Twin-Width V: Linear Minors, Modular Counting, and Matrix Multiplication

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

Ugo Giocanti
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Patrice Ossona de Mendez
Centre d'Analyse et de Mathématique Sociales CNRS UMR 8557, Paris, France
Computer Science Institute, Charles University (IUUK), Prague, Czech Republic

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

Abstract

We continue developing the theory around the twin-width of totally ordered binary structures (or equivalently, matrices over a finite alphabet), initiated in the previous paper of the series. We first introduce the notion of parity and linear minors of a matrix, which consists of iteratively replacing consecutive rows or consecutive columns with a linear combination of them. We show that a matrix class (i.e., a set of matrices closed under taking submatrices) has bounded twin-width if and only if its linear-minor closure does not contain all matrices. We observe that the fixed-parameter tractable (FPT) algorithm for first-order model checking on structures given with an $\mathcal{O}(1)$-sequence (certificate of bounded twin-width) and the fact that first-order transductions of bounded twin-width classes have bounded twin-width, both established in *Twin-width I*, extend to first-order logic with modular counting quantifiers. We make explicit a win-win argument obtained as a by-product of *Twin-width IV*, and somewhat similar to bidimensionality, that we call rank-bidimensionality. This generalizes the seminal work of Guillemot and Marx [SODA '14], which builds on the Marcus-Tardos theorem [JCTA '04]. It works on general matrices (not only on classes of bounded twin-width) and, for example, yields FPT algorithms deciding if a small matrix is a parity or a linear minor of another matrix given in input, or exactly computing the grid or mixed number of a given matrix (i.e., the maximum integer k such that the row set and the column set of the matrix can be partitioned into k intervals, with each of the k^2 defined cells containing a non-zero entry, or two distinct rows and two distinct columns, respectively).

Armed with the above-mentioned extension to modular counting, we show that the twin-width of the product of two conformal matrices A, B (i.e., whose dimensions are such that AB is defined) over a finite field is bounded by a function of the twin-width of A, of B, and of the size of the field. Furthermore, if A and B are $n \times n$ matrices of twin-width d over $\mathbb{F}_q$, we show that AB can be computed in time $\mathcal{O}_{d,q}(n^2 \log n)$.

We finally present an *ad hoc* algorithm to efficiently multiply two matrices of bounded twin-width, with a single-exponential dependence in the twin-width bound. More precisely, pipelined to observations and results of Pilipczuk et al. [STACS '22], we obtain the following. If the inputs are given in a compact tree-like form (witnessing twin-width at most d), called twin-decomposition of width d, then two $n \times n$ matrices A, B over $\mathbb{F}_2$ can be multiplied in time $4^{d+o(d)}n$, in the sense that a twin-decomposition of their product AB, with width $2^{d+o(d)}$, is output within that time, and each entry of AB can be queried in time $\mathcal{O}_d(\log \log n)$. Furthermore, for every $\varepsilon > 0$, the query time can be brought to constant time $\mathcal{O}(1/\varepsilon)$ if the running time is increased to near-linear $4^{d+o(d)}n^{1+\varepsilon}$. Notably, the running time is sublinear (essentially square root) in the number of (non-zero) entries.

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

Keywords and phrases Twin-width, matrices, parity and linear minors, model theory, linear algebra, matrix multiplication, algorithms, computational complexity

40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023).
Editors: Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté;
Article No. 15; pp. 15:1–15:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.STACS.2023.15

Related Version *Full Version*: https://arxiv.org/abs/2209.12023

Funding This work was supported by the ANR projects (French National Research Agency) TWIN-WIDTH (ANR-21-CE48-0014-01) and Digraphs (ANR-19-CE48-0013-01), and is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 810115 – Dynasnet).

1 Introduction

Since its introduction, the treewidth of a graph has proved to be a particularly important concept in graph theory, both in finite model theory [19], in algorithmic design (see for instance the textbook of Cygan et al. [12, Chapter 7]) and in structural analysis (see the Graph Minors series of Robertson and Seymour [27]). This invariant is strongly related to the concept of graph minor. Recall that a minor of a graph is a graph obtained by a succession of edge contractions and vertex or edge deletions. The treewidth of a graph is monotone with respect to this operation, in the sense that the treewidth of a minor of a graph G cannot be larger than the treewidth of G. By a classical theorem by Robertson and Seymour [28], a class of graphs has bounded treewidth if and only if its *minor closure* (that is, the set of all the minors of graphs in the class) does not contain all grids. In particular, a graph with huge treewidth admits a large square grid as a minor. This result, as well as its subsequent qualitative improvements (see [9], for instance), is the basis of the so-called *bidimensionality* algorithmic technique, a win-win argument leveraging low treewidth or the existence of a large grid minor [16].

This paper is the fifth of a series dedicated to a novel invariant of binary structures, the *twin-width* (see Section 2 for formal definitions). This invariant appeared to be particularly relevant for the study of *ordered* binary structures, and especially matrices over a finite alphabet [4]. Some of our results will only need the matrix entries to belong to a finite alphabet, while some will require the entries to belong to a finite field. A *submatrix* of a matrix is obtained by deleting some rows and columns. Most of our results concern (infinite) sets of matrices. In our framework, it will be natural to consider sets of matrices closed under the operation of taking a submatrix. Sets of matrices with this property are called *matrix classes*, analogously to permutation classes, which are classes of permutations closed under taking subpermutations. The alphabet or field being fixed (and having at least two elements), a matrix class is said to be *proper* if it does not include all matrices with entries in the prescribed alphabet or field.

The notion of a *rank Latin division* has been introduced in [4] (see definition in Section 3). It consists of a regular partition of the rows and columns delimiting blocks that either have constant entries or have full rank, in a globally controlled way, where full-rank blocks draw a universal permutation. Just as the grids act as witnesses of a large treewidth, the rank Latin divisions witness a large twin-width: a matrix has either small twin-width or has a submatrix with a large rank Latin division. This is effective: in FPT time, either a contraction sequence of the matrix (witnessing that the twin-width is low) is output or a large rank Latin division is found in a submatrix (witnessing that the twin-width is high).

In this paper, we introduce an operation that plays an analogous role with respect to the twin-width of ordered binary structures that taking a minor plays with treewidth. Applied to 0,1-matrices, this operation consists of a succession of row or column deletions, and replacements of two consecutive rows or columns by their entry-wise sum (modulo 2).

More generally, when applied to matrices over a finite field $\mathbb{F}_p$, this operation consists of a succession of replacements of two consecutive rows or columns by a linear combination of these (over $\mathbb{F}_p$), and any matrix over $\mathbb{F}_p$ obtained this way is a *linear minor* of the original matrix. We say that a matrix class *excludes* a matrix M as a linear minor if M is not a linear minor of a matrix in the class. As expected, every matrix is a linear minor of any matrix having a sufficiently large rank Latin division, thus classes with unbounded twin-width do not exclude any matrix as a linear minor (Lemma 18). It appears that this necessary condition is also sufficient.

