that, for ordered graphs of maximum degree d, reduces the bicliques in the overlap graph to a size at most linear in d.

Theorem 5 has direct algorithmic implications for classes of bounded stretch-width.

▶ **Proposition 8.** *There is an algorithm that solves Max Independent Set in graphs of bounded stretch-width with running time $2^{\tilde{O}(n^{8/9})}$.*

Pilipczuk [22] showed that any problem expressible in Existential Counting Modal Logic (ECML) admits a single-exponential fixed-parameter algorithm in treewidth. In particular, ECML model checking can be solved in polynomial time in any class with logarithmic treewidth. This logic allows existential quantifications over vertex and edge sets followed by an *arithmetic formula* and a *counting modal formula* that shall be satisfied from every vertex v. The arithmetic formula is a quantifier-free expression that may involve the cardinality of the vertex and edge sets, as well as integer parameters. Counting modal formulas enrich quantifier-free Boolean formulas with $\Diamond^S\varphi$, whose semantics is that the current vertex v has a number of neighbors satisfying φ in a prescribed ultimately periodic set S of non-negative integers.

The logic ECML+C gives to ECML the power of also using in the arithmetic formula the number of connected components in subgraphs induced by some vertex or edge sets. There is a Monte-Carlo polynomial-time algorithm for ECML+C in graphs of treewidth at most a logarithm function in their number of vertices [22]. Most NP-hard graphs problems, such as Maximum Independent Set, Minimum Dominating Set, Steiner Tree, etc. are expressible in ECML+C; see [22, Appendix D] for the ECML+C formulation of several examples.

▶ **Corollary 9.** *Problems definable in ECML (resp. ECML+C) can be solved in polynomial time (resp. randomized polynomial time) in bounded-degree graphs of bounded stretch-width.*

Perspectives. Proposition 8 and Corollary 9 constitute some preliminary pieces of evidence of the algorithmic amenability of classes of bounded stretch-width. We ask several questions. How can the running time of Proposition 8 be improved? (As far as we know, there could be a polynomial-time algorithm for any problem defined in ECML on graphs of bounded stretch-width.) As for twin-width, an approximation algorithm for stretch-width of (unordered) graphs remains open. Lemma 7 gives some hope that this question might be easier than its twin-width counterpart, especially among *sparse* graphs.

Can we lift the bounded-degree requirement in Theorem 6, that is, is there a function f and a constant c, such that the stretch-width of any ($\geqslant f(n)$)-subdivision of any n-vertex graph is at most c? Our separating example showing that bounded stretch-width is strictly more general than bounded clique-width (Theorem 2) yields graphs of essentially logarithmic clique-width. Is that true in general?

▶ **Conjecture 10.** *For every class $\mathcal{C}$ of bounded stretch-width, there is a constant c such that for every n-vertex graph $G \in \mathcal{C}$ the clique-width of G is at most $c \log n$.*

We ask the same question with *rank-width* instead of *clique-width*, which would be more algorithmically helpful. One interpretation of Theorem 5 is that graphs of bounded maximum degree and bounded stretch-width have logarithmic treewidth. Whether the *bounded-degree* constraint can be relaxed to the mere absence of large bipartite complete subgraphs is related to Conjecture 10. A positive answer to Conjecture 10 would indeed imply this relaxation, as Gurski and Wanke have shown that graphs without $K_{t,t}$ subgraphs have treewidth at most their clique-width times $3t$ [18]. A natural future work would consist of using the witness of low stretch-width to get improved algorithms compared to those attained with a witness of low twin-width.

Related work. Our work is in line with twin-width [11], and the reduced parameters [12]. Theorem 1 closely follows a similar proof in the sixth paper of the twin-width series [10], while Theorem 3 is inspired by the fourth paper [9], and notably the so-called *rich divisions*.

Finding the right logic for a given width parameter, or the right width parameter for a given logic has been a common goal ever since Courcelle's and Courcelle-Makowsky-Rotics's theorems [13, 14] relating treewidth with MSO_2, and clique-width with MSO_1.

Recent developments (all from 2023) include an efficient model checking of the new logic A&C DN (an extension of Existential MSO_1) on classes of bounded mim-width [5], the new parameter flip-width [26], which could lead to an efficient first-order (FO) model checking in a very general class, and efficient model checking algorithms for FO extensions with disjoint-paths predicates in proper minor-closed classes [17], and in proper topological-minor-closed classes [24].

Classes with logarithmic treewidth, although not a priori defined as such, are somewhat rare. To our knowledge, the first such example is the class of triangle-free graphs with no theta (see [25] for the lower bound, and [1], for the upper bound). Another example consists of graphs without $K_{t,t}$ subgraph and bounded induced cycle packing number [6]. We add a new family: graphs of bounded maximum degree and bounded stretch-width. Note that these three families are all incomparable.

2 Preliminaries

For $i \leqslant j$ two integers, we denote the set of integers that are at least i and at most j by $[i, j]$, and $[i]$ is a short-hand for $[1, i]$. We use the standard graph-theoretic notations. In particular, for a graph G, we denote by $V(G)$ its set of vertices and by $E(G)$ its set of edges. If $S \subseteq V(G)$, the *subgraph of G induced by S*, denoted $G[S]$ is the graph obtained from G by removing the vertices not in S.

2.1 Contraction sequences and twin-width

Twin-width is a graph parameter introduced by Bonnet, Kim, Thomassé, and Watrigant [11]. A possible definition involves the notions of *trigraphs*, *red graphs*, and *contraction sequences*. A *trigraph* is a graph with two types of edges: black (regular) edges and red (error) edges. The *red graph* $\mathcal{R}(H)$ of a trigraph H consists of ignoring its black edges, and considering its red edges as being normal (black) edges. We may say *red neighbor* (or *red neighborhood*) to simply mean a neighbor (or neighborhood) in the red graph. A (vertex) *contraction* consists of merging two (non-necessarily adjacent) vertices, say, u, v into a vertex w, and keeping every edge wz black if and only if uz and vz were previously black edges. The other edges incident to w become red (if not already), and the rest of the trigraph remains the same. A *contraction sequence* of an n-vertex graph G is a sequence of trigraphs $G = G_n$, $\dots, G_1$ such that G_i is obtained from G_{i+1} by performing one contraction. A *d-sequence* is a contraction sequence in which every vertex of every trigraph has at most d red edges incident to it. In other words, every red graph of the sequence has maximum degree at most d. The *twin-width* of G, denoted by $\mathrm{tww}(G)$, is then the minimum integer d such that G admits a d-sequence. Figure 1 gives an example of a graph with a 2-sequence, i.e., of twin-width at most 2.

