Mathematical Programming
https://doi.org/10.1007/s10107-023-02048-x

FULL LENGTH PAPER

Series A

Characterization of matrices with bounded Graver bases and depth parameters and applications to integer programming

Marcin Briański[1] · Martin Koutecký[2] · Daniel Král'[3] · Kristýna Pekárková[3] · Felix Schröder[4]

Received: 5 August 2022 / Accepted: 2 December 2023

Abstract

An intensive line of research on fixed parameter tractability of integer programming is focused on exploiting the relation between the sparsity of a constraint matrix A and the norm of the elements of its Graver basis. In particular, integer programming is fixed parameter tractable when parameterized by the primal tree-depth and the entry complexity of A, and when parameterized by the dual tree-depth and the entry complexity of A; both these parameterization imply that A is sparse, in particular, the number of its non-zero entries is linear in the number of columns or rows, respectively. We study preconditioners transforming a given matrix to a row-equivalent sparse matrix if it exists and provide structural results characterizing the existence of a sparse row-equivalent matrix in terms of the structural properties of the associated column matroid. In particular, our results imply that the ℓ_1-norm of the Graver basis is bounded by a function of the maximum ℓ_1-norm of a circuit of A. We use our results to design a parameterized algorithm that constructs a matrix row-equivalent to an input matrix A that has small primal/dual tree-depth and entry complexity if such a row-equivalent matrix exists. Our results yield parameterized algorithms for integer programming when parameterized by the ℓ_1-norm of the Graver basis of the constraint matrix, when parameterized by the ℓ_1-norm of the circuits of the constraint matrix, when

The first author was partially supported by the Polish National Science Center grant (BEETHOVEN; UMO-2018/31/G/ST1/03718). The second author was partially supported by Charles University project UNCE 24/SCI/008 and by the project 19-27871X of GA ČR. The third author was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 648509). This publication reflects only its authors' view; the ERC Executive Agency is not responsible for any use that may be made of the information it contains. The third and fourth authors were supported by the MUNI Award in Science and Humanities (MUNI/I/1677/2018) of the Grant Agency of Masaryk University.

✉ Kristýna Pekárková
kristyna.pekarkova@mail.muni.cz

Extended author information available on the last page of the article

Published online: 20 January 2024

parameterized by the smallest primal tree-depth and entry complexity of a matrix row-equivalent to the constraint matrix, and when parameterized by the smallest dual tree-depth and entry complexity of a matrix row-equivalent to the constraint matrix.

Keywords Integer programming · Width parameters · Matroids · Graver basis · Tree-depth · Fixed parameter tractability

Mathematics Subject Classification 90C10 · 05B35 · 90C27

1 Introduction

Integer programming is a problem of fundamental importance in combinatorial optimization with many theoretical and practical applications. From the computational complexity point of view, integer programming is very hard: it is one of the 21 problems shown to be NP-complete in the original paper on NP-completeness by Karp [37] and remains NP-complete even when the entries of the constraint matrix are zero and one only. On the positive side, Kannan and Lenstra [35, 45] showed that integer programming is polynomially solvable in fixed dimension, i.e., with a fixed number of variables. Another prominent tractable case is when the constraint matrix is totally unimodular, i.e., all determinants of its submatrices are equal to 0 or ± 1, in which case all vertices of the feasible region are integral and so linear programming algorithms can be applied.

Integer programming (IP) is known to be tractable for instances where the constraint matrix of an input instance enjoys a certain block structure. The two most important cases are the cases of 2-stage IPs due to Hemmecke and Schultz [27], further investigated in particular in [1, 13, 31, 39, 40, 44], and n-fold IPs introduced by De Loera et al. [15] and further investigated in particular in [11, 12, 19, 26, 34, 44]. IPs of this kind appear in various contexts, see e.g. [32, 41, 42, 48]. These (theoretical) tractability results complement well a vast number of empirical results demonstrating tractability of instances with a block structure, e.g. [2–4, 22, 23, 38, 49–51].

There tractability results on IPs with sparse constraint matrices can be unified and generalized using depth and width parameters of graphs derived from constraint matrices. Ganian and Ordyniak [24] initiated this line of study by showing that IPs with bounded primal tree-depth $\mathrm{td}_P(A)$ of a constraint matrix A and bounded coefficients of the constraint matrix A and the right hand side b can be solved efficiently. Levin, Onn and the second author [44] widely generalized this result by showing that IPs with bounded $\|A\|_\infty$ and bounded primal tree-depth $\mathrm{td}_P(A)$ or dual tree-depth $\mathrm{td}_D(A)$ of the constraint matrix A can be solved efficiently; such IPs include 2-stage IPs, n-fold IPs, and their generalizations.