▶ **Theorem 1.** *A matrix class over a finite field has bounded twin-width if and only if it excludes some matrix as a linear minor.*

While this characterization of bounded twin-width is related to those expressed in terms of matrix divisions (see Section 2.3 for definitions, and [5, 4] for the corresponding results), the operational nature of linear minors make them closer to what is currently missing in the unordered setting: an operation that is to twin-width what graph minors are to treewidth. In addition, even if the current paper only deals with finite fields, linear minors appear to naturally extend to the case of infinite fields, when the other invariants based on matrix divisions fail to do so. As such, Theorem 1 paves the way to the adequate extension of twin-width to (ordered) binary structures on infinite domains.

Our proof of Theorem 1 involves some (finite) model theoretic arguments. From a model theoretical point of view, a matrix over a finite alphabet of size p is seen as a structure with two linearly ordered sets of elements, the row and column index sets, and p binary predicates expressing the presence of a particular symbol at a specific entry. The logical formulas we will consider will allow distinguishing row and column indices, comparing indices of a same sort, and testing whether the entry of the matrix defined by two indices contains a given symbol.

It appears that the twin-width of ordered structured behaves very nicely with respect to first-order logic and (as we shall see) its modulo-counting extension. This situation is reminiscent of the relation of treewidth (and cliquewidth) with monadic second-order logic [10] and its modulo-counting extension [11].

Indeed, it follows from the results proved in the first paper of the series [5], that first-order model checking (that is: the problem of deciding whether a first-order (FO) sentence φ is satisfied on a structure) is fixed-parameter tractable on matrices over a finite alphabet, when parametrized by φ, the size of the alphabet, and the twin-width of the matrix, provided that some so-called *d-sequence* witnessing the upper bound on the twin-width is given together with the matrix. Here we observe that this result extends to the more expressive first-order logic with modulo-counting (FO+MOD), which is the logic obtained by adding to the standard first-order constructions new quantifiers $\exists^{i[p]}$, where "$\exists^{i[p]}x\ \varphi(x)$" expresses that the number of witnesses x for the formula φ is congruent to i modulo p.

Logical formulas also allow defining new structures from an original structure. This is the essence of the notion of transduction. A *transduction* of binary structures T first colors the elements of a given binary structure $\mathbf{A}$ in all possible ways, thus constructing a set of colored structures. Then, each of these colored structures gives rise to a new binary structure by means of fixed logical formulas, thus constructing a set $\mathsf{T}(\mathbf{A})$ of derived structures, the transduction of $\mathbf{A}$ by T. A *simple interpretation* is the same without the initial coloring process, hence a given structure produces a single other structure. A set $\mathcal{D}$ of structures is a transduction of a set $\mathcal{C}$ of structures if there exists a transduction T with $\mathcal{D} \subseteq \mathsf{T}(\mathcal{C})$. A set $\mathcal{C}$ of structures is *monadically dependent* if the set of all finite graphs is not a transduction of $\mathcal{C}$. (While this actually follows from [1], we will take here this characteristic property as a definition of monadic dependence.) Note that it has been recently proved [8] that, for hereditary classes of structures (like matrix classes) monadic dependence coincides with the classical notion of dependence (or NIP), which is one of the most fundamental dividing lines in model theory; a proof of such a collapse in the particular case of hereditary classes of ordered graphs was previously shown in [4].

In [5], it was proved that for every FO-transduction T of binary structures, the maximum twin-width of a structure in $\mathsf{T}(\mathbf{A})$ is bounded by a function (depending on T) of the twin-width of $\mathbf{A}$. We also extend this result to FO+MOD-transductions. As an example, there is a transduction L_p such that for every matrix M over $\mathbb{F}_p$, the set $\mathsf{L}_p(M)$ is exactly the set of all linear minors of M. Thus, the closure by linear minors of a matrix class with bounded twin-width also has bounded twin-width, from which Theorem 1 follows.

Together with the results established in [4], this leads to the following equivalence, where the equivalence with the properties in bold is proved in the current paper.

▶ **Theorem 2.** *Given a matrix class $\mathcal{M}$ over a finite field, the following are equivalent.*
(i) *$\mathcal{M}$ has bounded twin-width;*
(ii) ***$\mathcal{M}$ excludes a linear minor;***
(iii) *$\mathcal{M}$ is monadically dependent;*
(iv) *every matrix class that is an* FO*-transduction of $\mathcal{M}$ is proper;*
(v) ***every matrix class that is an* FO+MOD*-transduction of $\mathcal{M}$ is proper;***
(vi) *$\mathcal{M}$ is small (i.e. the number of $n \times n$ matrices in $\mathcal{M}$ is at most $2^{O(n)}$);*
(vii) ***every* FO+MOD*-transduction of $\mathcal{M}$ is small;***
(viii) *$\mathcal{M}$ is subfactorial (i.e. the number of $n \times n$ matrices in $\mathcal{M}$ is less than $n!$, for sufficiently large n).*

Assuming that $\mathsf{FPT} \neq \mathsf{AW}[*]$*, those conditions are further equivalent to:*
(ix) FO*-model checking is* FPT *on $\mathcal{M}$;*
(x) FO+MOD***-model checking is* FPT *on $\mathcal{M}$.***

We now consider some consequences of these results (for more, see long version).

We call *rank-bidimensional* a parameterized problem defined on matrices whenever the presence of a large rank Latin division in a submatrix incurs an (easy) FPT algorithm. Thus, we get the following.

▶ **Theorem 3.** *Every* FO+MOD*-definable rank-bidimensional problem is in* FPT.

From Theorem 3, we obtain FPT algorithms for deciding if a (small) matrix is a linear minor of another matrix, for exactly computing the grid number, mixed number, and grid rank of a matrix (see Section 2 for definitions).

Next we show that, over a finite field, the square M^2 of a matrix M with bounded twin-width has bounded twin-width, by expressing the squaring operation as an FO+MOD-transduction. From the characterization in terms of large rank Latin division of submatrices, it follows that if two matrices A and B have small twin-width, then so does the matrix $\left(\begin{smallmatrix} 0 & A \\ B & 0 \end{smallmatrix}\right)$. As $\left(\begin{smallmatrix} 0 & A \\ B & 0 \end{smallmatrix}\right)^2 = \left(\begin{smallmatrix} AB & 0 \\ 0 & BA \end{smallmatrix}\right)$, we deduce:

▶ **Theorem 4.** *There is a computable function $f : \mathbb{N}^2 \to \mathbb{N}$ such that the following holds. Let A and B be two conformal matrices over a finite field $\mathbb{F}_q$, both of twin-width at most d. Then the twin-width of the product AB is at most $f(d, q)$.*

Note that, by similar arguments, the sum of two conformal matrices over $\mathbb{F}_q$, both of twin-width at most d, has twin-width at most $f'(d, q)$, for some computable function f'.