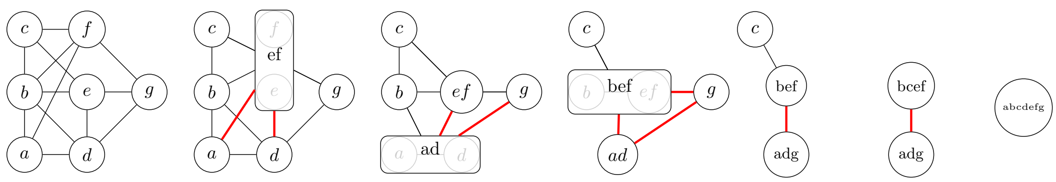

Figure 1 A 2-sequence witnessing that the initial graph has twin-width at most 2.

2.2 Partition sequences

Partition sequences yield an equivalent viewpoint to contraction sequences. Instead of dealing with a sequence of trigraphs $G = G_n, \ldots, G_1$, we now have a sequence of partitions $\mathcal{P}_n, \ldots, \mathcal{P}_1$ of $V(G)$, with $\mathcal{P}_n = \{\{v\} \mid v \in V(G)\}$ 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 $\mathcal{P}_1 = \{V(G)\}$. Now one can obtain the red graph $\mathcal{R}(G_i)$ of G_i, as the graph whose vertices are the parts of $\mathcal{P}_i$, and whose edges link two parts $X \neq Y \in P_i$ 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, or no vertex of X is adjacent to a vertex of Y. We will also denote $\mathcal{R}(G_i)$ by $\mathcal{R}(\mathcal{P}_i)$.

2.3 Separation number

When dealing with treewidth in Section 7 it will more convenient to think of it in terms of the functionally equivalent *separation number*. A *separation* (A, B) of a graph G is such that $A \cup B = V(G)$ and there is no edge between $A \setminus B$ and $B \setminus A$. The *order* of the separation (A, B) is $|A \cap B|$. A separation (A, B) is *balanced* if $\max(|A \setminus B|, |B \setminus A|) \leqslant \frac{2}{3}|V(G)|$. The *separation number* $\mathrm{sn}(G)$ of G is the smallest integer s such that every subgraph of G admits a balanced separation of order at most s. It is not difficult to show that for every graph G, $\mathrm{sn}(G) \leqslant \mathrm{tw}(G) + 1$. Dvorák and Norin showed the converse linear dependence:

▶ **Lemma 11** ([15]). *For every graph G, $tw(G) \leqslant 15sn(G)$.*

Note that if for some positive constant $c < 1$, every subgraph H of G has a separation (A, B) that is *c-balanced*, in the sense that $\max(|A \setminus B|, |B \setminus A|) \leqslant c|V(H)|$ of order at most s, then every subgraph of G has a balanced separation of order $\lceil \frac{\log c}{\log(2/3)} \rceil \cdot s$. In particular, by Lemma 11, $\mathrm{tw}(G) = O(s)$.

3 Stretch-width

An ordered graph is a pair $(G, \prec)$ where G is a graph and $\prec$ a strict total order on $V(G)$. We write $u \preccurlyeq v$ whenever $u \prec v$ or $u = v$. Let $(G, \prec)$ is an ordered graph, and $X \subseteq V(G)$. We now define some objects depending on $\prec$, but as the order will be clear from the context, we omit it from the corresponding notations.

The minimum and maximum of X along $\prec$ are denoted by $\min(X)$ and $\max(X)$, respectively. The *convex closure* or *span* of X is $\mathrm{conv}(X) := \{v \in V(G) \mid \min(X) \preccurlyeq v \preccurlyeq \max(X)\}$. Two sets $X, Y \subseteq V(G)$ are *in conflict*[5], or X *conflicts with* Y, if $\mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset$. Note that this does not imply that X and Y themselves intersect, and indeed we will mostly use this notion for two disjoint sets X, Y.

Let now $\mathcal{P}$ be a partition of $V(G)$, $\mathcal{R}(\mathcal{P})$ its red graph, and $X \in \mathcal{P}$. We say that $Y \in \mathcal{P} \setminus \{X\}$ *interferes* with X if Y conflicts with $N_{\mathcal{R}(\mathcal{P})}[X]$. Note that it may well be that Y interferes with X, but not vice versa. The *stretch* of the part $X \in \mathcal{P}$, denoted by $\mathrm{str}(X)$, is then defined as the number of parts in $\mathcal{P}$ interfering with X. In turn, the *stretch* of $\mathcal{P}$ is the maximum over every part $Z \in \mathcal{P}$ of $\mathrm{str}(Z)$. The *stretch-width* of the ordered graph $(G, \prec)$, denoted by $\mathrm{stw}(G, \prec)$, is the minimum, taken among every partition sequence $\mathcal{P}_n, \ldots, \mathcal{P}_1$ of G, of $\max_{i \in [n]} \mathrm{str}(\mathcal{P}_i)$. Finally the *stretch-width* of G, denoted by $\mathrm{stw}(G)$, is the minimum of $\mathrm{stw}(G, \prec)$ taken among every total order $\prec$ on $V(G)$.

[5] In a similar context in [9], the verb *overlap* was also used. In this paper, we will reserve *overlap* for intersecting intervals (actually edges) that are not nested, notion which we will later use.