Most of the existing algorithms for IPs assume that the input matrix is already given in its sparse form. This is a substantial drawback as existing algorithms cannot be applied to instances that are not sparse but can be transformed to an equivalent sparse instance. For example, the matrix in the left below, whose dual tree-depth is 5, can be transformed by elementary row operations to the matrix with dual tree-depth 2 given in the right; a formal definition of tree-depth is given in Sect. 2.1, however, just the visual appearance of the two matrices indicates which is likely to be more amenable to algorithmic techniques.

$$\begin{pmatrix} 2&2&1&2&1&3&1\\ 2&1&1&1&2&1&1\\ 2&2&2&2&2&2&1\\ 2&1&1&2&2&1&1\\ 2&2&1&2&1&3&2 \end{pmatrix} \rightarrow \begin{pmatrix} 2&1&0&1&1&2&1\\ 0&1&1&0&0&1&0\\ 1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&0&2&0&1 \end{pmatrix}$$

This transformation is an example of a preconditioner that transforms an instance of integer programming to an equivalent one that is more amenable to existing methods for solving integer programming and whose existence we investigate in this paper.

Preconditioning a problem to make it computationally simpler is a ubiquitous preprocessing step in mathematical programming solvers. An interesting link between matroid theory and preconditioners to sparsity of matrices was exhibited by Chan and Cooper together with the second, third and fourth authors [8, 9]. In particular, they proved the following structural characterization of matrices that are *row-equivalent*, i.e., can be transformed by elementary row operations, to a matrix with small dual tree-depth: a matrix is row-equivalent to one with small dual tree-depth if and only if the column matroid of the matrix has small contraction*-depth (see Theorem 1 below). In this paper, we further explore this uncharted territory by providing a structural characterization of matrices row-equivalent to matrices with small primal tree-depth, designing efficient algorithms for finding preconditioners with respect to both primal and dual tree-depth, and relating complexity of circuits and Graver basis of constraint matrices.

1.1 Our contribution

We now describe the results presented in this paper in detail. We opted not to interrupt the presentation of our results with various notions, some of which may be standard for some readers, and rather collect all definitions in a single section—Sect. 2. We remark that the primal tree-depth of a matrix A is a structural parameter that measures the complexity of interaction between the columns of A, and the dual tree-depth of a matrix A measures the complexity of interaction between the rows of A.

1.1.1 Characterization of depth parameters

Observe that the column matroid of the matrix is preserved by row operations, i.e., the column matroid of row-equivalent matrices is the same. The main structural result of [8, 9] is the following characterization of the existence of a row-equivalent matrix with small dual tree-depth in terms of the structural parameter of the column matroid [8, Theorem 1]. We remark that the term branch-depth was used in [8, 9] in line with the terminology from [36] but as there is a competing notion of branch-depth [16], we decided to use a different name for this depth parameter throughout the paper to avoid confusion.

Theorem 1 *For every non-zero matrix A, it holds that the smallest dual tree-depth of a matrix row-equivalent to A is equal to the contraction*-depth of $M(A)$, i.e.,* $\mathrm{td}^*_D(A) = \mathrm{c}^*\mathrm{d}(A)$.

We discover structural characterizations of the existence of a row-equivalent matrix with small primal tree-depth and the existence of a row-equivalent matrix with small incidence tree-depth.

Theorem 2 *For every matrix A, it holds that the smallest primal tree-depth of a matrix row-equivalent to A is equal to the deletion-depth of $M(A)$, i.e.,* $\mathrm{td}^*_P(A) = \mathrm{dd}(A)$.

Theorem 3 *For every matrix A, it holds that the smallest incidence tree-depth of a matrix row-equivalent to A is equal to contraction*-deletion-depth of $M(A)$ increased by one, i.e.,* $\mathrm{td}^*_I(A) = \mathrm{c}^*\mathrm{dd}(A) + 1$.

1.1.2 Interplay of circuit and graver basis complexity

Graver bases play an essential role in designing efficient algorithms for integer programming. We show that the maximum ℓ_1-norm of a circuit of a matrix A and the maximum ℓ_1-norm of an element of the Graver basis of A, which are denoted by $c_1(A)$ and $g_1(A)$, respectively, are functionally equivalent.