We now consider the problem from an algorithmic point of view. From a computational perspective, the data structures used to encode matrices over a finite field are crucial. Encoding matrices as bipartite binary structures allows using the machinery developed for (ordered) graphs. In this setting, natural witnesses for twin-width boundedness are *d-sequences* (or, *contraction sequences*), which we mentioned earlier when discussing first-order model checking complexity on classes with bounded twin-width (see Section 2.2 for a formal definition). A naive implementation of the algorithm presented in [4, Theorem 2] runs in time $\exp(\exp(\mathcal{O}(d^2 \log d)))n^3$, and outputs a $2^{\mathcal{O}(d^4)}$-sequence if the twin-width is indeed at most d. We show how to bring the dependence in n down to $\mathcal{O}_d(n^2 \log n)$ (Theorem 8).

Gajarský et al. [18], building on Pilipczuk et al. [26], showed that given an n-vertex graph (or binary structure) G with a d-sequence and a first-order formula $\varphi(x_1, \dots, x_k)$, one can compute in time $\mathcal{O}_{d,\varphi}(n^{1+\varepsilon})$ (resp. $\mathcal{O}_{d,\varphi}(n)$) a data structure that answers for any query $v_1, \dots, v_k \in V(G)$ whether $G \models \varphi(v_1, \dots, v_k)$ in time $\mathcal{O}_{d,\varphi}(1/\varepsilon)$ (resp. $\mathcal{O}_{d,\varphi}(\log \log n)$). This result can be extended to FO+MOD. Then, the squaring operation can be performed by means of a simple interpretation, which gives a near-linear representation of M^2 (in the domain size, that is, sublinear in the number of matrix entries) where entries can be queried in constant time.

We thus obtain an algorithm that takes two matrices of bounded twin-width (*without* witnesses) and outputs their product in quasilinear time in the number of entries.

▶ **Theorem 5.** *Given two $n \times n$ matrices A and B over $\mathbb{F}_q$, both of twin-width at most d, there is an algorithm to compute their product AB in time $\mathcal{O}_{d,q}(n^2 \log n)$.*

However, this algorithm is not practical due to the acute dependence on the twin-width bound. We thus place ourselves in a setting where inputs are already in compact form (witnessing low twin-width). The use of an adapted internal representation is a classical technique of digital computing (Fourier transform, redundant representation of numbers, etc.). Likewise, it appears that convenient representations of matrices of bounded twin-width for matrix computations are *twin-decompositions* [3, 6]. Informally, a twin-decomposition is a tree whose leaves are bijectively mapped to the domain of the structure (here, to the row and column indices), and internal nodes are ordered and naturally correspond to contractions. The binary relations (here, the entries) are encoded by additional edges joining pairs of nodes of the tree, and respecting some specific rules. Every binary structure with bounded twin-width has a twin-decomposition with linearly many extra edges, hence the twin-decomposition forms a degenerate graph. The *width* of the twin-decomposition is related to this degeneracy (see Section 2 for precise definitions). Notice that a twin-decomposition of constant width takes quasilinear space to describe a set of binary relations with possibly quadratically many pairs. We show that a twin-decomposition can be computed from a contraction sequence in quadratic time and observe that, conversely, a contraction sequence can be computed from a twin-decomposition in linear time.

Our last contribution is an *ad hoc* efficient matrix multiplication algorithm for matrices over $\mathbb{F}_p$ with bounded twin-width, which we state here in the case of matrices over $\mathbb{F}_2$.

▶ **Theorem 6** (Theorem 7+[26]). *Let A and B be two $n \times n$ matrices over $\mathbb{F}_2$ given in the form of twin-decompositions of width at most d. For every $\varepsilon > 0$, there is a $4^{d+o(d)}n^{1+\varepsilon}$-time algorithm that outputs a twin-decomposition of the product AB of width $2^{d+o(d)}$ and a data structure of size $2^{d+o(d)}n^{1+\varepsilon}$ such that querying an entry of AB takes time $\mathcal{O}(1/\varepsilon)$.*

The following (Theorem 7) is our main technical contribution: an algorithm computing a twin-decomposition of the square M^2 of a matrix M from a twin-decomposition of M, which extends to a matrix multiplication algorithm for matrices each represented by a twin-decomposition.

▶ **Theorem 7.** *Let A and B be two $n \times n$ matrices over $\mathbb{F}_2$ given in the form of twin-decompositions of width at most d. There is a $4^{d+o(d)}n$-time algorithm that outputs a twin-decomposition of the product AB of width $2^{d+o(d)}$.*

Contrary to Theorem 5 that hides a non-elementary dependence in the twin-width bound, the dependence in d given by Theorem 7 is single-exponential. In addition, since Theorem 7 assumes that a twin-decomposition is given in input, the running time is sublinear in the number of matrix entries, n^2 (as opposed to quasilinear for Theorem 5).

The entries of AB can then be queried in time essentially the height of the twin-decomposition, which can be made logarithmic. However, by computing the data structure introduced by Pilipczuk et al. [26] in $\mathcal{O}(d'n^{1+\varepsilon})$ time and space where d' upperbounds the width of a twin-decomposition of the matrix, the entry queries can be performed in $\mathcal{O}(1/\varepsilon)$ time. Theorems 6 and 7 carry over on any finite field $\mathbb{F}_q$ with running time $q^{2d+o(d)}n$ and $2^{\mathcal{O}_q(d)}n^{1+\varepsilon}$, respectively.

An intriguing question concerns the existence of such results over infinite fields (starting with $\mathbb{Q}$). We do not have a direct definition of twin-width of matrices over $\mathbb{Q}$ based on contraction sequences. However linear-minor freeness naturally carries to infinite fields, and thus, it is natural to consider that a class of matrices over $\mathbb{Q}$ has bounded twin-width if its closure under linear minors is not the set of all matrices. This can be equivalently stated via the notion of *grid rank* of a matrix M, i.e., the largest k for which there is a $k \times k$ subdivision of M in which every block has rank at least k. Note that if a matrix has grid rank k, then any linear minor has grid rank at most k. Indeed, one can even show that a class of matrices has bounded grid rank if and only if it does not contain some matrix as a linear minor. We believe that computing the product of two matrices over $\mathbb{Q}$ with bounded grid rank should be done in almost quadratic time, however, we lack a structural decomposition as in the finite field case.

There is a vast literature on computing matrix multiplication, or other natural primitives of linear algebra, on classes of structured matrices. We give a few references on rank-structured matrices (see for instance [15, 21, 7, 30, 29]) and matrices of bounded treewidth.

A square matrix has quasiseparable order s if all its submatrices that are completely above the main diagonal, or completely below it, have rank at most s. Note that on adjacency matrices this is equivalent to the *linear rank-width* parameter (a dense analogue of pathwidth). Pernet [24] shows that multiplying two $n \times n$ matrices with quasiseparable order s can be done in time $\mathcal{O}(s^{\omega-2}n^2)$, where ω is the exponent of matrix multiplication, or $\mathcal{O}(s^3 n)$ if the matrices are given in a suitable compact form [25]. The closely-related semiseparable matrices also have efficient multiplication algorithms [31]. So-called $\mathscr{H}$-matrices (for hierarchical matrices) and $\mathscr{H}^2$-matrices admit almost linear-time algorithms for vector-matrix multiplication [7].