4 Matrix characterization

Let us first reinterpret the definition of stretch-width on symmetric (ordered) matrices. A (symmetric) *partition* of a (symmetric) matrix M is a pair $(\mathcal{R}, \mathcal{C})$ such that $\mathcal{R}$ is a partition of the row set of M, $\text{rows}(M)$, $\mathcal{C}$ is a partition of the column set, $\text{columns}(M)$, and $\mathcal{C}$ is symmetric to $\mathcal{R}$, i.e., two rows r_i and r_j are in the same part if and only if the symmetric columns c_i and c_j are in the same part. Hence, each row part corresponds to a (unique) symmetric column part. A *division* of a (symmetric) matrix M is a partition of M every row (resp. column) part of which is on consecutive rows (resp. columns). Given a row part $R \in \mathcal{R}$, and a column part $C \in \mathcal{C}$, the *zone* $R \cap C$ of M is the submatrix of M with row set R and column set C. The *diagonal zone* of $R \in \mathcal{R}$ is the zone $R \cap C$ where C is the symmetric part of R in columns. A zone is *non-constant* if it contains two distinct entries. A zone of a division may be called *cell*. A *partition sequence* of an $n \times n$ $0,1$-matrix M is a sequence $(\mathcal{R}_1, \mathcal{C}_1), \ldots, (\mathcal{R}_{n-1}, \mathcal{C}_{n-1})$ where $(\mathcal{R}_1, \mathcal{C}_1)$ is the *finest partition* (with n row parts and n column parts), $(\mathcal{R}_{n-1}, \mathcal{C}_{n-1})$ is the coarsest partition (with one row part and one column part), and for every $i \in [n-2]$, $(\mathcal{R}_{i+1}, \mathcal{C}_{i+1})$ is obtained by merging together two row parts (and the symmetric two column parts) of $(\mathcal{R}_i, \mathcal{C}_i)$.

So far, we were following the definitions of [11, 8]. Instead of defining the *error value* which leads to the twin-width of a matrix, we introduce the *stretch value*. The *stretch value* of a row part R of a matrix partition $(\mathcal{R}, \mathcal{C})$ is the number of row parts conflicting with R plus the number column parts conflicting with the union of columns parts C such that $R \cap C$ is non-constant or $R \cap C$ is diagonal. The *stretch value* of a column part is defined symmetrically. The *stretch value* of a partition $(\mathcal{R}, \mathcal{C})$ is the maximum stretch value of a part of $(\mathcal{R}, \mathcal{C})$. Finally, one can define the *stretch-width* of a $0,1$-matrix M as the minimum among every partition sequence $\mathcal{S}$ of M of the maximum stretch value among partitions of $\mathcal{S}$. Observe that for any ordered graph $(G, \prec)$, the stretch-width of $(G, \prec)$ is equal to the stretch-width of its adjacency matrix.

The following is the counterpart of the so-called *rich divisions* [9] tailored for stretch-width. If R is a set of rows and C a set of columns of a matrix M, we denote by $R \setminus C$ the zone $R \cap (\text{columns}(M) \setminus C)$, that is the submatrix formed by R deprived of the columns of C (and symmetrically for $C \setminus R$).

In a division $(\mathcal{R} = (R_1, \ldots, R_n), \mathcal{C} = (C_1, \ldots, C_m))$, a row part R_i is k-wide if for every k consecutive columns parts $C_j, \ldots, C_{j+k-1}$ containing the symmetric of R_i, $R_i \setminus \cup_{j \leqslant h \leqslant j+k-1} C_h$ contains at least k distinct rows. The *k-wideness* of column parts is defined symmetrically.

A division $(\mathcal{R}, \mathcal{C})$ is *k-wide* if all its row and column parts are k-wide. The division is *k-diagonal* if none of the row and column parts is k-wide. Given a set of rows (or columns) X of a matrix M, we keep the notation $\text{conv}(X)$ for the set of rows (or columns) of M with indices between the minimum and the maximum indices of X.

▶ **Theorem 12.** *For every integer k, if $stw(M) \leqslant k$, then M has no $9k$-wide division.*

Proof. Let $\mathcal{D} = (\mathcal{R}, \mathcal{C})$ be a (symmetric) division of M. Let $(\mathcal{R}'_1, \mathcal{C}'_1)$, $(\mathcal{R}'_2, \mathcal{C}'_2), \ldots$ be a (symmetric) partition sequence of M with stretch value at most k (i.e., witnessing that the stretch-width of M is at most k). Let s be the smallest integer for which there is a row part $R' \in \mathcal{R}'_s$ such that $\text{conv}(R')$ contains a row part $R \in \mathcal{R}$ of the division $\mathcal{D}$ (by symmetry, it happens at the same moment for a column part). We will prove that R is not $9k$-wide.

Let C' be the symmetric of R' in columns. Set $\mathcal{S} := \{T \in \mathcal{R}'_s \mid \text{conv}(T) \cap R \neq \emptyset\}$. Note that $\mathcal{S}$ is the set of row parts of $\mathcal{R}'_s$ that conflicts with R, and that R' is necessarily in $\mathcal{S}$. As $\text{conv}(R) \subset \text{conv}(R')$, every part in $\mathcal{S}$ conflicts with R' and it should hold that $|\mathcal{S}| \leqslant k$

because $(\mathcal{R}'_s, \mathcal{C}'_s)$ witnesses that $\text{stw}(M) \leqslant k$. For each T in $\mathcal{S}$, we define $C_T := \{c \in \text{columns}(M) \mid c \in C,\ C \in \mathcal{C}'_s$, and $C \cap T$ is non-constant or C is the symmetric of $T\}$. By assumption on the stretch value of $(\mathcal{R}'_s, \mathcal{C}'_s)$, we know that for each $T \in \mathcal{S}$, C_T conflicts with at most k parts of $\mathcal{C}'_s$. Let C' be the symmetric of R'. As T conflicts with R', the symmetric of T conflicts with C'. The symmetric of T being contained in C_T, C_T conflicts with C' which means that $\text{conv}(C') \cap \text{conv}(C_T)$ is not empty.

Let us consider $\bigcup_{T\in\mathcal{S}} C_T$. An element A of $\mathcal{C}'_s$ conflicts with $\bigcup_{T\in\mathcal{S}} C_T$ if and only if $\text{conv}(A) \cap \text{conv}(\bigcup_{T\in\mathcal{S}} C_T)$ is non-empty. As for each T of $\mathcal{S}$, $\text{conv}(C_T) \cap \text{conv}(C')$ is non-empty, there is T_1, T_2 in $\mathcal{S}$ such that the associated two parts C_{T_1} and C_{T_2} (informally the "leftmost" and the "rightmost") verify

$$\text{conv}(C_{T_1}) \cup \text{conv}(C_{T_2}) \cup \text{conv}(C') = \text{conv}(\bigcup_{T\in\mathcal{S}} C_T).$$

Note that C_{T_i} can be equal to C'. As C_{T_1} and C_{T_2} conflicts with C', $C' \cup C_{T_1} \cup C_{T_2}$ conflicts with at most $3k$ parts of $\mathcal{C}'_s$. Thus $\bigcup_{T\in\mathcal{S}} C_T$ conflicts with at most $3k$ parts of $\mathcal{C}'_s$.

