Twin-width: theory and applications

UPDATE: This proposal was written one year ago and is now (February 2022) partially updated. It will not be updated regularly. For the status of the open questions, check http://perso.ens-lyon.fr/edouard.bonnet/openQuestions.html **instead.**

Summary table of persons involved in the project:

Partner	Name	First name	Current Position	Role and responsibilities	Invol-vement
LIP, MC2	BONNET	Édouard	CR	Coordinator	36p·month
LIP, MC2	CHAKRABORTY	Dibyayan	postdoc (É. B.)		12p·month
LIP, MC2	DÉPRÉS	Hugues	PhD student (É. B.)		32p·month
LIP, MC2	GENIET	Colin	PhD student (S. T.)		32p·month
LIP, MC2	THOMASSÉ	Stéphan	PR		24p·month
LIP, MC2	WATRIGANT	Rémi	MCF		18p·month
LIP, MC2	?	?	1 or 2 postdocs		24p·month
LAMSADE	KIM	Eunjung	CR		7p·month
G-SCOP	ESPERET	Louis	CR		10p·month
G-SCOP	GIOCANTI	Ugo	PhD student (L. E. & S. T.)		14p·month

1 Context, positioning, and objectives

In the past two years, we have introduced and developed the theory around a novel graph-theoretic invariant, dubbed twin-width. This notion has turned out useful and fruitful in several areas of research including algorithmic graph theory, combinatorics, model theory, and algebra. An interesting aspect is that classes of bounded twin-width, while specifically structured, are mostly orthogonal to the current organization of graph theory. If we have at least partially explored some of the new lands, many more have appeared to us and are currently globally uncharted. The time seems right to try and expand our group with postdoctoral researchers and interested colleagues, who would bring some fresh lights and additional expertises into the project.

We start with an introduction to twin-width, since it is not a very standard notion yet.

1.1 A tour through twin-width

Cographs can be defined inductively by: The 1-vertex graph is a cograph, and if G_1 and G_2 are two cographs, then the disjoint union $G_1 \cup G_2$ and the complete join $G_1 + G_2$ are also cographs. This definition gives rise to a recursive linear-time algorithm solving many NP-hard problems on this particular class (see Fig. 1, left). The inception of clique-width [15, 48] in the early nineties, and of rank-width [43] a decade later, may both be thought of as generalizing this definition, the former, by allowing local joins, the latter, by allowing more complicated attachments. In the latter three papers, the authors show or observe that many problems can be solved faster when the clique-width/rank-width is bounded, and present cographs as a paradigmatic example of constant clique-width/rank-width.

Among the several characterizations of cographs, another one goes as follows: One can iteratively find two twins (i.e., two vertices with the same neighbors, outside of themselves) and contract them into one vertex, until the graph contains a single vertex. There is a perhaps more contrived algorithmic scheme to solve problems efficiently based on this equivalent definition (see Fig. 1, right). Yet, trying to generalize this alternative scheme, one could have come up with the notion of twin-width thirty years ago.

Now we allow to contract, or identify, two (possibly non-adjacent) vertices with a small number of private neighbors. We mark in red those edges that are incident to only one of the two contracted

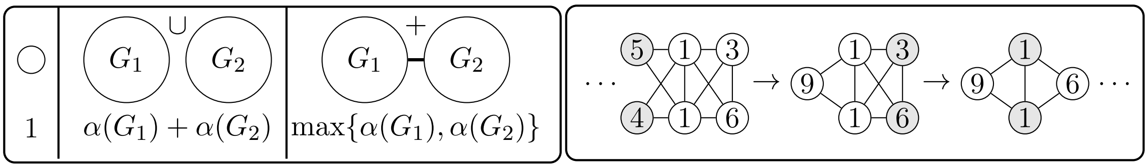

Figure 1: Two ways of solving Maximum Independent Set, that is, the size $\alpha(G)$ of a largest subset of pairwise non-adjacent vertices, on cographs. To the left, the classic non-tail recursive argument. To the right, the same idea made iterative: When the next two twins to contract are non-adjacent, we record their sum, when they are adjacent, their maximum. Initially all the vertices contain value 1, and eventually the single vertex will contain $\alpha(G)$.

vertices (see Fig. 2). This means that we work with trigraphs, where between a pair of vertices, we have either a non-edge, or a black edge, or a red edge.

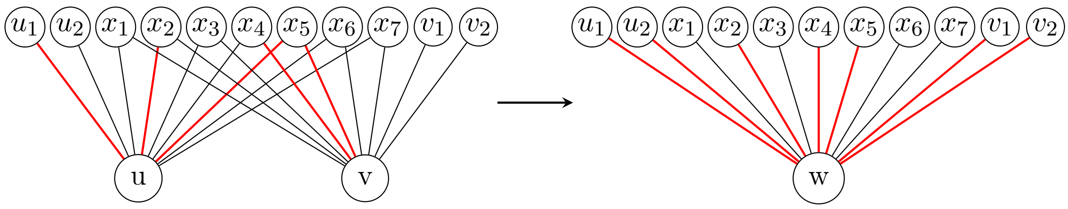

Figure 2: Contraction of vertices u and v into w, and how the edges of the trigraph are updated.

Red edges are "error edges", for which the adjacency is uncertain, and should be avoided as much as possible. The twin-width of an n-vertex graph G is the least integer d, such that G can be reduced to a single vertex by a sequence of contractions, where every trigraph of the sequence has maximum red degree (that is, maximum number of red edges incident to a vertex) at most d. The sequence of trigraphs, usually denoted $G = G_n, G_{n-1}, \ldots, G_1$, is then called a *d-sequence*.

Our path to the definition has been different. About eight years ago, Sylvain Guillemot and Dániel Marx wrote a beautiful paper [30] solving the Permutation Pattern problem. Given two permutations σ and π, Permutation Pattern asks whether the matrix of σ is a submatrix of the matrix of π. Guillemot and Marx found an algorithm with running time $2^{O(|\sigma|^2 \log |\sigma|)}|\pi|$ (now even $2^{O(|\sigma|^2)}|\pi|$), that is, linear if σ is fixed. The algorithm relies on a celebrated result in combinatorics, solution to the Füredi-Hajnal and Stanley-Wilf conjectures [35], the Marcus-Tardos theorem [38].

Theorem 1 ([38])**.** *For every integer k, there is a c_k such that every $n \times m$ 0,1-matrix with at least $c_k \max(n, m)$ entries 1 admits a $k \times k$ division with no cell being all 0.*

Guillemot and Marx define a width invariant for permutation patterns. They then proceed with a win-win strategy: Either the width of π is large, and the Marcus-Tardos theorem implies that σ (in fact, every $|\sigma| \times |\sigma|$ pattern) appears in π, or the width of π is small, and Permutation Pattern can be efficiently solved by dynamic programming. They conclude their introduction observing that "*[i]t 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.*" And indeed twin-width extends their very width to general graphs.