Theorem 4 *There exists a function $f_1 : \mathbb{N} \to \mathbb{N}$ such that the following holds for every rational matrix A with* $\dim \ker A > 0$*:*

$$c_1(A) \le g_1(A) \le f_1(c_1(A)).$$

The parameter $c_1(A)$ can be related to dual tree-depth and entry complexity as follows (we have opted throughout the paper to use entry complexity rather than $\|A\|_\infty$ as this permits to formulate our results for rational matrices rather than integral matrices, which is occasionally more convenient).

Theorem 5 *Every rational matrix A with* $\dim \ker A > 0$ *is row-equivalent to a rational matrix A' with* $\mathrm{td}_D(A') \le c_1(A)^2$ *and* $\mathrm{ec}(A') \le 2\lceil \log_2(c_1(A)+1) \rceil$.

Our results together with Theorem 9 imply that the following statements are equivalent for every rational matrix A:

- The ℓ_1-norm of every circuit of A, i.e., $c_1(A)$, is bounded.
- The ℓ_1-norm of every element of the Graver basis of A, i.e., $g_1(A)$, is bounded.
- The matrix A is row-equivalent to a matrix with bounded dual tree-depth and bounded entry complexity.
- The contraction*-depth of the matroid $M(A)$ is bounded, and the matrix A is row-equivalent to a matrix with bounded entry complexity (with any dual tree-depth).

1.1.3 Algorithms to compute matrices with small depth parameters

We also construct parameterized algorithms for transforming an input matrix to a row-equivalent matrix with small tree-depth and entry complexity if one exists. First, we design a parameterized algorithm for computing a row-equivalent matrix with small primal tree-depth and small entry complexity if one exists.

Theorem 6 *There exists a function $f : \mathbb{N}^2 \to \mathbb{N}$ and a fixed parameter algorithm for the parameterization by d and e that for a given rational matrix A:*

– *either outputs that A is not row-equivalent to a matrix with primal tree-depth at most d and entry complexity at most e, or*
– *outputs a matrix A' that is row-equivalent to A, its primal tree-depth is at most d and entry complexity is at most $f(d, e)$.*

The following algorithm for computing a row-equivalent matrix with small dual tree-depth was presented in [8, 9].

Theorem 7 *There exists a function $f : \mathbb{N}^2 \to \mathbb{N}$ and a fixed parameter algorithm for the parameterization by d and e that for a given rational matrix A with entry complexity at most e:*

– *either outputs that A is not row-equivalent to a matrix with dual tree-depth at most d, or*
– *outputs a matrix A' that is row-equivalent to A, its dual tree-depth is at most d and entry complexity is at most $f(d, e)$.*

We improve the algorithm by replacing the parameterization by the entry complexity of an input matrix with the parameterization by the entry complexity of the to be constructed matrix. Note that if a matrix A has entry complexity e and is row-equivalent to a matrix with dual tree-depth d, then Theorem 7 yields that A is row-equivalent to a matrix with dual tree-depth d and entry complexity bounded by a function of d and e. Hence, the algorithm given below applies to a wider set of input matrices than the algorithm from Theorem 7.

Theorem 8 *There exists a function $f : \mathbb{N}^2 \to \mathbb{N}$ and a fixed parameter algorithm for the parameterization by d and e that, for a given rational matrix A:*

– *either outputs that A is not row-equivalent to a matrix with dual tree-depth at most d and entry complexity at most e, or*
– *outputs a matrix A' that is row-equivalent to A, its dual tree-depth is at most d and entry complexity is at most $f(d, e)$.*

We point out the following difference between the cases of primal and dual tree-depth. As mentioned, if a matrix A has entry complexity e and is row-equivalent to a matrix with dual tree-depth d, then A is row-equivalent to a matrix with dual tree-depth d and entry complexity bounded by a function of d and e. However, the same is not true in the case of primal tree-depth. The entry complexity of every matrix with primal tree-depth equal to one that is row-equivalent to the following matrix A is linear in the number of rows of A, quite in a contrast to the case of dual tree-depth.