Observe that, except for C', every part in $\mathcal{C}'_s$ is covered by the union of two consecutive parts of $\mathcal{C}$. Part C' itself is covered by the union of three consecutive parts of $\mathcal{C}$: $\text{conv}(C')$ cannot cover two parts of $\mathcal{C}$ by minimality of s. Thus, overall, each part of $\mathcal{C}'_s$ is covered by the union of at most three consecutive parts of $\mathcal{C}$. Hence, if $\bigcup_{T\in\mathcal{S}} C_T$ conflicts with $3k$ parts of $\mathcal{C}'_s$, it is contained in $9k$ consecutive parts of $\mathcal{C}$, say $C_j, \ldots, C_{j+9k-1}$. Thus for any $T \in \mathcal{S}$, $T \setminus (C_j, \ldots, C_{j+9k-1})$ is constant, and so $R \setminus (C_j, \ldots, C_{j+9k-1})$ contains at most k different rows. ◀

▶ **Theorem 13** (⋆). *For every integer k, if M does not have a k-wide division, then M admits a sequence $(\mathcal{R}_1, \mathcal{C}_1), (\mathcal{R}_2, \mathcal{C}_2), \ldots$ every division of which is $2(k+1)$-diagonal.*

▶ **Theorem 14** (⋆). *If M admits a sequence of k-diagonal divisions, then stw$(M) \leqslant 4k^3$.*

▶ **Theorem 15.** *If a matrix M does not admit a k-wide division, then stw$(M) \leqslant 32(k+1)^3$.*

Proof. In fact, M admits a sequence of $2k$-diagonal divisions by Theorem 13. Applying Theorem 14 on this sequence outputs a witness of stretch-width $4\cdot(2(k+1))^3 = 32(k+1)^3$. ◀

▶ **Theorem 4** (⋆). *Given an integer k and a symmetric $n \times n$ $0,1$-matrix M, there is an $n^{O(1)}$-time algorithm that outputs a sequence witnessing that stw$(M) = O(k^3)$ or correctly reports that stw$(M) > k$.*

5 Overlap graph

Consider an ordered graph $(G, \prec)$, and think of $\prec$ as a *left-to-right* order (with the smallest vertex being the leftmost one). For any edge $e \in E(G)$, we denote by $L(e)$ (resp. $R(e)$) the left (resp. right) endpoint of e. Given two edges $e, f \in E(G)$, we say that e is *left of* f if $L(e) \preccurlyeq L(f)$, and e is *strictly left of* f if $L(e) \prec L(f)$. By extension, we say that $X \subset E(G)$ is *left of* (resp. *strictly left of*) $Y \subset E(G)$ if for every $e \in X$ and $f \in Y$, $L(e) \preccurlyeq L(f)$ (resp. $L(e) \prec L(f)$). If u, v are vertices of $(G, \prec)$, we denote by $[u, v]$ the set of vertices that are, in $\prec$, at least u and at most v. We also denote by $[\leftarrow, u]$ (resp. $[u, \rightarrow]$) the set of vertices that are at most u (resp. at least u).

We say that two edges e, f are *crossing* if $L(e) \prec L(f) \prec R(e) \prec R(f)$ (or symmetrically) and we denote $e \times f$ this relation. Observe that two edges sharing an endpoint are not crossing. The relation $\times$ is symmetric and anti-reflexive, hence defines an undirected graph on $E(G)$. We denote by $\text{Ov}(G, \prec)$ the graph $(E(G), \times)$. $\text{Ov}(G, \prec)$ is called the *overlap graph* of $(G, \prec)$; see Figure 2.

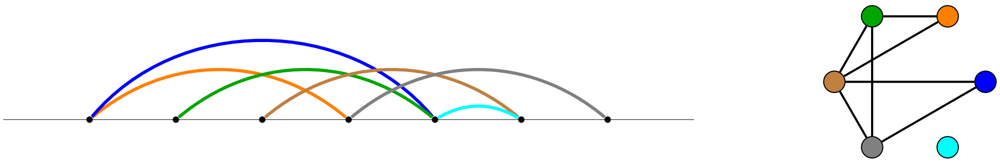

Figure 2 An ordered graph (left) and its overlap graph (right).

We relate the structure of $\mathrm{Ov}_{\prec}(G)$ and the stretch-width of G among bounded-degree graphs, by proving the following theorem:

▶ **Theorem 16.** *A class $\mathcal{C}$ of ordered graphs of bounded degree has bounded stretch-width if and only if $\{Ov(G,\prec) \mid G \in \mathcal{C}\}$ does not admit $K_{t,t}$ subgraph, for some integer t.*

The next two lemmas prove the forward implication, by considering a special point in the partition sequence. The last lemma of this section proves the backward implication, using the matrix characterization of Section 4. We say that a $K_{t,t}$ subgraph of $\mathrm{Ov}(G,\prec)$ is *clean* if the sides of the $K_{t,t}$ are $X, Y \subset E(G)$ such that X is strictly left of Y.