Classes with bounded twin-width turn out to be very general. They include bounded rank-, clique-, or boolean-width graphs, K_t-minor free graphs, K_t-free d-dimensional unit ball graphs, unit interval graphs, posets of bounded antichain size, some specific family of expanders, $\Omega(\log n)$-subdivisions of all the graphs [10, 4]. All these results are effective: On these classes, we can find in polynomial-time $O(1)$-sequences. A useful, and also effective, characterization of bounded twin-width, in order to establish some of these results, is the existence of a total order, said *mixed free*, on the vertex set such that the corresponding adjacency matrix does not admit large complex divisions. More precisely, a (sub)matrix is *mixed* if it has at least two distinct row vectors and at least two distinct column vectors. Then, an ordering of the vertex set of a graph G is *t-mixed free* if the adjacency matrix of G following that order does not admit a $t \times t$ division (i.e., a partition of the rows and of the columns in t intervals) where each of the t^2 cells is mixed. Using the Marcus-Tardos theorem, and following the lines of Guillemot and Marx, we showed that:

Theorem 2 ([10]). *Every graph admitting a t-mixed free order has twin-width bounded by $2^{2^{O(t)}}$.*

Despite the wide variety of classes with bounded twin-width, they allow linear-time fixed-parameter algorithms (building upon the basic scheme of Fig. 1). Equipped with an $O(1)$-sequence, one can solve in linear-time problems that are known intractable on general graphs. For instance k-Independent Set and k-Dominating Set,[1] for which any $f(k)n^{o(k)}$-time algorithm would enable an unlikely subexponential algorithm for 3-SAT, can be solved in time $2^{O(k)}n$ on graphs given with an $O(1)$-sequence [5]. More generally, still with an $O(1)$-sequence, one can model check any first-order sentence[2] in time $f(k)n$ [10]. More precisely, we designed the following algorithm.

Theorem 3 ([10]). *Given a graph G, a d-sequence $G = G_n, \ldots, G_1$, and a first-order sentence φ of quantifier depth ℓ, one can decide $G \models \varphi$ in time $f(\ell, d)n$.*

Admittedly the function f is horrendous; it is a tower of exponentials of height $O(\ell)$. Nevertheless this high dependency is unavoidable under some standard complexity-theoretic assumption, and first-order logic provides a fairly broad class of problems. Again, as we mentioned, faster algorithms exist on *specific* first-order expressible problems.

A *first-order (FO) transduction* of a graph class $\mathcal{C}$ consists of all the graphs obtainable by coloring the vertices of some $G \in \mathcal{C}$ with a constant number of colors, defining a new edge set by means of a first-order formula with two free variables (that formula may use the old edges and the colors), and taking an induced subgraph of the newly built graph (finally ignoring the colors). The following theorem shows that bounded twin-width classes are particularly robust, as far as model theory is concerned.

Theorem 4 ([10]). *Every FO transduction of a class with bounded twin-width has bounded twin-width.*

Results like Theorem 3 are known for the sparse classes introduced by Nešetřil and Ossona de Mendez [40]. Dvorák, Král, and Thomas [19] achieved a fixed-parameter linear-time algorithm for first-order model checking on classes with bounded expansion[3], while Grohe, Kreutzer, and Siebertz [28] obtained a fixed-parameter quasilinear-time algorithm on the more general (the *most* general, for *subgraph-closed* classes with such an algorithm) nowhere dense classes.[4]

These results are incomparable with Theorem 3, which also applies to dense classes (like unit interval graphs, or bounded clique-width classes), while bounded-degree graphs (a class with bounded expansion) have unbounded twin-width. The reason we know that bounded-degree graphs, even subcubic graphs, have unbounded twin-width is perhaps not the most satisfactory.

Theorem 5 ([4, 11]). *Every class of graphs of twin-width at most d contains at most $2^{O_d(n)} \cdot n!$ graphs labeled by $[n]$, and even at most $2^{O_d(n)}$ non-isomorphic n-vertex graphs.*

A class with a growth of Theorem 5 is said *small*. Since subcubic graphs have a larger labeled growth of $n^{\frac{3}{2}n+o(n)}$, they cannot have bounded twin-width. To conclude our tour, let us finally mention that twin-width explains a lot about hereditary (i.e., closed under taking induced substructures) classes of totally ordered graphs.

[1] The problems of finding, in a graph, at least k vertices that are pairwise non-adjacent, or at most k vertices whose closed neighborhood is the entire vertex set, respectively.

[2] Such as $\exists x_1 \exists x_2 \cdots \exists x_{2k-1} \exists x_{2k} \bigwedge_{1 \leqslant i \leqslant k} E(x_{2i-1}, x_{2i}) \wedge \bigwedge_{(i,j) \notin \{(1,2),(3,4),\ldots,(2k-1,2k)\}} \neg E(x_i, x_j)$ which corresponds to the k-Induced Matching, where one is asked to find a matching of k edges such that no edge of the graph has a common endpoint with two edges of the matching.

[3] A subgraph-closed class has *bounded expansion* if there is a function f such that for every integer p, no p-subdivision of a graph with minimum degree $f(p)$ is in the class.

[4] A subgraph-closed class is *nowhere dense* if there is a function f such that for every integer p, the p-subdivision of the clique $K_{f(p)}$ is not in the class.

Theorem 6 ([7]). *Let $\mathcal{C}$ be a hereditary class of ordered graphs. The following are equivalent.*

- *$\mathcal{C}$ has bounded twin-width.*
- *$\mathcal{C}$ is monadically dependent, that is, no transduction of $\mathcal{C}$ contains all graphs [1].*
- *First-order model checking is fixed-parameter tractable on $\mathcal{C}$.*[5]
- *$\mathcal{C}$ is small.*
- *$\mathcal{C}$ has less than $\sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} k! = n^{n/2+o(n)}$ graphs on n vertices.*

The gap in the growth of hereditary classes of ordered graphs, implied by the equivalence between the fourth and fifth items of Theorem 6, settled a conjecture of Balogh, Bollobás, and Morris [2]. Twin-width has been defined for unordered matrices on finite alphabets and binary structures [10], and for (ordered) matrices also on finite alphabets [7]. These two paths lead to functionally equivalent definitions for the twin-width of ordered graphs (or ordered binary structures) [7, Section 2].

1.2 Outline of the project

Our goal is to demonstrate that twin-width, although discovered relatively late, is an essential part of graph theory. We have started and wish to continue proving that, in the realm of graphs and its neighboring realms, twin-width is a key notion in understanding sporadically observed phenomena in a generic and simpler light, as well as in accessing new truths. The project splits into three work packages (WP1: foundation, WP2: theoretical applications, and WP3: practical applications).

The first package, WP1, aims to continue developing the theory of twin-width, exporting it to new settings, streamlining the notion and the central results, and making this output as accessible as it should. This package further splits into three tasks. The first task entails obtaining new results, the missing pieces that some silences during the tour may have suggested; mainly, an efficient approximate algorithm finding contraction sequences, and explicit examples of bounded-degree graphs with unbounded twin-width. The second task of WP1 is to find useful extensions of twin-width to other objects, such as matrices over infinite fields, matroids, triple systems/3-dimensional tensors, set systems/tensors, as well as useful restrictions of bounded twin-width classes, such as only allowing edge contractions, or adding to the structure a spanning tree order. The usefulness of the extensions is measured as the extent to which nice properties of (graphic) twin-width are preserved in a more general or different setting. The usefulness of the restrictions may take various forms: characterizing well-established classes with twin-width-like invariants, providing preliminary steps for which the algorithm sought by the first task of WP1 is more amenable, or obtaining still quite general bounded twin-width classes with enhanced algorithmic and structural properties. If the first task asks for new results and the second, for new "definitions," the third task is not concerned with novelty. Instead it deals with keeping the theory as accessible as possible, while cleaning the exposition and streamlining the proofs as our understanding matures. Concretely this will materialize by writing a monograph on twin-width toward the end of the project.