One can naturally extend the treewidth graph parameter to 0,1-matrices M by considering the treewidth of the bipartite graph whose biadjacency matrix is M. Fomin et al. [17] show how to compute the determinant, the rank, and to solve a linear system defined by an $n \times n$ matrix of treewidth k, in time $k^{\mathcal{O}(1)}n$. It was recently shown by Dong et al. [13] how to solve linear programs in expected almost linear-time on matrices of bounded treewidth.

2 Preliminaries

We denote by $[i, j]$ the set of integers $\{i, i+1, \ldots, j-1, j\}$, and $[i]$ is a short-hand for $[1, i]$. We use the standard graph-theoretic notations: $V(G)$, $E(G)$, $N_G[S]$, $N_G(S)$ respectively denote the vertex set, edge set, closed neighborhood of S, open neighborhood of S. Given a matrix M, we may interchangeably denote by $M_{x,y}$ or $M[x, y]$ the entry of M at row x and column y.

2.1 Binary structures and matrices

A relational *signature* Σ is a finite set of relation symbols R, each with a specified arity $r \in \mathbb{N}$. A *Σ-structure* $\mathbf{A}$ is defined by a set A (the *domain* of $\mathbf{A}$) together with a relation $R^{\mathbf{A}} \subseteq A^r$ for each relation symbol $R \in \Sigma$ with arity r. The syntax and semantics of first-order formulas over Σ, or *Σ-formulas* for brevity, are defined as usual. A *binary structure* is a Σ-structure such that every relation symbol of Σ has arity at most 2. An *ordered binary structure* is a structure $\mathbf{A}$ over a signature Σ consisting of unary and binary relation symbols which includes the symbol $<$, defining in $\mathbf{A}$ a total order on the domain of $\mathbf{A}$.

A matrix M over a finite alphabet A with rows R and columns C is viewed as an ordered binary structure with domain $R \uplus C$, equipped with the following relations:

- a unary relation R interpreted as the set of rows,
- an antisymmetric binary relation $<$ which defines a total order on $R \uplus C$, extending the total orders on the rows and columns of M in such a way that the rows precede the columns,
- one binary relation E_a, for each $a \in A$, where $E_a(r, c)$ holds if and only if r is a row, c is not (hence is a column), and a is the entry of M at row r and column c.

2.2 Contraction sequences and twin-width

A *trigraph* G has vertex set $V(G)$, (black) edge set $E(G)$, and red edge set $R(G)$, with $E(G)$ and $R(G)$ being disjoint. The *set of neighbors* $N_G(v)$ of a vertex v in a trigraph G consists of all the vertices adjacent to v by a black or red edge. A d-trigraph is a trigraph G such that the *red graph* $(V(G), R(G))$ has degree at most d. In that case, we also say that the trigraph has *red degree* at most d. A (vertex) *contraction* or *identification* in a trigraph G consists of merging two (non-necessarily adjacent) vertices u and v into a single vertex z, and updating the edges of G in the following way. Every vertex of the symmetric difference $N_G(u) \triangle N_G(v)$ is linked to z by a red edge. Every vertex x of the intersection $N_G(u) \cap N_G(v)$ is linked to z by a black edge if both $ux \in E(G)$ and $vx \in E(G)$, and by a red edge otherwise. The rest of the edges (not incident to u or v) remain unchanged. We insist that the vertices u and v (together with the edges incident to these vertices) are removed from the trigraph.

A *d-sequence* (or *contraction sequence*) is a sequence of d-trigraphs $G_n, G_{n-1}, \dots, G_1$, where $G_n = G$, $G_1 = K_1$ is the graph on a single vertex, and G_{i-1} is obtained from G_i by performing a single contraction of two (non-necessarily adjacent) vertices. We observe that G_i has precisely i vertices, for every $i \in [n]$. The twin-width of G, denoted by $\text{tww}(G)$, is the minimum integer d such that G admits a d-sequence. See Figure 1 for an illustration.

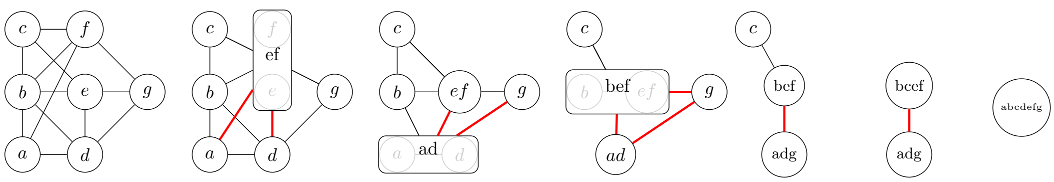

Figure 1 A 2-sequence witnessing that the initial graph has twin-width at most 2.

Twin-width can be generalized from graphs to binary structures in some (functionally) equivalent ways [5, 2]. Here we choose the following definition.

On general binary structures, red edges exist between two vertices $x, y \in V(G_i)$ whenever there are up to four vertices $u \neq v, u' \neq v' \in V(G)$ such that u and u' (which might be the same vertex) were contracted (together with possibly other vertices) to form x, similarly

v and v' were contracted to form y, and the atomic types of (u, v) and of (u', v'), or of (v, u) and of (v', u'), are distinct. If instead, all such pairs (u, v) have the same atomic type, this shared atomic type labels the edge xy in G_i. Contraction sequences, d-sequences, and twin-width are then similarly defined.

In particular, we now have a definition of twin-width for the matrices of Section 2.1.

2.3 Matrix divisions

We will often denote by R (resp. C) the sets of row (resp. column) indices of a matrix M. For $X \subseteq R$ and $Y \subseteq C$, we denote by $M[X, Y]$ the submatrix of M consisting of the entries at rows in X and columns in Y.

A (k, ℓ)*-division* of a matrix M is a pair of partitions $(\mathcal{R} = \{R_1, \dots, R_k\}, \mathcal{C} = \{C_1, \dots, C_\ell\})$ of R and C, respectively, such that every R_i corresponds to consecutive rows, and every C_j, to consecutive columns. A k*-division* is a (k, k)-division. We may call a set of consecutive row or column indices an *interval.* The *grid number*, *mixed number*, *grid rank*, respectively, of a matrix is the largest integer k such that M has a k-division $(\{R_1, \dots, R_k\}, \{C_1, \dots, C_k\})$ for which, for every $i, j \in [k]$, respectively,

- $M[R_i, C_j]$ has a non-zero entry,
- $M[R_i, C_j]$ has at least two distinct rows and at least two distinct columns,
- $M[R_i, C_j]$ has at least k distinct rows or at least k distinct columns.

The divisions are called k*-grid minor*, k*-mixed minor*, and *rank-*k *division*, respectively. An *interval minor* of a matrix M is a matrix N with k rows and ℓ columns such that M has a (k, ℓ)-division $(\{R_1, \dots, R_k\}, \{C_1, \dots, C_\ell\})$ such that for every $i \in [k]$ and every $j \in [\ell]$, $N_{i,j}$ is an entry of $M[R_i, C_j]$. We call k-Grid Minor, k-Mixed Minor, Rank-k Division, Interval Minor Containment, respectively, the computational problems of deciding if an input matrix has grid number at least k, mixed number at least k, grid rank at least k, and if a matrix is an interval minor of another matrix.