▶ **Lemma 17.** *For every ordered graph $(G,\prec)$, if $Ov(G,\prec)$ contains a $K_{t,t}$ as a subgraph, then $Ov(G,\prec)$ contains a clean $K_{\lfloor t/2\rfloor,\lfloor t/2\rfloor}$ subgraph.*

Proof. Assuming that $\mathrm{Ov}(G,\prec)$ has a $K_{t,t}$ subgraph, there is two disjoint sets $X, Y \subset E(G)$ each of size t such that for every $x \in X$ and $y \in Y$, $x \times y$. Let $L(x_1) \preccurlyeq L(x_2) \preccurlyeq \ldots \preccurlyeq L(x_t)$ be the elements of X, and $L(y_1) \preccurlyeq L(y_2) \preccurlyeq \ldots \preccurlyeq L(y_t)$, the elements of Y. As $x_{\lfloor t/2\rfloor}$ and $y_{\lfloor t/2\rfloor}$ are crossing, either $L(x_{\lfloor t/2\rfloor}) \prec L(y_{\lfloor t/2\rfloor})$ or $L(y_{\lfloor t/2\rfloor}) \prec L(x_{\lfloor t/2\rfloor})$. The sides of the clean $K_{\lfloor t/2\rfloor,\lfloor t/2\rfloor}$ are $\{x_1,\ldots,x_{\lfloor t/2\rfloor}\}$ and $\{y_{\lfloor t/2\rfloor},\ldots,y_t\}$ in the former case, and $\{y_1,\ldots,y_{\lfloor t/2\rfloor}\}$ and $\{x_{\lfloor t/2\rfloor},\ldots,x_t\}$ in the latter. ◀

▶ **Lemma 18** ($\star$). *For any ordered graph $(G,\prec)$, if $\Delta(G) \leqslant d$ and $stw(G,\prec) \leqslant t$, then $Ov(G,\prec)$ does not contain $K_{N,N}$ with $N = 4td^2$ as a subgraph.*

▶ **Lemma 7.** *For every ordered graph $(G,\prec)$ and positive integer N, if $Ov(G,\prec)$ does not contain $K_{N,N}$ as a subgraph, then $stw(G,\prec) \leqslant 32(2N+1)^3$.*

Proof. Let $(G,\prec)$ be an ordered graph such that $\mathrm{Ov}(G,\prec)$ does not contain $K_{N,N}$ as a subgraph, and let M be the adjacency matrix of $(G,\prec)$. We prove that $\mathrm{stw}(M) \leqslant 32(2N+1)^3$.

Suppose, for the sake of contradiction, that $\mathrm{stw}(M) > 32(2N+1)^3$. By Theorem 15, there is a $2N$-wide division $(\mathcal{R} = \{R_1,\ldots,R_k\}, \mathcal{C} = \{C_1,\ldots,C_k\})$ of M. In particular, for any row R_i, $R_i \setminus C_{i-N+1},\ldots,C_{i+N-1}$ contains more that $2N$ different rows. Let D be the union of the zones $R_i \cap C_j$ such that $|i-j| < N$, that is, the $2N-1$ "longest" diagonals of zones of the division $(\mathcal{R},\mathcal{C})$. As, for every $i \in [k]$, the number of distinct rows in $R_i \setminus D$ (resp. distinct columns in $C_i \setminus D$) is at least $2N$, $R_i \setminus D$ (resp. $C_i \setminus D$) contains at least $2N$ 1-entries.

To simplify the coming notations, let denote by $\|M'\|$ the number of 1-entries of any submatrix M' of M. For example, $\|R_i \setminus D\| \geqslant 2N$. Observe that R_i (resp. C_j) is split by D in at most two sets $R_i^{\leftarrow}$ and $R_i^{\rightarrow}$ (resp. $C_j^{\uparrow}$ and $C_j^{\downarrow}$), namely, $R_i^{\leftarrow} = \bigcup_{j \leqslant i-N} R_i \cap C_j$ and $R_i^{\leftarrow} = \bigcup_{j \geqslant i+N} R_i \cap C_j$.

Observe that for every i, j such that $i+1 \leqslant j < i+N$, each 1-entry of $R_i^{\rightarrow}$ (resp. $C_i^{\downarrow}$) and 1-entry of $C_j^{\uparrow}$ (resp. $R_j^{\rightarrow}$) correspond to crossing edges in $(G,\prec)$. As $\mathrm{Ov}(G,\prec)$ does not contain any $K_{N,N}$ subgraph we have, for every i, j such that $i+1 \leqslant j < i+N$:

1. $\min(\|R_i^{\to}\|, \|C_j^{\uparrow}\|) < N$, and
2. $\min(\|C_i^{\downarrow}\|, \|R_j^{\leftarrow}\|) < N$.

Indeed, if the first item does not hold, N 1-entries in $R_i^{\to}$ and N 1-entries in $C_j^{\uparrow}$ form the two sides of a $K_{N,N}$.

We finally prove by induction on i that, while $2i \leqslant k$, the property $\|R_{2i}^{\to}\| > N$, henceforth called $(\mathcal{Q}_i)$, holds. Note that $R_0^{\leftarrow}$ is empty. Thus $\|R_0^{\to}\| \geqslant 2N > N$, hence $(\mathcal{Q}_0)$ holds. Now assume that $(\mathcal{Q}_i)$ holds. By the first item, we have $\|C_{2i+1}^{\uparrow}\| < N$. Thus $\|C_{2i+1}^{\downarrow}\| > N$, since

$$C_{2i+1} \setminus D = C_{2i+1}^{\downarrow} \cup C_{2i+1}^{\uparrow} \text{ and } \|C_{2i+1} \setminus D\| \geqslant 2N.$$

Symmetrically, by the second item, $\|R_{2i+2}^{\leftarrow}\| < N$, and hence $\|R_{2i+2}^{\to}\| > N$. Thus $(\mathcal{Q}_{i+1})$ holds. As $R_{k-N+1}^{\to}$ is empty, $(\mathcal{Q}_i)$ can no longer be true when $2i \geqslant k - N + 1$, a contradiction. Therefore $\mathrm{stw}(M) \leqslant 32(2N)^3$. ◀

6 Subdivisions

When subdividing the edges of an ordered graph, there is a simple way of updating its vertex ordering without creating larger bicliques in its overlap graph.