In the second work package, WP2, challenges are of two kinds: improved bounds/algorithms on structures of bounded twin-width, and advances on a priori *non-twin-width-related* open questions with elements of our theory. Both challenges resonate with the first work package. The former kind is connected to the aim of understanding bounded twin-width structures better, whereas the latter is more related to the essence of the theory and to the second task of WP1 on extensions and restrictions of twin-width. The applications concern finite model theory, approximation algorithms, finitely generated groups, labeling schemes, linear algebra, and computational geometry. The third and last work package, WP3, deals with practical implementation and investigates real-life scenarios where twin-width could be useful.

1.3 WP1: Foundation of twin-width

We start with the most pressing open questions on the theory of twin-width.

[5] This item implies the other ones under the complexity assumption FPT$\neq$AW[$*$].

1.3.1 Task 1: The missing pieces

For every class $\mathcal{C}$ that we showed of bounded twin-width, our proof came with a polynomial-time algorithm reporting $O(1)$-sequences for graphs of $\mathcal{C}$. Yet we do not know how to efficiently approximate the twin-width in full generality. The big missing piece is:

Question 1. *Find an efficient algorithm that, given a graph G and an integer k, correctly reports that the twin-width of G is greater than k, or outputs an $f(k)$-sequence of G.*

For the theory, the magnitude of f does not really matter, as long as f is computable. Of course, for practical purposes, getting f as low as possible is an important additional challenge. To make Question 1 formal, we should say what we mean by *efficient*. There are several possible answers. Ideally the algorithm would run in time $k^{O(1)}m$ on m-edge graphs. For most applications, it is not crucial that the dependency in k is polynomial. A running time in $g(k)m$ for any function g, would allow to conclude that first-order model checking is linear-time (in the number of edges) solvable on graphs of bounded twin-width, invoking Theorem 3. The next best thing would be a fixed-parameter algorithm in $g(k)n^c$, on n-vertex graphs, for some hopefully small constant c. Finally even a slice-wise polynomial algorithm in $n^{g(k)}$ would have some merits. With such an approximation, we could conclude that first-order model checking is polynomial-time solvable on graphs of bounded twin-width.

We resolved Question 1 for ordered graphs, or equivalently, for matrices on finite alphabets seen as ordered structures.

Theorem 7 ([7])**.** *Given as input an $n \times n$ matrix M over an alphabet of constant size, and an integer k, there is an $2^{2^{O(k^2 \log k)}} n^{O(1)}$-time algorithm which*
- *correctly reports that the twin-width of M if greater than k, or*
- *outputs a $2^{O(k^4)}$-sequence for M.*

Besides improving the approximation ratio, getting the running time to $f(k)n^2$ (linear in the input size), or even to $f(k)n$ for sparse matrices, would strengthen Theorem 6. Indeed it would yield a fixed-parameter linear-time (instead of polynomial-time) algorithm for model checking on ordered binary structures. A simple implementation of the current algorithm runs in $f(k)n^3$.

Theorem 7 gives some hope and a new perspective for (unordered) graphs. During the tour, we mentioned that finding a mixed free order is enough to efficiently compute a contraction sequence. When a mixed free order is added to the structure, the twin-width remains unchanged. This is perhaps too strong a property. We now know that finding any total order that, when added, do not make the twin-width go from bounded to unbounded, would be sufficient to solve Question 1 (see Fig. 3).

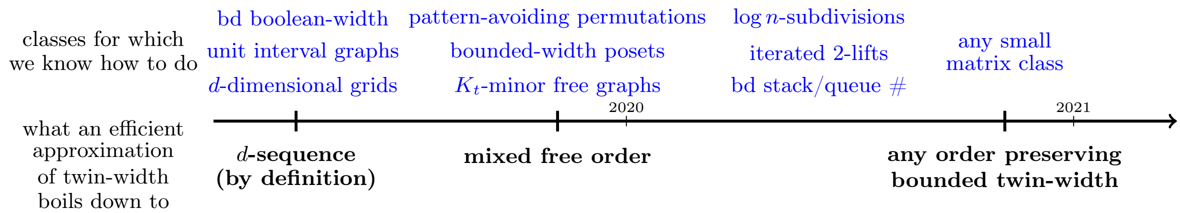

Figure 3: Progress on computing $O(1)$-sequences and classes shown of bounded twin-width.

On bounded-degree graphs, the problem of finding a sequence can be expressed without resorting to trigraphs. As the *black degree* (i.e., the number of black edges incident to a vertex) cannot increase in our trigraph contractions, we are simply asking if there is a sequence of vertex identifications that keeps the degree below a constant threshold. Independently of twin-width, this is a fairly natural question. Similar problems have already been investigated (more often than not, of successively contracting an input graph to fall into a given class) but not this particular one. It is still a challenging question, since some cubic expanders have bounded twin-width, whereas random cubic n-vertex graphs subdivided $o(\log n)$ times (which offer a lot of contractions that do not increase the degree) have almost surely twin-width $\omega(1)$ [4].

An obstacle to solving this problem is that scaling lower bounds to the twin-width of bounded-degree graphs turn out to be elusive. If we know that subcubic graphs globally have unbounded twin-width due to a counting argument, we are currently unable to:

Question 2. *Pinpoint a family $\{G_i \mid i \in \mathbb{N}\}$ where every G_i is a subcubic graph of twin-width at least i.*

This problem constitutes a typical example of "finding hay in a haystack": Every cubic graph almost surely works. Yet Question 2 is open even for more restricted graph parameters like queue-number (or the related stack number, track number, and geometric thickness) whose unboundedness on bounded-degree graphs is only known via a counting argument [49]. Thus figuring this question out will not only help our understanding in twin-width, but provide a missing construction in these contexts. The answer, contrary to the question, need not be difficult. One likely needs to find the right angle or perspective, which could come from multiplying the equivalent views on twin-width; a natural part of theory building.

The flip side of the same coin is that we need to better understand the structure of bounded twin-width graphs. A usual impulse is to see how twin-width relates to (combinations of) the principal graph invariants: chromatic number, clique number, treewidth, Hadwiger number, etc. Bounded twin-width graphs are χ-bounded [5], that is, one can color any graph with *function of twin-width and clique number* many colors. Although we do not know if the dependency of that function in the clique number can be made polynomial.

Question 3. *Are classes with bounded twin-width polynomially χ-bounded?*

The polynomial χ-boundedness of graphs of bounded rank-width (or equivalently clique-width) has been recently obtained [3]. It uses tools from the algebraic theory of formal languages, and mainly a theorem of Colcombet [14] building upon the so-called Factorization Forest theorem of Simon [46]. In essence, the latter results say that for every regular language L, each word w can label the leaves of a (universally) bounded-height tree (with large arity) in such a way that the membership to L of infixes of w can be determined in constant time. Somehow this can be applied to the tree layout of a rank-decomposition. Although bounded twin-width graphs are more general than bounded rank-width graphs, they also admit a tree-based sparse model (called *ordered union tree* [5] or *twin-model* [11]). A counterpart of the previous machinery on bounded twin-width graphs is of interest for the χ-binding function, but also in the context of graph compression and labeling schemes (see Section 1.4.4).