2.4 Computing contraction sequences

Efficiently (in polynomial time, FPT time, or even slice-wise polynomial time) approximating the twin-width of a binary structure remains a challenging open question. However, such an algorithm is known for totally ordered binary structures, or matrices over a finite alphabet [4]. Following [4], we bring the complexity from a cubic down to an almost quadratic dependence in the number of rows and columns.

▶ **Theorem 8.** *Given an $n \times n$ symmetric matrix M over a finite alphabet A, and an integer d, there is an algorithm running in time $\mathcal{O}_{d,|A|}(n^2 \log n)$ that*

- *either correctly reports that the twin-width of M is larger than d,*
- *or outputs an $\exp(\exp(\exp(\mathcal{O}_{|A|}(d^4))))$-sequence for M.*

2.5 Twin-decompositions

A twin-decomposition of a graph G also uses the framework of a rooted carving decomposition, i.e., a rooted binary tree whose leaves are in one-to-one correspondence with the vertices of G. In the case of twin-decompositions though, the internal nodes of the rooted binary tree are totally ordered and the *width* is quite different from how carving-width is defined.

A rooted binary tree with $n - 1$ internal nodes bijectively labeled by ℓ on $[n - 1]$ is said *ranked* if whenever u and v are two distinct internal nodes such that v is a descendant of u, then $\ell(u) < \ell(v)$ holds. By convention, we then decide that all the leaves are labeled $+\infty$

(or equivalently n). For every $i \in [n]$, the *i-th border* of a ranked tree $\mathcal{T}$ with $n-1$ internal nodes is the set of maximal rooted subtrees whose roots have label at least i. We denote by $B_i(\mathcal{T})$ the i-th border of $\mathcal{T}$. It can be easily shown (using that $\mathcal{T}$ is a ranked tree) that the i-th border of $\mathcal{T}$ consists of exactly i subtrees. We denote by $r(\mathcal{T})$ the root of $\mathcal{T}$.

A *twin-decomposition* of an n-vertex graph G is a pair $(\mathcal{T}, \mathcal{B})$ where

(a) $\mathcal{T}$ is a rooted binary **t**ree, ranked by ℓ on $[n-1]$, whose leaves are in one-to-one correspondence with $V(G)$, and

(b) $\mathcal{B}$ (for **b**icliques) is a set of edges over $V(\mathcal{T})$ (disjoint from edge set of $\mathcal{T}$), such that:

1. $\mathcal{B}$ partitions the edge set of G, where an edge between $u, v \in V(\mathcal{T})$ is interpreted as the biclique of G linking every leaf in the subtree of $\mathcal{T}$ rooted at u, to every leaf in the subtree of $\mathcal{T}$ rooted at v, and
2. No edge of $\mathcal{B}$ *crosses* an i-th border, i.e., links a node in a subtree $\mathcal{T}'$ of $B_i(\mathcal{T})$ but *not* the root of $\mathcal{T}'$ to a vertex outside every subtree of $B_i(\mathcal{T})$.

The width of the twin-decomposition $(\mathcal{T}, \mathcal{B})$ is again defined as the maximum red degree among every vertex of every trigraph G_i (as previously defined). See Figure 2 for an illustration of a twin-decomposition corresponding to a particular contraction sequence.

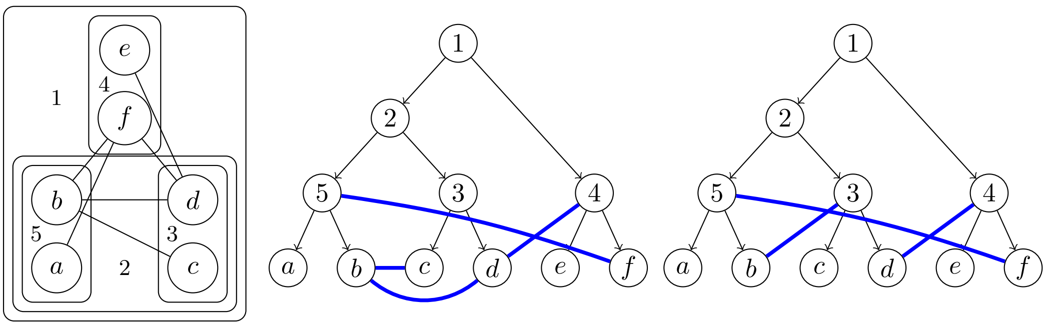

Figure 2 Left: a graph G with a contraction sequence (or partition sequence), where trigraph G_i is obtained after performing the contraction labeled i. Center: the twin-decomposition corresponding to this contraction sequence, with the edges of $\mathcal{B}$ in blue. Right: a ranked tree $\mathcal{T}$ and a partition $\mathcal{B}$ of the edges of G that does *not* make for a twin-decomposition, since the edge $b3$ crosses $B_5(\mathcal{T})$ (and $B_4(\mathcal{T})$).

For our intended purposes in this extended abstract, we only described twin-decompositions for graphs but they readily generalize to binary structures (see long version).

2.6 Computing a twin-decomposition from a contraction sequence

There is an easy linear (in the input size, i.e., in the number of edges) algorithm that, given a d-sequence, computes a corresponding twin-decomposition (of same width d). The *list of triples* of a contraction sequence $G_n, \ldots, G_i, \ldots, G_1$ is $(u_n, v_n, z_n), \ldots, (u_i, v_i, z_i), \ldots, (u_2, v_2, z_2)$ such that the contraction of u_i and v_i in G_i into a new vertex z_i results in G_{i-1}.

▶ **Theorem 9.** *A twin-decomposition of width d of an n-vertex graph G given with the list of triples of a d-sequence can be computed in time $\mathcal{O}(n^2)$.*

A consequence of the second paper of the series is that, given twin-decompositions of bounded width, one can find twin-decompositions of (larger) bounded width, where in addition the tree $\mathcal{T}$ has logarithmic depth.

▶ **Theorem 10** ([2, see Lemma 23 and Proposition 22]). *For every integer d, there is a larger integer D such that every n-vertex graph of twin-width d admits a twin-decomposition of width at most D and depth $\mathcal{O}_d(\log n)$.*

Given a twin-decomposition $(\mathcal{T}, \mathcal{B})$ of G with width d and depth h, the presence of an edge between two vertices of G can be decided in time $\mathcal{O}(dh)$. This yields a linear-space representation of G with edge queries in logarithmic time. Pilipczuk et al further show that:

▶ **Theorem 11** ([26]). *Given a twin-decomposition of width d of an n-vertex graph G, and any $\varepsilon > 0$, there is a data structure of size $\mathcal{O}(dn^{1+\varepsilon})$, computable in time $\mathcal{O}(dn^{1+\varepsilon})$ that supports edge queries of G in time $\mathcal{O}(1/\varepsilon)$.*

All the results mentioned in this section extend from graphs to binary structures.