▶ **Lemma 19.** *Let $(G, \prec)$ be an ordered graph, and H be obtained by subdividing an edge of G. There is an order $\prec'$ such that, for every integer t, if $Ov(G, \prec)$ has no $K_{t,t}$ subgraph, then $Ov(H, \prec')$ has no $K_{t,t}$ subgraph.*

Proof. Let $e = uv$ be the edge of G subdivided to form H, and let $w \in V(H)$ be the new vertex resulting from this subdivision. The total order $\prec'$ is obtained from $\prec$, by adding w next to u, say, just to its right. This way $\mathrm{Ov}(H, \prec')$ is simply $\mathrm{Ov}(G, \prec)$ plus an isolated vertex. Indeed the edge $uw \in E(H)$ is an isolated vertex in $\mathrm{Ov}(H, \prec')$, since u and w are consecutive along $\prec'$, whereas $wv \in E(H)$ crosses the same edges as uv was crossing. ◀

We now define a long subdivision process that is actually "erasing" large bicliques in the overlap graph of a bounded-degree graph. Let uv be an edge of an ordered graph $(G, \prec)$, with h vertices between u and v, say, $u \prec u_1 \prec u_2 \prec \ldots \prec u_h \prec v$. We describe an $h+1$-subdivision of uv in $(G, \prec)$ that we call *flattening of uv*. We delete uv, and create $h+1$ new vertices $w_1, \ldots, w_{h+1}$ such that $u \prec w_1 \prec u_1 \prec w_2 \prec u_2 \prec \ldots \prec w_h \prec u_h \prec w_{h+1} \prec v$. We then create the edges uw_1, w_iw_{i+1} for every $i \in [h]$, and $w_{h+1}v$. We may say that these *edges stem* from uv. An iterated subdivision of $(G, \prec)$ chooses a total order on the edges of G, and iteratively flattens the edges of G in this order (note that the created edges are *not* flattened themselves); see Figure 3.

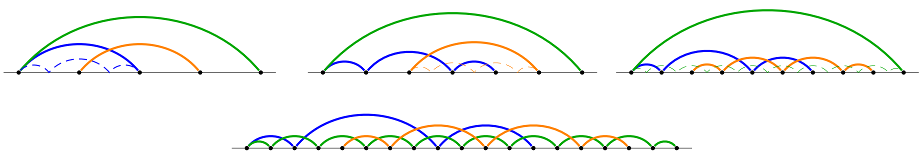

Figure 3 An iterated subdivision. Created edges have the color of the edge they stem from.

▶ **Lemma 20.** *Any iterated subdivision $(G', \prec')$ of an ordered graph $(G, \prec)$ of maximum degree d, is such that $Ov(G', \prec')$ has no $K_{2d+2,2d+2}$ subgraph.*

Proof. Assume for the sake of contradiction that $\mathrm{Ov}(G', \prec')$ has a $K_{2d+2,2d+2}$ subgraph. Then by Lemma 17, $\mathrm{Ov}(G', \prec')$ has a clean $K_{d+1,d+1}$ subgraph. Let X, Y be the two sides of this clean biclique, where X is left of Y. As every vertex of G' (like G) is incident to at most d edges, there is $\{x_1, x_2\} \subseteq X$ and $\{y_1, y_2\} \subseteq Y$ such that $L(x_1) \preccurlyeq L(x_2) \prec L(y_1) \prec L(y_2) \prec R(x_i) \prec R(x_{3-i}) \prec R(y_j) \preccurlyeq R(y_{3-j})$ with $i, j \in [2]$.

As x_1 and x_2 cross y_1 and y_2, there is no $i, j \in [2]$ such that x_i and y_j stem from the same edge of G. We can thus assume without loss of generality that the last edge among x_1, x_2, y_1, y_2 to be created is in X (since the argument is symmetric if this happens in Y), i.e., x_i for some $i \in [2]$. When x_i is created, the vertices $L(y_1)$ and $L(y_2)$ already exist and form a non-trivial interval since $L(y_1) \prec L(y_2)$. This contradicts the construction of the iterated subdivision, since x_i jumps over $[L(y_1), L(y_2)]$, when it should have at least created an intermediate vertex in $[L(y_1), L(y_2)]$. ◀

▶ **Theorem 6.** *Every $(\geqslant n2^m)$-subdivision of every n-vertex m-edge graph G of maximum degree d has stretch-width at most $32(4d + 5)^3$.*

Proof. Let G be any graph of $\mathcal{C}$ with n vertices and m edges, and let G'' be any $(\geqslant n2^m)$-subdivision of G. Choose an arbitrary order $\prec$ of $V(G)$. Let $(G', \prec')$ be the iterated subdivision of $(G, \prec)$, choosing an arbitrary order on the edges G. By Lemma 20, $\mathrm{Ov}(G', \prec')$ has no $K_{2d+2,2d+2}$ subgraph. Every edge of G is subdivided at most $n2^m$ times by the process of iterated subdivision. By Lemma 19, the edges of G' can be further subdivided to obtain G'' such that $\mathrm{Ov}(G'', \prec'')$ has no $K_{2d+2,2d+2}$ subgraph, for some vertex ordering $\prec''$. Therefore, by Section 5, $\mathrm{stw}(G'', \prec'') \leqslant 32(4d + 5)^3$, and in particular, $\mathrm{stw}(G'') \leqslant 32(4d + 5)^3$. ◀

▶ **Corollary 21.** *There are graph classes with bounded stretch-width and maximum degree, and yet unbounded treewidth.*

Proof. Consider the family $\Gamma_1, \Gamma_2, \ldots$, where Γ_k is the $k^2 2^{2k(k-1)}$-subdivision of the $k \times k$-grid, for every positive integer k. The graphs from this family have degree at most 4, and stretch-width at most 296352, but unbounded treewidth since $\mathrm{tw}(\Gamma_k) = k$. ◀

The above argument gives an example of n-vertex graphs with bounded degree and stretch-width, and treewidth $\Omega(\sqrt{\log n})$. We can do better by picking the vertex ordering $\prec$, and the order on the edges (for the iterated subdivision) more carefully. We simply order

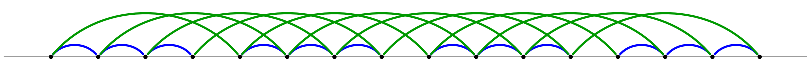

Figure 4 The 4×4 grid ordered row by row, with the horizontal edges in blue, and vertical edges in green.

the grid row by row, and from left to right within each row; see Figure 4. We perform the iterated subdivision of this ordered grid, with the following edge ordering. First we flatten every horizontal edge (in blue), in any order. When this is done, the total number of vertices has less than doubled. Then we flatten every vertical edge (in green) from left to right. It can be observed that, starting from the $k \times k$ grid, we now obtain an iterated subdivision with less than 2^{ck} vertices, for some constant c. Thus, there are n-vertex graphs with bounded degree and stretch-width, and treewidth $\Omega(\log n)$.