If this approach fails to improve the χ-binding function, it might simply be because graphs of twin-width at most d are *not* polynomially χ-bounded. After all, they significantly extend bounded rank-width graphs whose polynomial χ-boundedness is a delicate result. Therefore they could form a counter-example to the following important conjecture by Louis Esperet [22]: Every hereditary class that is χ-bounded is polynomially χ-bounded. In particular, this conjecture implies that trees T satisfying the Gyárfás-Sumner conjecture (that T-free graphs are χ-bounded) also satisfy the Erdős-Hajnal conjecture (that T-free graphs have a polynomially large clique or independent set). The P_t-free graphs make for another natural candidate to try and disprove Esperet's conjecture, but it seems easier to build "complicated" bounded twin-width graphs rather than P_t-free ones.

UPDATE: Sokołowski and Mi. Pilipczuk have announced a partial resolution (by the positive) of Question 3 showing that classes of bounded twin-width are quasipolynomially χ-bounded, and Esperet's conjecture has been disproved [12] (obviously not using twin-width).

A crucial component of treewidth and rank-width theory is the existence of exploitable certificates that the width is large. For treewidth they take the shape of well-linked sets, brambles, and grid minors, and for rank-width, tangles. These dual objects render algorithms approximating those widths possible. If, on the way to producing a decomposition, the algorithm gets stuck, it does so because it discovers such a certificate. The certificate can then be output as a token of the width being larger than anticipated. We did find a twin-width dual for ordered graphs, and obtained an algorithm following the same principle. However we currently do not have a convenient certificate of high twin-width for (unordered) binary structures, one that could help with Question 1.

Question 4. *Is there a convenient dual to twin-width?*

Recently we realized that bounded rank-width/clique-width/boolean-width has a natural reinterpretation in terms of contraction sequences. We discuss this further in the second task of WP1. This brings some hope that the sought dual could come from a variant on the theme of tangles.

Another noteworthy strength of treewidth is the companion concept of graph minor, whose operations preserve the treewidth bound, while it sets a well-quasi-order over graphs. Among the several operations that do not increase twin-width, we hope to find one with the same property.

Question 5. *Is there a well-quasi-ordering associated to twin-width?*

If so, it is perhaps possible to rewrite some parts of the *Graph Minors* series [45] in a more general setting. One year before twin-width was defined, a new kind of decomposition was developed for classes including many (but not limited to) proper minor-closed classes (see for instance [17, 18]). This decomposition is dubbed *product structure theorem* (after Robertson and Seymour's structure theorem for every proper minor-closed class), and it says that graphs from these classes are subgraphs of a strong product between a path and a graph of bounded tree-width. It has been the crucial new ingredient in settling long-standing open questions in graph drawing, adjacency labeling schemes, and non-repetitive coloring. We started to discuss some connections of this program with twin-width [4] but it feels like there is more to unravel, in particular in the new lights that these two notions shed on graphs excluding a fixed minor. We will come back to it in the third task of WP1 and in WP2.

1.3.2 Task 2: Generalizations and restrictions of twin-width

Extensions. As twin-width strengthens our understanding of graphs, and even more so, of ordered graphs or 0,1-matrices (see Theorem 6), the natural next step is to export it to more general, or simply different, settings.

If we have a *twin-width* of unordered [10] and ordered matrices [7], our definitions only fit matrices on finite alphabets. In the second work package, we propose to develop faster algorithms on bounded twin-width matrices. It is thus opportune to try and generalize our framework to matrices over an infinite field. A natural extension of the contraction process is to allow contractions of pairs of (row or column) vectors that are close to being collinear. This is a viable attempt, although it does not preserve all the characterizations of twin-width of $0,1$-matrices. On the other hand, more conservative definitions may simply be too restrictive. The philosophical question of what *should* twin-width be ensues.

Question 6. *What do we want to call* bounded twin-width *for matrices over infinite fields?*

This very issue is more poignant when generalizing twin-width to hypergraphs. We should here start with the simpler case of uniform hypergraphs, even 3-uniform hypergraphs. And since, as far as twin-width is concerned, we understand ordered structures better, let us even begin with 3-dimensional (ordered) tensors. Several (non-equivalent) definitions of twin-width are possible based on contracting similar slices, on the absence of complex divisions, on introducing auxiliary binary relations, or on moving to a higher-dimensional space.

Question 7. *What do we want to call* bounded twin-width *for tensors or hypergraphs?*

Concretely two model-theoretic characterizations of bounded twin-width for ordered binary structures (dependence and monadic dependence) do not collapse on ordered ternary structures. Therefore there is a real choice to make for the higher-dimensional twin-width. If we are optimistic, a possible stance is that *several* distinct twin-width-inspired definitions will prove useful. Concretely we want to use twin-width to obtain (some, all is impossible) of the equivalences of Theorem 6 for higher-arity relations.

Matroids are another common generalization of graphs. For matroids representable over finite fields, twin-width can be defined as the minimum twin-width of a matrix representing the matroid. With that definition, we can show that the cycle matroid of any $K_{t,t}$-free, bounded twin-width graph has bounded twin-width itself. It is possible (and would be interesting) that the converse holds. A clear application is model checking on matroids. The computational complexity of monadic second-order (MSO) model checking on linear matroids over finite fields is well understood: It is tractable precisely on classes with bounded branch-width [33]. For FO model checking, though, only partial results are known (see for instance [26]). Hopefully the inception of twin-width in that context may help shaping up a dichotomy of the kind known for MSO model checking.

Restrictions. As already mentioned, we recently realized that bounded boolean-width (which is equivalent to bounded clique-width or rank-width) can be characterized in terms of our contraction sequences. To specialize bounded twin-width to bounded boolean-width one just needs to add the requirement that every trigraph of the sequence has its red connected components (i.e., the connected components of the graph induced by the red edges only) of bounded size. One can then drop the condition on the maximum red degree which is now redundant. This provides an arguably simpler definition of bounded clique-width/rank-width graphs, where no labels nor tree layout is necessary. It is tempting to revisit the main results on the topic and see whether we can get simpler proofs or new insights with that characterization.

Question 8. *Is the characterization of bounded clique-width/rank-width via contraction sequences (pedagogically) useful?*

UPDATE: We believe it is. It for instance gives a unifying view for FPT MSO model checking in bounded clique-width classes (which is roughly as far as one can go) and FO model checking in several monadically dependent classes; see [9] and `http://perso.ens-lyon.fr/edouard.bonnet/talk/jga21.pdf`.

Still following the line of reformulating bits of graph theory in terms of contraction sequences, proper minor-closed classes can be characterized as subgraph-closed sparse (i.e., $K_{t,t}$ free) classes with bounded *spanning twin-width*. The *spanning twin-width* is the minimum twin-width of a binary structure consisting of the initial graph and one of its rooted spanning forests encoded as a tree order (with a directed edge from any node to any descendant node). This definition may seem a bit arbitrary, but it generalizes the notable case when the mixed free order may follow a Hamiltonian path. In that particular case, contractions can be realized along this path as *edge* contractions. Let us actually call *edge twin-width* this very restriction. Spanning and edge twin-widths could coincide. If they do, we may not be so far from an approximation of twin-width when restricted to proper minor-closed classes.