2.7 Parameterized complexity of model checking

First-order (FO) matrix model checking asks, given a matrix M (or a totally ordered binary structure $\mathcal{S}$) and a first-order sentence φ, if $M \models \varphi$ holds, that is, if φ is true in M. FO model checking is fixed-parameter tractable (FPT) on a matrix class $\mathcal{M}$, with respect to the sentence size and the input matrix, if there exists a constant c and a computable function f, such that $M \models \varphi$ can be decided in time $f(|\varphi|)\,(m+n)^c$, for every $n \times m$ matrix $M \in \mathcal{M}$ and FO sentence φ.

FO model checking of general (unordered) graphs is AW[$*$]-complete [14], and thus very unlikely to be FPT. Indeed FPT $\neq$ AW[$*$] is a much weaker assumption than the already widely-believed Exponential Time Hypothesis [22], and if false, would in particular imply the existence of a subexponential algorithm solving 3-SAT. FO model checking of general binary structures of bounded twin-width given with an $\mathcal{O}(1)$-sequence can even be solved in linear FPT time $f(|\varphi|)\,|U|$, where U is the domain of the structure [5].

Gajarský et al. [18] reproved that result using a different, and more standard formalism. Building on Theorem 11, they also presented an algorithm that inputs a binary structure given with an $\mathcal{O}(1)$-sequence and a formula with some free variables, and after some preprocessing in near-linear time, can answer queries of the form *does the given tuple satisfy the formula in the structure* in constant time.

▶ **Theorem 12** ([18]). *For every $\varepsilon > 0$, given a binary Σ-structure $\mathbf{A}$ on a domain of size n, a d-sequence of $\mathbf{A}$, and a first-order Σ-formula $\varphi(x_1, \ldots, x_k)$, there is a data structure computable in time $\mathcal{O}_{d,\varphi}(n^{1+\varepsilon})$ that given any query of the form $v_1, \ldots, v_k \in A$ reports in time $\mathcal{O}_{d,\varphi}(1/\varepsilon)$ whether $\mathbf{A} \models \varphi(v_1, \ldots, v_k)$ holds.*

In classes of ordered binary structures or matrices, one need not require that the contraction sequence is given in input. Indeed there is an FPT approximation algorithm, that takes a matrix M of twin-width d, and outputs a $g(d)$-sequence of M in time $h(d)\,|M|^{\mathcal{O}(1)}$ [4]. Hence, FO matrix model checking can be solved in FPT time $f(|\varphi|)\,|M|^{\mathcal{O}(1)}$ [4] in classes of bounded twin-width. We observe that this algorithm extends to FO+MOD model checking.

▶ **Theorem 13** (follows with some small adjustments from [5, Section 7]). *Given a d-sequence of a binary structure $\mathcal{S}$, and a prenex* FO+MOD*-sentence φ of quantifier rank ℓ, one can decide $\mathcal{S} \models \varphi$ in time $f(\ell, d)n$ for some computable non-elementary function f.*

The same applies to the fact that bounded twin-width is preserved by transductions.

▶ **Theorem 14** (similarly follows from [5, Section 8]). *Let $\mathcal{C}$ be a set of binary structures with bounded twin-width, and* T *be an* FO+MOD*-transduction. Then* $\mathsf{T}(\mathcal{C})$ *has bounded twin-width.*

3 Equivalence of bounded twin-width and linear-minor freeness

Alternatively to the definition given in introduction, linear and parity minors can be characterized by matrix divisions. A *parity minor* of a matrix M is any $n \times m$ matrix N obtained by summing up every cell of an (n,m)-division of a submatrix of M. We denote that by $N \leqslant_{\mathrm{pm}} M$. See Figure 3 for an illustration.

$$M = \begin{bmatrix} 1&0&1&1&0&0&1&1\\ 0&1&1&0&0&1&0&1\\ 0&0&0&0&0&0&0&1\\ 1&1&0&0&1&0&1&1\\ 1&0&0&1&1&0&1&0\\ 0&0&1&1&1&1&0&0\\ 0&0&1&1&0&0&0&1 \end{bmatrix} \qquad \left[\begin{array}{cc|cccc|cc} 1&0&1&1&0&0&1&1\\ \hline 0&1&1&0&0&1&0&1\\ 0&0&0&0&0&0&0&1\\ \hline 1&1&0&0&1&0&1&1\\ 1&0&0&1&1&0&1&0\\ \hline 0&0&1&1&1&1&0&0\\ 0&0&1&1&0&0&0&1 \end{array}\right] \qquad N = \begin{bmatrix} 1&1&0\\ 0&0&1\\ 0&0&0\\ 0&1&1 \end{bmatrix}$$

Figure 3 A parity minor, equivalently linear minor, N of a matrix M over $\mathbb{F}_2$. In the middle, the deleted rows and columns of M are in light gray, while the $(4,3)$-division of the remaining submatrix giving rise to N is represented with solid black lines.

Let us call $\mathbb{F}$-*weighting of* M any mapping $w : \mathrm{rows}(M) \cup \mathrm{cols}(M) \to \mathbb{F}$. Equivalently a linear minor of a matrix M over a finite field $\mathbb{F}$ is any $n \times m$ matrix N obtained from an $\mathbb{F}$-weighting w and (n,m)-division $\mathcal{D} = (\{R_1, \ldots, R_n\}, \{C_1, \ldots, C_m\})$ of M by replacing every cell $M[R_i, C_j]$ of $\mathcal{D}$ by the single entry $\sum_{r \in R_i, c \in C_j} w(r)w(c)M_{r,c}$, that is, setting $N_{i,j} = \sum_{r \in R_i, c \in C_j} w(r)w(c)M_{r,c}$. We denote that by $N \leqslant_{\mathrm{lm}} M$.

▶ **Observation 15.** *Over $\mathbb{F}_2$, parity minors and linear minors coincide.*

The *linear-minor closure* of $\mathcal{M}$, denoted by $\mathrm{Clos}_{\mathrm{lm}}(\mathcal{M})$, is the matrix class of all matrices N which are linear minors of some M in $\mathcal{M}$. We say that $\mathcal{M}$ is *parity-minor free* (resp. *linear-minor free*) if its parity-minor closure (resp. linear-minor closure) is not the set of all $\mathbb{F}$-matrices. We first show that bounded twin-width matrix classes over finite fields are linear-minor free, and in particular parity-minor free.

▶ **Lemma 16.** *Let $\mathbb{F}$ be a finite field. For every matrix class $\mathcal{M}$ over $\mathbb{F}$ of bounded twin-width, $\mathcal{M}$ is linear-minor free.*

Proof sketch. We can express the set of linear minors of a matrix by a FO+MOD-transduction. Indeed the modulo-counting quantifiers allow one to write a first-order formula summing up the entries in a given contiguous submatrix. Thus by Theorem 14, $\mathrm{Clos}_{\mathrm{lm}}(\mathcal{M})$ has bounded twin-width, and cannot be the set of all $\mathbb{F}$-matrices. ◀

For the converse, we will need the notion of *rank Latin divisions* previously introduced [4]. For any integers $k \geqslant 2$ and $d \geqslant 1$, a *rank-k Latin d-division* of a $kd^2 \times kd^2$ matrix M is a regular d-division $\mathcal{D}$ of M that can be refined into a regular d^2-division $((R_1, \ldots, R_{d^2}), (C_1, \ldots, C_{d^2}))$ such that

- $\forall i \in [d^2]$, $M[R_i, C_j]$ is constant for every $j \in [d^2]$ but one j_i for which it has rank k,
- $\forall j \in [d^2]$, $M[R_i, C_j]$ is constant for every $i \in [d^2]$ but one i_j for which it has rank k,
- and every cell of $\mathcal{D}$ contains exactly one $M[R_i, C_j]$ with rank k.