7 Classes with bounded Δ + stw have logarithmic treewidth

For any edge e of an ordered graph $(G, \prec)$, we denote $e^i =]L(e), R(e)[$, the *interior of* e, and $e^o := [\leftarrow, L(e)[\cup]R(e), \rightarrow]$, the *exterior of* e; note that $L(e)$ and $R(e)$ are neither part of e^i nor of e^o. The *length* of e according to $\prec$ is $\ell(e, \prec) = R(e) - L(e)$. When F is a set of edges we define $\ell(F, \prec)$ to be the maximum length of an edge of F. We say that a set C of vertices is a *c-balanced separator of* G when there is a c-balanced separation (A, B) of G such that $C = A \cap B$. In an ordered graph $(G, \prec)$ a set of vertices C is a *left/right c-balanced separator* if there exists a c-balanced separation (A, B) where A contains the initial interval of length $c \cdot n$, B contains the final interval of length $c \cdot n$ and $C = A \cap B$.

To simplify the notations, if the vertices of $(G, \prec)$ are $v_1 \prec \cdots \prec v_n$, we will write $G\langle i, j\rangle$ instead of $G[[v_i, v_j]]$ and $\langle i, j\rangle$ instead of $[v_i, v_j]$.

▶ **Lemma 22** ($\star$). *For any ordered n-vertex graph $(G, \prec)$, if $\Delta(Ov(G, \prec)) \leqslant d$ and $\ell(E(G), \prec) \leqslant \lambda n$, then G admits a left/right $(1/2 - \lambda)$-balanced separator of size at most $d + 2$.*

▶ **Lemma 23** ($\star$). *For every ordered graph $(G, \prec)$, if $\Delta(Ov(G, \prec)) \leqslant d$ then G admits a $1/6$-balanced separator of size $2d + 4$.*

We say that a set S of edges of $(G, \prec)$ is a *rainbow* if for every pair e, f of S, $e^i \subset f^i$ or $f^i \subset e^i$. Notice that a rainbow induces an independent set in $\mathrm{Ov}(G, \prec)$. When S contains t edges we say that S is a *t-rainbow*, or a rainbow of order t. The following is an application of Dilworth's theorem on permutation graphs. It can be found in [27].

▶ **Lemma 24** ([27]). *Let $(G, \prec)$ be an ordered graph, such that $Ov(G, \prec)$ does not contain some K_t. Then, for every vertex v of $V(G)$, for every set F of edges from $[\leftarrow, v[$ to $]v, \rightarrow]$ we have $|F| \leqslant kt$ where k is the maximum order of a rainbow of F.*

For any rainbow S, we denote by $S^i := \bigcup_{e \in S} e^i$. A *rainbow over v* is a rainbow S contained in the set of edges from $[\leftarrow, v[$ to $]v, \rightarrow]$. A *maximum rainbow over v* is a rainbow of maximum cardinality among the rainbows over v.

▶ **Lemma 25** ($\star$). *Let $(G, \prec)$ be an ordered graph such that $Ov(G, \prec)$ does not contain a $K_{t,t}$ subgraph. Then, if S is a maximum rainbow over $v \in V(G)$, there is a vertex $x \in S^i$ and a set U that separates $[\leftarrow, x[$ from $]x, \rightarrow]$ with U of size $\leqslant g(t)(\log(\ell(S, \prec)) + 1)$, for a function g such that $g(t) = O(t^4)$.*

▶ **Theorem 26.** *For any ordered graph $(G, \prec)$, if $Ov(G, \prec)$ does not contain any $K_{t,t}$, then G contains a 1/12-balanced separator of order at most γt^4, for some constant γ.*

Proof. Let $(G, \prec)$ be an ordered graph on n vertices, such that $\mathrm{Ov}(G, \prec)$ does not contain any $K_{t,t}$. Let F be the set of edges over a vertex $v \in V(G)$, and let S be a maximum rainbow over v. Denote by $e_1, \ldots, e_k$ the edges in S with $e^i_{j+1} \subsetneq e^i_j$. We build $X_{a,b}$ and Y_a in the same way as in the proof of Lemma 25.

Suppose that there are $3t + 2$ edges of S with length in $[n/12, 11n/12]$. We denote these edges by $e_i, \ldots, e_j$, with $j = i + 3t + 2$. We consider the set of edges Z, which is the union of the Y_a and $X_{a,b}$ for $a \in [i, j]$ and $b \in [a - t, a + t]$, and erase it by removing $L_Z = \{L(z) \mid z \in Z\}$. As in the proof of Lemma 25, this operation removes $O(t^4)$ vertices of G. Let $G' = G - L_Z$ be the obtained graph.

Let x, y two vertices, respectively in $]L(e_{i+t}), L(e_{i+2t})[$ an $]R(e_{i+2t}), R(e_{i+t})[$. If one of these sets is empty, say $]L(e_{i+t}), L(e_{i+2t})[$, then any path going from $[\leftarrow, L(e_{i+t})[\cup]R(e_{i+t}), \rightarrow]$ to $[L(e_{i+2t}), R(e_{i+2t})]$ has a vertex in $]R(e_{i+2t}), R(e_{i+t})[$. Then, we only need to consider y in the following (if y also does not exist, the graph is already separated).

By construction, any edge over x is contained in $[L(e_i), L(e_j)]$, or is going from $[\leftarrow, L(e_i)]$ to $[R(e_i), \rightarrow]$. Thus, by applying Lemma 25 on $G_x = G'[\leftarrow, R(e_i)]$ over x, we find a set U_x separating $[\leftarrow, u_x]$ from $[u_x, \rightarrow]$, where u_x is in $[L(e_i), L(e_j)]$. We do the same on the right side, considering $G_y = G[L(e_i), \rightarrow]$ and finding U_y and u_y such that $u_y \in [R(e_j), R(e_i)]$. Set $U_y \cup U_y$ separates $]u_x, u_y[$ from $]u_x, u_y]^c$. As $u_y - u_x$ is between $n/11$ and $n/12$, we found a $n/12$-balanced separator of G of size at most $1 + 2g(t) \cdot \log \ell(S, \prec)$.