Question 9. *Is there an efficient algorithm to approximate the edge twin-width?*

We have reasons to believe that this question is strictly easier than Question 1. Indeed both short and long subdivisions of cliques have unbounded edge twin-width, hence do not need to be distinguished by the algorithm. On the contrary, an approximation of twin-width has to somehow realize that $o(\log n)$-subdivisions are "no-instances", while $\Omega(\log n)$-subdivisions are "yes-instances" [4]; the latter fact coming from the non-trivial property that every permutation is the product of $O(\log n)$ permutations whose matrices are $O(1)$-mixed minor free.

1.3.3 Task 3: Accessibility and diffusion

The twin-width series of papers already spans hundreds of pages. While we tried to write everything as simply and clearly as possible, it still requires quite a bit of effort for an interested reader to get up to date with the current theory. As all the active areas of theoretical computer science and combinatorics are expanding and speeding up, it is increasingly hard to keep up even with one's very field. The personal experience of the coordinator is that the list of papers, results, or simply tricks that he *should* read or know almost invariably grows in a ruthless backlog. The one situation where this list decreases is when a nice expository paper, an online tutorial, a detailed survey, a monograph, or a textbook is released on a topic of his interest.

In two or three years, we should have enough material and perspective for a first version of a monograph on twin-width. As an intermediate step toward that goal, the coordinator plans to write his habilitation thesis on the topic. The advantage of a monograph over the original papers are manyfold: slower pace, streamlined exposition, better organization, all-in-one, etc. Oral presentations can still convey ideas more effectively or, at least, nicely supplement the write-ups. We have already given several talks and tutorials on twin-width, most of which have been recorded and are publicly available (for instance [50, 41]). Throughout the project, we will give many more presentations. If at some point it becomes helpful, we will organize the recorded talks, as well as short zoom-ins on important proof elements, in a convenient fashion (playlist, virtual mind map, etc.). A couple of classes in a second-year master course by R. Watrigant, S. Thomassé, and the coordinator, opening next September [21] will be devoted to twin-width.

Let us finally mention that cleaning the theory can also be motivated by concrete questions. For instance, we currently have at least three proofs that planar graphs have bounded twin-width: via K_t-minor freeness, via the queue number, and directly via the product structure theorem [18]. All these approaches use at some point Theorem 2 (or rather a sparse version of it) and therefore only provide a gigantic upper bound. In the following, we primarily ask for an elementary proof *not* using Theorem 2 and, thereby, the Marcus-Tardos theorem.

Question 10. *Find a better upper bound of the twin-width of planar graphs.*

UPDATE: Jacob and Pilipczuk [34] prove that planar graphs have twin-width at most 183, and B., Kwon, and Wood [20] show that planar graphs have reduced-bandwidth (where the bound on the red graphs is on the bandwidth rather than on the degree) at most 466, and that classes of genus g have reduced-bandwidth $O(g)$.

Getting the exact value, which most likely has a single digit, is currently very much out of reach. We do not have a particularly good lower bound: The dodecahedron tells us that the twin-width of planar graphs is at least 4, and we do not know of an example with twin-width at least 5. Related to that particular challenge, a more general question is to tighten the connection between mixed freeness and twin-width.

Question 11. *Obtain an improved twin-width bound in Theorem 2.*

A single-exponential dependence should be obtainable for sparse classes. In general classes, though, this will require more effective, and hopefully simpler, arguments. Improving the dependence significantly further is impossible due to a randomized construction by Fox [23].

Either Theorem 2 or Theorem 4 allows us to derive that bounded twin-width is preserved under taking (non-induced) subgraphs in $K_{t,t}$-free graph classes. An alternative and elementary proof of that fact would lead to a good bound for the twin-width of planar graphs (even apex-minor-free graphs), via the product structure theorem.

Question 12. *Give a direct reason why the twin-width of subgraphs of G is bounded by a function of twin-width and biclique number of G.*

1.4 WP2: Theoretical applications

We detail some promising areas of application for twin-width, what we know, and possible ways forward.

1.4.1 Model checking and finite model theory

As previously stated, first-order (FO) model-checking can be solved efficiently (in fixed-parameter linear time and quasi-linear time, respectively) on binary structures given with an $O(1)$-sequence and on nowhere dense classes. On general graphs, this problem is AW[$*$]-complete, a strong indication that fixed-parameter algorithms –in the size of the sentence to check– are impossible. A class $\mathcal{C}$ is *monadically dependent* (or *monadically NIP*) if there is no first-order transduction from $\mathcal{C}$ onto the class of all graphs [1], and *monadically independent* otherwise. There is an optimistic, yet believable, conjecture (see for instance in [25]) that every monadically dependent class admits a fixed-parameter tractable FO model checking. If true, this is likely to be as far as one can get on hereditary classes: FO model checking is "morally"[6] as hard on monadically independent classes as it is on general graphs.

Question 13. *Is FO model checking fixed-parameter tractable on every monadically NIP class?*

To start somewhere with this ambitious conjecture, a natural thing to do is to look for monadically NIP common generalizations of bounded twin-width and nowhere dense classes, still capturing some favorable features of either or both kinds of classes. As a first step, algorithms for superclasses of bounded twin-width and bounded expansion, or even bounded twin-width and bounded degree, would be interesting. There are several avenues to explore.

One of them is the notion of low twin-width covers. Generally a class $\mathcal{C}$ has *low X covers* if there are functions f and g, and for every integer k, the vertex set of any $G \in \mathcal{C}$ can be covered by $f(k)$ sets $A_1, \ldots, A_{f(k)}$, such that every subset of $V(G)$ of size k lies entirely in one A_j, and every $G[A_i]$ has parameter X bounded by $g(k)$. Classes with low twin-width covers obviously generalize bounded twin-width (by taking $f(k) = 1$ and $g(k) = O(1)$) and classes of bounded expansion (which coincide with low treedepth covers). The same lift of nice properties from bounded treedepth to bounded expansion classes [19, 27], may also work from bounded twin-width to low twin-width covers. Notably, though, it is not immediate that classes with low twin-width covers are monadically dependent; which is a necessary condition for this approach to be fruitful.

Another direction is to relax the twin-width definition. Instead of requesting the contraction sequence to end at a single vertex, we can ask for a *partial* sequence of contractions reaching a target class of trigraphs, such as bounded degree. Indeed a partial $O(1)$-sequence to a bounded degree graph allows, by Gaifman locality theorem [24] combined with Theorem 3, to solve efficiently FO model checking on the original graph. Whether that still holds when the target class is of bounded expansion or nowhere dense is more challenging.

A class $\mathcal{C}$ of finite structures is *stable* if it cannot interpret arbitrarily long linear orders. More concretely, no first-order formula $\phi(x, y)$ is such that for every integer h, there is $G_h \in \mathcal{C}$ and h distinct vertices $a_1, \ldots, a_h \in V(G_h)$, with $G_h \models \phi(a_i, a_j)$ holding if and only if $i \leqslant j$. NIP classes (which, for binary structures, may coincide with monadically NIP ones) comprise all the stable classes, such

[6]Some technicalities come in the way of a formal statement and proof, stemming from the coloring of the transduction and the fact that one could potentially require graphs from the monadically independent class of size superpolynomial in n to produce all the n-vertex graphs.

as bounded-degree graphs, as well as "structured" unstable classes, like unit interval graphs or classes with bounded clique-width. The intuition that NIP structures consist of a stable and an order-like parts is made precise by Simon [47]. This decomposition happens at the level of the types, that is, sets of formulas holding on the structure for a fixed tuple and set of constants (or *parameters*). Bounded twin-width, which is orthogonal to stability and fundamentally rests upon a linear order, certainly qualifies as "order-like". This motivates the following question.