It was previously shown that matrix classes with unbounded twin-width contain matrices with rank-k Latin d-divisions for arbitrarily large values of k and d. For our purpose we will only need $k = 2$ and d diverging.

▶ **Lemma 17** ([4]). *Let $\mathcal{M}$ be a matrix class of unbounded twin-width over a finite field. Then for every d, there is a matrix $M \in \mathcal{M}$ with a rank-2 Latin d-division.*

Equipped with that technical lemma, we can show the following (see long version).

▶ **Lemma 18.** *Let $\mathbb{F}$ be a finite field and $\mathcal{M}$ be a matrix class of $\mathbb{F}$-matrices. If $\mathcal{M}$ is linear-minor free then it has bounded twin-width.*

4 Fixed-parameter algorithms for matrix division problems

We show how to use Theorem 13 and the approximation algorithm of matrix twin-width [4, Theorem 2], to decide matrix problems involving a division (of a submatrix) with some FO+MOD-definable properties in fixed-parameter time. This is based on a win-win argument generalizing the algorithmic scheme of Guillemot and Marx [20] to solve Permutation Pattern, and somewhat resembling the bidimensionality technique [16]. It allows for instance to detect a k-grid minor or a k-mixed minor in an $n \times n$ matrix, or to decide if a $k \times k$ matrix is a parity or linear minor of an $n \times n$ matrix in time $f(k)n^{\mathcal{O}(1)}$.

▶ **Theorem 19** ([4]). *Given as input an $n \times m$ matrix M over a fixed finite field $\mathbb{F}$, and an integer k, there is an $f(k)(n+m)^{\mathcal{O}(1)}$-time algorithm which returns*
- *either a rank-k Latin division of a submatrix of M,*
- *or a contraction sequence certifying that $\operatorname{tww}(M) \leqslant g(k)$.*

where f and g are computable functions.

A parameterized problem Π, taking as input a matrix over a fixed finite field and a non-negative integer, is FO+MOD-*definable* if, there is a computable function f, and for every non-negative integer k, there is a FO+MOD[τ] sentence $\varphi_{\Pi,k}$ of size $f(k)$ such that (M,k) is a YES-instance of Π if and only if $M \models \varphi_{\Pi,k}$. A parameterized problem Π, taking as input a matrix M and a non-negative integer k, is said *rank-bidimensional* if, for some computable functions f and g, the existence of a rank-$f(k)$ Latin division in M (i.e., a submatrix of M has a rank-$f(k)$ Latin division) permits to decide Π in time $g(k)|M|^{\mathcal{O}(1)}$. We can now state the main observation of this section.

▶ **Theorem 20.** *Every* FO+MOD*-definable rank-bidimensional problem is in* FPT*.*

Proof. Let Π be a rank-bidimensional problem, with computable functions f and g. Let (M,k) be an input of Π. We run the algorithm of Theorem 19 with parameter $f(k)$. In time $f'(k)|M|^{\mathcal{O}(1)}$, this either yields a rank-$f(k)$ division of a submatrix of M, and we can decide (M,k) in further time $g(k)|M|^{\mathcal{O}(1)}$ (since Π is rank-bidimensional), or a $g'(k)$-sequence of M, and we can conclude by Theorem 13 (since Π is FO+MOD-definable). ◀

As a corollary of Theorem 20, we obtain for instance that deciding if N is a parity minor of M is fixed-parameter tractable in the size of N. Indeed we show (in the long version) how to express the parity minor containment with FO+MOD sentences.

▶ **Theorem 21.** *Let N be a $k \times k$ matrix, and M be an $n \times m$ matrix, both over $\mathbb{F}_2$. One can decide $N \leqslant_{pm} M$ in time $f(k)(n+m)^{\mathcal{O}(1)}$ for some computable function f.*

Similarly we can invoke Theorem 20 for the following problems. In the following theorem, Linear Minor Containment is the problem of deciding $N \leqslant_{\mathrm{lm}} M$, given two matrices N, M over a fixed finite field, parameterized by $|N|$.

▶ **Theorem 22.** *Linear Minor Containment, Interval Minor Containment, k-Grid Minor, k-Mixed Minor, Rank-k Division, Permutation Pattern are in* FPT*.*

5 Products of bounded twin-width matrices over a finite field

First we show that if a matrix class $\mathcal{M}$ over a finite field has bounded twin-width, then so does its set of squares $\mathcal{M}^2 = \{AB : \text{A and B are two conformal matrices of } \mathcal{M}\}$.

▶ **Theorem 23.** *There is a function $f : \mathbb{N}^3 \to \mathbb{N}$ such that for every conformal matrices A and B over $\mathbb{F}_q$, The product AB (over $\mathbb{F}_q$) has twin-width at most $f(\text{tww}(A), \text{tww}(B), q)$.*

Proof. Since $\begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix} = \begin{pmatrix} AB & 0 \\ 0 & BA \end{pmatrix}$ and

$$\text{tww}\left(\begin{pmatrix} 0 & X \\ Y & 0 \end{pmatrix}\right) = \text{tww}\left(\begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}\right) = \max(\text{tww}(X), \text{tww}(Y), 2),$$

we shall just prove that there is a function g such that M^2 has twin-width at most $g(\text{tww}(M))$, for every square matrix M over $\mathbb{F}_q$. There is a simple FO+MOD-interpretation S such that $\mathsf{S}(M) = M^2$. Indeed one can keep the relations R and $\prec$ as in M, and express $E_i^{M^2}(x, y)$ as

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

where $\tilde{i}$ is the element of $\mathbb{F}_q$ corresponding to relation E_i. The expression $\tilde{j}\tilde{k}$ is a product in $\mathbb{F}_q$, while $a(j, k) \cdot (\tilde{j}\tilde{k})$ is the sum of $a(j, k)$ occurrences of $\tilde{j}\tilde{k}$. As every element of $(\mathbb{F}_q, +)$ has an order dividing q, it is enough to count the number of pairs $(\tilde{j}, \tilde{k}) = (M_{x,z}, M_{z,y})$ modulo q, which the formula does. We finally invoke Theorem 14 to conclude that $\text{tww}(M^2)$ is bounded by a function of $\text{tww}(M)$ and q. ◀

▶ **Theorem 24.** *Let q be a prime power, and d be a natural. Let A, B be two $n \times n$ matrices over $\mathbb{F}_q$, both of twin-width at most d. One can compute the product AB in time $\mathcal{O}_{d,q}(n^2 \log n)$.*

Proof. By Theorem 8, we compute an $\mathcal{O}_{d,q}(1)$-sequence for

$$M = \begin{pmatrix} 0 & A \\ B & 0 \end{pmatrix}, \text{ in time } \mathcal{O}_{d,q}(n^2 \log n).$$

We conclude either by turning this contraction sequence into a twin-decomposition in time $\mathcal{O}_{d,q}(n^2)$, by Theorem 9, and invoking the upcoming practical matrix squaring of Theorem 25, or by combining the FO+MOD-interpretation of Theorem 23 (and Theorem 14) with the efficient algorithm of Gajarský et al. [18] (see Theorem 12) to compute the interpretations of bounded twin-width structures. We can finally read off the top-left block AB in M^2 in time $\mathcal{O}_{d,q}(n^2)$.