Hence, assume now that for any vertex v of G, the number of edges of length in $[n/12, 11n/12]$ in a maximum rainbow over v is bounded by $3t + 1$. Thus, for every v, the set M_v of edges over v going from $[n/12, v - n/12]$ to $[v + n/12, 11n/12]$ is of size at most $2t(3t + 1)$ by Lemma 24.

Set $u = v_{n/3}$ and $v = v_{2n/3}$. By deleting the set $A = \{L(e) \mid e \in M_u \cup M_v\}$, we get a new ordered graph $H = (G - A, \prec)$. Consider $H_u = H[\leftarrow, 11n/12]$, and $H_v = H[n/12, \rightarrow]$. Then the length of a maximum rainbow over u (resp. v) in H_u (resp. H_v) is at most $n/12$.

Hence we can apply Lemma 25 on H_u over u, and on H_v over v. This yields two sets W_u, W_v of size $O(t^4 \log n)$ and two vertices w_u, w_v such that $|w_u - u|$ (resp. $|w_v - v|$) is at most $n/12$, and such that W_u separates $[\leftarrow, w_u]$ from $[w_u, \rightarrow]$ in H_u (and resp. for v). The set $W_u \cup W_v$ is then separating $]w_u, w_v[$ from $[w_u, w_v]^c$. ◀

▶ **Theorem 27.** *Let G be any graph on n vertices, such that $\Delta(G) \leqslant d$ and $stw(G) \leqslant t$. Then G contains a $1/12$-balanced separator of size at most $\gamma(d^2 t)^4 \log n$, for a constant γ.*

Proof. Let G a graph on n vertices, such that $\text{stw}(G) \leqslant t$ and $\Delta(G) \leqslant d$. Let $\prec$ be an order such that $(G, \prec)$ has stretch-width t. By Lemma 18, $\text{Ov}(G, \prec)$ does not admit any $K_{4td^2,4td^2}$ as a subgraph. Hence the Theorem 26 ensures the existence of a $1/12$-balanced separator of $(G, \prec)$ (hence of G) of size $\gamma(4td^2)^4 \log n$. ◀

By Lemma 11 and Theorem 27, we obtain the bound on the treewidth of a graph of bounded degree and bounded stretch-width.

▶ **Theorem 5.** *There is a c such that for every graph G, $tw(G) \leqslant c\Delta(G)^8 stw(G)^4 \log |V(G)|$.*

References

1 Tara Abrishami, Maria Chudnovsky, Sepehr Hajebi, and Sophie Spirkl. Induced subgraphs and tree-decompositions III. Three-path-configurations and logarithmic tree-width. *Advances in Combinatorics*, 2022.

2 Ambroise Baril, Miguel Couceiro, and Victor Lagerkvist. Linear bounds between cliquewidth and component twin-width and applications, 2023. URL: `https://ramics20.lis-lab.fr/slides/slidesAmbroise.pdf`.

3 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. `doi:10.4230/LIPIcs.ICALP.2022.18`.

4 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. `doi:10.4230/LIPIcs.STACS.2023.10`.

5 Benjamin Bergougnoux, Jan Dreier, and Lars Jaffke. A logic-based algorithmic meta-theorem for mim-width. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3282–3304. SIAM, 2023. doi:10.1137/1.9781611977554.ch125.

6 Marthe Bonamy, Edouard Bonnet, Hugues Déprés, Louis Esperet, Colin Geniet, Claire Hilaire, Stéphan Thomassé, and Alexandra Wesolek. Sparse graphs with bounded induced cycle packing number have logarithmic treewidth. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3006–3028. SIAM, 2023. doi:10.1137/1.9781611977554.ch116.

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

8 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. *Combinatorial Theory*, 2(2), 2022. doi:10.5070/C62257876.

9 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. doi:10.1145/3519935.3520037.

10 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.

11 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. doi:10.1145/3486655.

12 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. arXiv:2202.11858.

13 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

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

15 Zdenek Dvorák and Sergey Norin. Treewidth of graphs with balanced separations. *J. Comb. Theory, Ser. B*, 137:137–144, 2019. doi:10.1016/j.jctb.2018.12.007.

16 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

17 Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Model-checking for first-order logic with disjoint paths predicates in proper minor-closed graph classes. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3684–3699. SIAM, 2023. doi:10.1137/1.9781611977554.ch141.

18 Frank Gurski and Egon Wanke. The tree-width of clique-width bounded graphs without $K_{n,\ n}$. In Ulrik Brandes and Dorothea Wagner, editors, *Graph-Theoretic Concepts in Computer Science, 26th International Workshop, WG 2000, Konstanz, Germany, June 15-17, 2000, Proceedings*, volume 1928 of *Lecture Notes in Computer Science*, pages 196–205. Springer, 2000. doi:10.1007/3-540-40064-8_19.

19 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

20 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

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

22 Michal Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In Filip Murlak and Piotr Sankowski, editors, *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, volume 6907 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2011. doi:10.1007/978-3-642-22993-0_47.

23 Svatopluk Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15(2):307–309, 1974.

24 Nicole Schirrmacher, Sebastian Siebertz, Giannos Stamoulis, Dimitrios M. Thilikos, and Alexandre Vigny. Model checking disjoint-paths logic on topological-minor-free graph classes. *CoRR*, abs/2302.07033, 2023. doi:10.48550/arXiv.2302.07033.

25 Ni Luh Dewi Sintiari and Nicolas Trotignon. (theta, triangle)-free and (even hole, k_4)-free graphs - part 1: Layered wheels. *J. Graph Theory*, 97(4):475–509, 2021. doi:10.1002/jgt.22666.

26 Szymon Toruńczyk. Flip-width: Cops and robber on dense graphs. *CoRR*, abs/2302.00352, 2023. doi:10.48550/arXiv.2302.00352.

27 Jakub Černý. Coloring circle graphs. *Electronic Notes in Discrete Mathematics*, 29:457–461, 2007. European Conference on Combinatorics, Graph Theory and Applications. doi:10.1016/j.endm.2007.07.072.