Question 14. *Is FO model checking on (monadically) NIP classes Turing-reducible to itself on bounded twin-width and on stable classes?*

A positive answer to Question 14 would bring us three steps away from resolving Question 13. The three last steps would consist of settling Question 1, obtaining an algorithm for structurally nowhere dense classes (that is, transductions of nowhere dense classes), and confirming the conjecture that stability coincides with structurally nowhere denseness.

1.4.2 Approximation algorithms

If we have a fairly good understanding of the computational edge that $O(1)$-sequences give as far as exact algorithms are concerned, we know far less about approximation algorithms. Maximum Independent Set is a notoriously inapproximable problem on general graphs. For any $\varepsilon > 0$, a polytime $n^{1-\varepsilon}$-approximation implies that P = NP [31, 51]. If we obtained constant-factor approximations for Minimum Dominating Set on graphs given with an $O(1)$-sequence [5], the approximability status of Maximum Independent Set on bounded twin-width graphs remains totally elusive.

Question 15. *What is the best polytime approximation factor achievable for* Maximum Independent Set *on bounded twin-width graphs given with an $O(1)$-sequence?*

Even an $O(\sqrt{n})$-approximation would be new and seems non-trivial, while it is ruled out on general graphs. Interestingly any constant-approximation for this problem can be turned into an approximation scheme [5], that is, a family of approximation algorithms with ratio arbitrarily close to 1. On every standard class shown to have bounded twin-width, at least a constant-approximation can be routinely obtained. Yet we are skeptical that such an approximation algorithm is generalizable to every graph of twin-width at most d. We then wonder where the hard instances hide.

Question 15 is also intended as a meta-question. One can ask the same for any optimization problem that is first-order definable. In such a framework, one tries to minimize or maximize the size of X such that $\varphi(X)$ holds, where φ is a first-order formula with a single free set-variable, X. The stark contrast observed between Maximum Independent Set and Minimum Dominating Set indicates that this question may have a rich answer, in the form of a classification splitting formulas $\varphi(X)$ for which, say, a constant-approximation is possible from formulas for which such an algorithm is unlikely. It is also interesting to see whether we can find better approximations for Integer Linear Programming on bounded twin-width matrices.

1.4.3 Finitely generated groups

Twin-width can be generalized to infinite (possibly uncountable) graphs due to its characterization with vertex orderings excluding unbounded mixed minors. We showed that the boundedness of twin-width for finitely generated Cayley graphs do not depend on the set of generators [4]. Thus we may speak of the twin-width of a finitely generated group. This is interesting on its own and relevant to a conjecture we previously made on hereditary classes with small growth. Every bounded twin-width class is *small* [4], in the sense that there are at most $n!c^n$ n-vertex labeled members of the class, for some constant c. We previously conjectured there that the converse holds for hereditary classes, and proved it in the case of linearly ordered classes [7].

Question 16. [small conjecture, now refuted] *Are all hereditary small classes of bounded twin-width?*

In particular, this would have implied that the finite induced subgraphs of any bounded-degree infinite Cayley graph have bounded twin-width. Very recently, we ruled out this option thereby disproving the small conjecture [6]: There *are* small hereditary classes of unbounded twin-width. Our counterexample is based on a group-theoretic construction due to Osajda [42], in turn refining a classic construction of so-called *Gromov monsters* [29]. Osajda builds a finitely generated infinite Cayley graph that isometrically contains infinitely many graphs from any sequence of finite bounded-degree graphs whose girth and diameter diverge with the same speed. This result allows us to refute Question 16 because on one hand, n-vertex cubic graphs with girth and diameter $\Theta(\log n)$ almost surely have superconstant twin-width, and on the other hand, the class of all the finite induced subgraphs of a bounded-degree Cayley graph is small [4], and by design, hereditary.

As a side note related to Section 1.4.1, the following could still receive a positive answer.

Question 17. *Is FO model checking fixed-parameter tractable on small hereditary classes?*

As another side note, the counterexample to Question 16 being probabilistic, we still do not know of an explicit small family of bounded-degree graphs with unbounded twin-width. Now more related to the current section, we henceforth know that the property of having bounded twin-width non-trivially splits the finitely generated groups in two categories.

Question 18. *Does bounded/unbounded twin-width reflect a known dichotomy of finitely generated groups?*

As we previously conjectured that only the bounded twin-width case were populated, it goes without saying that the first examples that come to mind have bounded twin-width. For instance, the free groups (infinite r-ary trees), $\mathbb{Z}^d$ (infinite d-dimensional grids), the Lamplighter group, products of two bounded twin-width groups, Abelian groups, all have bounded twin-width. Thus we can *order* a wide variety of groups (but not all) in such a way that the trace of every group element on its corresponding Cayley table (or rather its infinite equivalent) avoids a fixed pattern. This property should not help solving the classical problems from algorithmic group theory (word problem, subgroup membership, knapsack, etc.), as they tend to remain undecidable on very simple groups (like direct products of free groups). However this somewhat unexpected combinatorial property shared by the "non-pathological" finitely generated groups should be of some use. We hope that our exchanges with group theorists (like Romain Tessera with whom we started a collaboration [6]) will reveal one, or a solution to Question 18.

1.4.4 Labeling schemes

An *adjacency labeling scheme* (or *labeling scheme*, for short) formalizes the compression of graph encodings, when graphs come from a fixed class $\mathcal{C}$. It consists of a decoding function $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ and for every $G \in \mathcal{C}$ a labeling function $\ell : V(G) \to \{0,1\}^*$, such that $A(\ell(u), \ell(v)) = 1$ if and only if uv is an edge of G. One then tries to make the words encoding the vertices (the images of ℓ), called *labels*, as short as possible. An *$f(n)$-bit labeling scheme* is such that $|\ell(v)| \leqslant f(n)$ for every n-vertex graph $G \in \mathcal{C}$ and every $v \in V(G)$. Bounded twin-width classes admit $O(\log n)$-bit labeling schemes [4]. There are three directions in which we want to improve this result: efficiency, compactness, and generality.

A labeling scheme is algorithmically useful if one can encode and decode quickly. A typical quantitative delineation is to further require that the labeling (encoding function) can be computed in polynomial time, and the decoding function, in constant time in the word RAM model with words of size $\Theta(\log n)$. Such a labeling scheme is then said *efficient*.

Question 19. *Do bounded twin-width classes have* efficient *$O(\log n)$-bit labeling schemes?*

Our labeling scheme has neither of the two efficiency properties. The encoding requires an $O(1)$-sequence of the graph, thus currently cannot be done in polynomial time. It is to be expected that every labeling scheme of bounded twin-width graphs shares that requirement. Therefore Question 19 is tied to Question 1. However a positive answer to Question 1 does not settle Question 19 just yet. Indeed the decoding takes $O(\log n)$ time. Getting that time down to $O(1)$, at least in average, should involve a data structure useful in several places (see for instance Section 1.4.5). Let us illustrate that point with a direct and closely related question. Every n-vertex graph with twin-width $O(1)$ can be compressed in a sparse structure on $2n-1$ vertices: namely, an n-leaf binary tree augmented by $O(n)$ "transversal" edges [4, 5, 11]. Given only the sparse representation, we can query edges in time $O(\log n)$.