If we chose the former approach, we now have a twin-decomposition $(\mathcal{T}, \mathcal{B})$ of AB. We can initialize an $n \times n$ matrix to all 0 entries, and for each edge of $\mathcal{B}$ labeled ℓ, fill the corresponding entries with ℓ. This takes quadratic time since we access each matrix entry at most once. If we instead went with the latter approach, we shall simply make $(q-1)n^2$ constant-time queries to build AB, $q - 1$ for each entry of AB. ◀

The most technical contribution of the paper is the following, stated and shown in the language of edge-colored graphs in the long version. For every prime power $q = p^\alpha$, we let $m(q)$ denote the cost of basic arithmetic computations in $\mathbb{F}_q$; here only addition, subtraction and multiplication are needed.

▶ **Theorem 25.** *For every prime power $q \geqslant 2$, there is an $\mathcal{O}(m(q)d^2q^{2d}n)$-time algorithm that, given a twin-decomposition $(\mathcal{T}, \mathcal{B})$ of width d of a symmetric $n \times n$ matrix M over $\mathbb{F}_q$, outputs a twin-decomposition of width $\mathcal{O}(d^2q^d)$ of M^2.*

Note that in practice, if $q = p^\alpha$ with p prime, one can choose $m(q) = \mathcal{O}(\log_p(q)^2 \log(p)^2)$ (see for example [23, Table 2.8]). The comparatively good dependence on twin-width is achieved by working directly on the twin-decomposition and problem-specific insights. The algorithm follows three steps. In the first step, a contraction sequence is computed for M^2 as a refinement of the one for M. The contraction sequence naturally yields the tree $\mathcal{T}'$ of a twin-decomposition $(\mathcal{T}', \mathcal{B}')$, where one shall now find $\mathcal{B}'$. The next step aims to obtain a set $\mathcal{B}_1$ encoding M^2 (but not yet a twin-decomposition). This set is then cleaned in the third and last step into a twin-decomposition $(\mathcal{T}', \mathcal{B}')$ of M^2 of width at most exponential in the width of the twin-decomposition $(\mathcal{T}, \mathcal{B})$.

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

3 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`.

4 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`.

5 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`.

6 Édouard Bonnet, Jaroslav Nesetril, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations. *CoRR*, abs/2102.06880, 2021. `arXiv:2102.06880`.

7 Steffen Börm, Lars Grasedyck, and Wolfgang Hackbusch. Introduction to hierarchical matrices with applications. *Engineering analysis with boundary elements*, 27(5):405–422, 2003.

8 S. Braunfeld and M.C. Laskowski. Existential characterizations of monadic NIP, 2022. `doi:10.48550/arXiv.2209.05120`.

9 Julia Chuzhoy and Zihan Tan. Towards tight(er) bounds for the excluded grid theorem. *J. Comb. Theory, Ser. B*, 146:219–265, 2021. `doi:10.1016/j.jctb.2020.09.010`.

10 B. Courcelle. Graph rewriting: an algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, chapter 5, pages 142–193. Elsevier, Amsterdam, 1990.

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

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

13 Sally Dong, Yin Tat Lee, and Guanghao Ye. A nearly-linear time algorithm for linear programs with small treewidth: a multiscale representation of robust central path. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1784–1797. ACM, 2021. doi:10.1145/3406325.3451056.

14 Rodney G. Downey, Michael R. Fellows, and Udayan Taylor. The parameterized complexity of relational database queries and an improved characterization of W[1]. In Douglas S. Bridges, Cristian S. Calude, Jeremy Gibbons, Steve Reeves, and Ian H. Witten, editors, *First Conference of the Centre for Discrete Mathematics and Theoretical Computer Science, DMTCS 1996, Auckland, New Zealand, December, 9-13, 1996*, pages 194–213. Springer-Verlag, Singapore, 1996.

15 Yuli Eidelman and Israel Gohberg. On a new class of structured matrices. *Integral Equations and Operator Theory*, 34(3):293–324, 1999.

16 Fedor V. Fomin, Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensionality. In *Encyclopedia of Algorithms*, pages 203–207. Springer, 2016. doi:10.1007/978-1-4939-2864-4_47.

17 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, Michal Pilipczuk, and Marcin Wrochna. Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. *ACM Trans. Algorithms*, 14(3):34:1–34:45, 2018. doi:10.1145/3186898.

18 Jakub Gajarský, Michal Pilipczuk, Wojciech Przybyszewski, and Szymon Torunczyk. Twin-width and types. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 123:1–123:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.123.

19 Erich Grädel, Phokion G Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y Vardi, Yde Venema, Scott Weinstein, et al. *Finite Model Theory and its applications*. Springer, 2007.

20 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. doi:10.1137/1.9781611973402.7.

21 Wolfgang Hackbusch. A sparse matrix arithmetic based on H-matrices. part I: introduction to h-matrices. *Computing*, 62(2):89–108, 1999. doi:10.1007/s006070050015.

22 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.

23 Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001. URL: http://www.cacr.math.uwaterloo.ca/hac/.

24 Clément Pernet. Computing with quasiseparable matrices. In Sergei A. Abramov, Eugene V. Zima, and Xiao-Shan Gao, editors, *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016*, pages 389–396. ACM, 2016. doi:10.1145/2930889.2930915.

25 Clément Pernet and Arne Storjohann. Time and space efficient generators for quasiseparable matrices. *J. Symb. Comput.*, 85:224–246, 2018. doi:10.1016/j.jsc.2017.07.010.

26 Michal Pilipczuk, Marek Sokolowski, and Anna Zych-Pawlewicz. Compact representation for matrices of bounded twin-width. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 52:1–52:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.52.

27 Neil Robertson and Paul D. Seymour. Graph minors. I. Excluding a forest. *J. Comb. Theory, Ser. B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5.

28 Neil Robertson and Paul D. Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986. `doi:10.1016/0095-8956(86)90030-4`.

29 Christopher De Sa, Albert Gu, Rohan Puttagunta, Christopher Ré, and Atri Rudra. A two-pronged progress in structured dense matrix vector multiplication. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1060–1079. SIAM, 2018. `doi:10.1137/1.9781611975031.69`.

30 Raf Vandebril, Marc Van Barel, Gene Golub, and Nicola Mastronardi. A bibliography on semiseparable matrices. *Calcolo*, 42(3):249–270, 2005.

31 Jianlin Xia, Shivkumar Chandrasekaran, Ming Gu, and Xiaoye S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numer. Linear Algebra Appl.*, 17(6):953–976, 2010. `doi:10.1002/nla.691`.