Question 20. *Can n-vertex graphs of bounded twin-width be represented by binary structures with $O(n)$ edges, and adjacency queries done in amortized constant time?*

UPDATE: Pilipczuk, Sokołowski, and Zych-Pawlewicz [44] have obtained a linear-sized data structure (computable in quasilinear time from $O(1)$-mixed free adjacency matrices) which enables queries in time $O(\log \log n)$.

Bounded twin-width classes are small, and as such, may in principle have labeling schemes using only $(1+o(1))\log n$ bits. The product structure theorem has recently yielded $(1+o(1))\log n$-bit labeling schemes for *flat* classes [16], a particular case of sparse classes with bounded twin-width. Flat classes include apex-minor free graphs, K_t-minor free graphs with bounded degree, and k-planar graphs, but K_6-minor free graphs do not make a flat class.

Question 21. *Do bounded twin-width classes have $(1+o(1))\log n$-bit labeling schemes?*

A first step is to ask Question 21 for K_t-minor free graphs. The twin-width and the product structure theories tell two new stories on (some) proper minor-closed classes. The former is perhaps too general whereas the latter is not general enough. There might be a third story fitting proper minor-closed classes more tightly, a looser decomposition than subgraphs of strong products but stronger than mixed-minor free adjacency matrices.

Let us now move on in the direction of greater generality. The implicit graph conjecture asserts that every hereditary factorial class (i.e., with growth $n^{O(n)}$) has an $O(\log n)$-bit labeling scheme. As already stated, we showed the implicit graph conjecture in the special case of bounded twin-width classes [4]. Interval graphs are an example of an unbounded twin-width factorial class with a straightforward $2\log n$-bit labeling scheme. If we want to make significant further progress on the implicit graph conjecture with the twin-width approach, we should be able to unify these two results. Is there any "decomposition" of interval graphs into bounded twin-width pieces? This voluntarily vague question could be useful in other contexts, depending on the nature of the decomposition. For instance, it could extend our unified understanding of χ-bounded classes.

UPDATE: Hatami and Hatami [32] have refuted the implicit graph conjecture.

1.4.5 Linear algebra

The linear-time dependency of our exact algorithms [10, 5] renders twin-width an interesting parameter, not only for NP-hard problems, but for problems within P. For instance, we showed how to compute in $O(n)$-time single-source shortest paths of n-vertex graphs given with an $O(1)$-sequence [5]. Since these graphs can be dense, this is sublinear in the number of edges of the input. This is made possible by a succinct representation of total size $O(n)$ alluded to in Section 1.4.4. We can compute in quasi-linear time the product of two matrices given with such a succinct representation [8]. It would be better to require a mere contraction sequence, and in principle, a linear-time dependency is possible.

Question 22. *Is there an $O(n)$-time algorithm to multiply two $n \times n$ matrices of bounded twin-width given with an $O(1)$-sequence?*

More generally, we need a fast algorithm (ideally in linear time) to go from a contraction sequence to a succinct representation. Question 22 deals with *ordered* matrices, on which we can find in polynomial time $O(1)$-sequences when they exist. To make a positive answer to Question 22 totally genuine, we need to find a sequence in time linear in the input size (as discussed in the first task of WP1). Only then, we can claim to have an $O(n)$-time algorithm to multiply two *sparse* $n \times n$ matrices of bounded twin-width. A linear-time Gaussian elimination of systems described by bounded twin-width matrices may lead to faster algorithms for Maximum Matching in planar graphs (see [39]).

1.4.6 Computational geometry

One can export structural (hyper)graph parameters, such as clique-width, to point sets. A purely combinatorial approach is to consider the clique-width of the ordered 3-uniform hypergraph formed by all the triples of points oriented clockwise [13]. After we get a satisfying notion of twin-width for ordered hypergraphs (see second task of WP1), we can follow this approach and see how general point sets of bounded twin-width are. We can then ask for efficient algorithms solving hard geometric problems for which only the order types count (like visibility ones), on inputs of bounded twin-width.

For metric problems, however, the actual point coordinates matter. In that context, we want a contraction sequence that takes distances into account. This brings us back to graphs, by considering the twin-width of (unit) disk graphs. Although one can imagine that, for some applications such as clustering, it would make sense to keep a purely geometric viewpoint. For instance, contracting two points p, q would (delete them and) create a new point at the weighted barycenter of p and q.

Question 23. *Can we define successful notions of twin-width for point sets?*

We also think that the Marcus-Tardos theorem, despite his geometric flavor (sufficiently dense point sets on the integer lattice are "two-dimensional"), have not been leveraged all that much in geometric problems. This is just a couple of the several questions that we want to investigate at the interface between geometry and twin-width.

1.5 WP3: Practical applications

1.5.1 Heuristics and algorithms to find contraction sequences

The first step of WP3 is to confirm (or challenge) our intuition that the twin-width of most real-life networks is small. Large social networks typically contain dense communities of hundreds or thousands of nodes. That alone makes their treewidth at least of that order, thus not directly useful for exact computation [37]. We estimate that their twin-width is one or two orders of magnitude smaller. Road networks tend to contain large grid-like structures, thus even their rank-width is high. Their planarity (or near-planarity) makes them of low twin-width.

We want to compute lower and upper bounds of twin-width for the usual benchmarked networks. In addition to calibrating our expectations regarding what algorithms based on twin-width can achieve, this will initiate a healthy interplay between the theory and the practice of approximating twin-width.

1.5.2 Implementing specific twin-width-based algorithms

If the algorithm generically solving FO model checking is clearly *not* a practical one, specific central problems like k-Independent Set, admit reasonable, single-exponential algorithms [5]. More precisely, the latter problem has an $O(k^2 d^{2k} n)$-time algorithm on n-vertex graphs given with a d-sequence.

The hidden constant of this algorithm is good, and the dependency in d and k should in practice be somewhat better than the worst-case analysis bound of $k^2 d^{2k}$. We wish to implement this algorithm and compare its execution time to the ones of exact algorithms based on other parameterizations or approaches. If the selected problem for the edition of the programming contest PACE (https://pacechallenge.org/) coinciding with the postdoc employment is amenable to the twin-width approach, we will participate in the challenge.

Twin-width may also be useful for classical (non-parameterized) problems, like Max Independent Set. If there are excellent solvers for the latter problem based on branch-and-reduce (intertwined kernelization and branch-and-bound) [36], these algorithms do not perform well in dense instances. It would be interesting to see how our approach fares in practice on real-life dense instances. In the context of twin-width-based algorithms for classical problems, the contraction sequences should now minimize the number of connected subgraphs in what is induced by the red edges (rather than the maximum red degree) [5].

1.5.3 Possible connections with other areas

Hierarchical clustering. In data mining, hierarchical clustering aims to organize datapoints at the leaves of a rooted binary tree in such a way that the datapoints of every rooted subtree constitutes a cluster. The similarity between datapoints is modeled by a weighted graph. Then, hierarchical clustering can be turned into a precise combinatorial problem by specifying a cost function that the tree should minimize. There is a tempting parallel to draw with our contraction sequences. We do benefit from considering the contraction process sometimes forward (parameterized algorithms, etc.) and sometimes backward (χ-boundedness, etc.), very much like hierarchical clustering benefits from the agglomerative (bottom-up) and divisive (top-down) approaches. Twin-width can loosely be thought of as a hierarchical clustering where similarity is based on the neighborhood's closeness. Admittedly, in the definition of twin-width, the order in which clusters are aggregated/divided is important. We want to determine whether this is just a vague resemblance or this has more profound ties.

Image compression. The succinct representation of bounded twin-width graphs (see Section 1.4.4) compresses an n-vertex graph with possibly $\Theta(n^2)$ edges into a word of length $O(n)$. As we have a notion of twin-width for matrices on finite alphabets, we may talk of the twin-width of an image. We want to investigate if dynamic schemes similar to twin-width are already known and used for image compression, how small the twin-width of typical images is, and whether the twin-width approach can be performant in that context.

References

[1] 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.

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

[3] Marthe Bonamy and Michal Pilipczuk. Graphs of bounded cliquewidth are polynomially χ-bounded. *Avances in Combinatorics, CoRR*, abs/1910.00697, 2019.

[4] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1977–1996. SIAM, 2021.

[5] 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.

[6] Édouard Bonnet, Colin Geniet, Romain Tessera, and Stéphan Thomassé. Twin-width VII: groups and the small conjecture. *In preparation*, 2022.

[7] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Toruńczyk. Twin-width IV: ordered graphs and matrices. *CoRR*, abs/2102.03117, Accepted at STOC 2022.

[8] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, and Stéphan Thomassé. Twin-width V: parity minors, modular counting, and matrix multiplication. *Manuscript*, 2022.

[9] 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.

[10] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: Tractable FO Model Checking. *J. ACM*, 69(1):1–46, nov 2022.

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

[12] Marcin Brianski, James Davies, and Bartosz Walczak. Separating polynomial χ-boundedness from χ-boundedness. *CoRR*, abs/2201.08814, 2022.

[13] Onur Çagirici, Petr Hlinený, Filip Pokrývka, and Abhisekh Sankaran. Clique-width of point configurations. In Isolde Adler and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science - 46th International Workshop, WG 2020, Leeds, UK, June 24-26, 2020, Revised Selected Papers*, volume 12301 of *Lecture Notes in Computer Science*, pages 54–66. Springer, 2020.

[14] 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.

[15] Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-rewriting hypergraph grammars. *J. Comput. Syst. Sci.*, 46(2):218–270, 1993.

[16] Vida Dujmovic, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 577–588. IEEE, 2020.

[17] Vida Dujmovic, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar graphs have bounded queue-number. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 862–875. IEEE Computer Society, 2019.

[18] Zdenek Dvorák, Tony Huynh, Gwenaël Joret, Chun-Hung Liu, and David R. Wood. Notes on graph product structure theory. *CoRR*, abs/2001.08860, 2020.

[19] Zdenek Dvorák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *J. ACM*, 60(5):36:1–36:24, 2013.

[20] David Wood Édouard Bonnet, O-joung Kwon. Reduced bandwidth: a qualitative strengthening of twin-width in minor-closed classes (and beyond). *Manuscript*, 2022.

[21] Édouard Bonnet, Stéphan Thomassé, Rémi Watrigant. Master 2 course "graph decompositions". http://perso.ens-lyon.fr/edouard.bonnet/graphDecompositions.htm, 2021-2022.

[22] Louis Esperet. Polynomial χ-boundedness. https://oc.g-scop.grenoble-inp.fr/esperet/Main/Problems, 2012.

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

[24] Haim Gaifman. On local and non-local properties. In J. Stern, editor, *Proceedings of the Herbrand Symposium*, volume 107 of *Studies in Logic and the Foundations of Mathematics*, pages 105–135. Elsevier, 1982.

[25] Jakub Gajarský, Petr Hlinený, Jan Obdrzálek, Daniel Lokshtanov, and M. S. Ramanujan. A new perspective on FO model checking of dense graph classes. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 176–184, 2016.

[26] Tomas Gavenciak, Daniel Král, and Sang-il Oum. Deciding first order properties of matroids. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2012.

[27] Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. In Martin Grohe and Johann A. Makowsky, editors, *Model Theoretic Methods in Finite Combinatorics - AMS-ASL Joint Special Session, Washington, DC, USA, January 5-8, 2009*, volume 558 of *Contemporary Mathematics*, pages 181–206. American Mathematical Society, 2009.

[28] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017.

[29] Mikhail Gromov. Random walk in random groups. *Geometric & Functional Analysis GAFA*, 13(1):73–146, 2003.

[30] 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.

[31] 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. IEEE Computer Society, 1996.

[32] Hamed Hatami and Pooya Hatami. The implicit graph conjecture is false. *CoRR*, abs/2111.13198, 2021.

[33] Petr Hlinený. Branch-width, parse trees, and monadic second-order logic for matroids. *J. Comb. Theory, Ser. B*, 96(3):325–351, 2006.

[34] Hugo Jacob and Marcin Pilipczuk. Bounding twin-width for bounded-treewidth graphs, planar graphs, and bipartite graphs. *CoRR*, abs/2201.09749, 2022.

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

[36] Sebastian Lamm, Christian Schulz, Darren Strash, Robert Williger, and Huashuo Zhang. Exactly solving the maximum weight independent set problem on large real-world graphs. In *Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments, ALENEX 2019*, pages 144–158. SIAM, 2019.

[37] Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data. In Pablo Barceló and Marco Calautti, editors, *22nd International Conference on Database Theory, ICDT 2019, March 26-28, 2019, Lisbon, Portugal*, volume 127 of *LIPIcs*, pages 12:1–12:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

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

[39] Marcin Mucha and Piotr Sankowski. Maximum matchings in planar graphs via gaussian elimination. *Algorithmica*, 45(1):3–20, 2006.

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

[41] Frontiers of Parameterized Complexity youtube channel. Édouard Bonnet, Twin-width. https://www.youtube.com/watch?v=P0_2ya3et24&t=2017s&ab_channel=FrontiersofParameterizedComplexity, 2020.

[42] Damian Osajda et al. Small cancellation labellings of some infinite graphs and applications. *Acta Mathematica*, 225(1):159–191, 2020.

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

[44] Michał Pilipczuk, Marek Sokołowski, and Anna Zych-Pawlewicz. Compact representation for matrices of bounded twin-width. *CoRR*, abs/2110.08106, 2021.

[45] Neil Robertson and Paul D. Seymour. Graph minors I–XXIII. 1983–2010.

[46] Imre Simon. Factorization forests of finite height. *Theor. Comput. Sci.*, 72(1):65–94, 1990.

[47] Pierre Simon. Type decomposition in NIP theories. *Journal of the European Mathematical Society*, 22(2):455–476, 2019.

[48] Egon Wanke. k-NLC Graphs and Polynomial Algorithms. *Discret. Appl. Math.*, 54(2-3):251–266, 1994.

[49] David R. Wood. Bounded-degree graphs have arbitrarily large queue-number. *Discret. Math. Theor. Comput. Sci.*, 10(1), 2008.

[50] IBS Discrete Mathematics Group youtube channel. Eun Jung Kim, Twin-width: tractable FO model checking. https://www.youtube.com/watch?v=d67WI0I8urs&t=14s&ab_channel=IBSDiscreteMathematicsGroup, 2020.

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