$$\begin{pmatrix} 1\,2\,0 & 0 & \cdots & 0 & 0\,0\,0 \\ 0\,1\,2 & 0 & \cdots & 0 & 0\,0\,0 \\ 0\,0\,1 & 2 & \cdots & 0 & 0\,0\,0 \\ \vdots\ \vdots & \ddots & \ddots & & \vdots\ \vdots \\ \vdots\ \vdots & & \ddots & \ddots & \vdots\ \vdots \\ 0\,0\,0 & 0 & \cdots & 1 & 2\,0\,0 \\ 0\,0\,0 & 0 & \cdots & 0 & 1\,2\,0 \\ 0\,0\,0 & 0 & \cdots & 0 & 0\,1\,2 \end{pmatrix}$$

1.1.4 Fixed parameter algorithms for integer programming

One of the open problems in the area, e.g. discussed during the Dagstuhl workshop 19041 "New Horizons in Parameterized Complexity", has been whether integer programming is fixed parameter tractable when parameterized by $g_1(A)$, i.e., by the ℓ_1-norm of an element of the Graver basis of the constraint matrix A. Our results on the interplay of dual tree-depth, the circuit complexity and the Graver basis complexity of a matrix yield an affirmative answer. The existence of appropriate preconditioners that we establish in this paper implies that integer programming is fixed parameter tractable when parameterized by

- $g_1(A)$, i.e., the ℓ_1-norm of the Graver basis of the constraint matrix,
- $c_1(A)$, i.e., the ℓ_1-norm of the circuits of the constraint matrix,
- $\mathrm{td}^*_P(A)$ and $\mathrm{ec}(A)$, i.e., the smallest primal tree-depth and entry complexity of a matrix row-equivalent to the constraint matrix, and
- $\mathrm{td}^*_D(A)$ and $\mathrm{ec}(A)$, i.e., the smallest dual tree-depth and entry complexity of a matrix row-equivalent to the constraint matrix.

We believe that our new tractability results significantly enhance the toolbox of tractable IPs as the nature of our tractability conditions substantially differ from prevalent block-structured sparsity-based tractability conditions. The importance of availability of various forms of tractable IPs can be witnessed by n-fold IPs, which were shown fixed-parameter tractable in [26], and, about a decade later, their applications have become ubiquitous, see e.g. [6, 7, 10, 11, 28, 32, 33, 42, 43].

1.1.5 Hardness results

As our algorithmic results involve computing depth decompositions of matroids for various depth parameters in a parameterized way, we establish computational hardness of these parameters in Theorem 12, primarily for the sake of completeness of our exposition. In particular, computing the following matroid parameters is NP-complete:

- deletion-depth,
- contraction-depth,

- contraction-deletion-depth,
- contraction*-depth, and
- contraction*-deletion-depth.

2 Preliminaries

In this section, we fix the notation used throughout the paper. We start with general notation and we then fix the notation related to graphs, matrices and matroids.

The set of all positive integers is denoted by $\mathbb{N}$ and the set of the first k positive integers by $[k]$. If A is a linear space, we write $\dim A$ for its dimension. If K is a subspace of A, the *quotient space* A/K is the linear space of the dimension $\dim A - \dim K$ that consists of cosets of A given by K with the natural operations of addition and scalar multiplication; see e.g. [25] for further details. The quotient space A/K can be associated with a linear subspace of A of dimension $\dim A - \dim K$ formed by exactly a single vector from each coset of A given by K; we will often view the quotient space as such a subspace of A and write $w + K$ for the coset containing a vector w. For example, if A is $\mathbb{R}^3$ and K is the linear space generated by $(0, 0, 1)$, A/K can be associated with (or viewed as) the 2-dimensional space formed by vectors $(x, y, 0)$, $x, y \in \mathbb{R}$.

2.1 Graphs

All graphs considered in this paper are loopless simple graphs unless stated otherwise. If G is a graph, then we write $V(G)$ and $E(G)$ for the vertex set and the edge set of G, respectively. If W is a subset of vertices of a graph G, then $G \setminus W$ is the graph obtained by removing the vertices of W (and all edges incident with them), and $G[W]$ is the graph obtained by removing all vertices not contained in W (and all edges incident with them). If F is a subset of edges of a graph G, then $G \setminus F$ is the graph obtained by removing the edges contained in F and G/F is the graph obtained by contracting all edges contained in F and removing resulting loops and parallel edges (while keeping one edge from each group of parallel edges).

We next define the graph parameter *tree-depth*, which is the central graph parameter in this paper. The *height* of a rooted tree is the maximum number of vertices on a path from the root to a leaf, and the *height* of a rooted forest, i.e., a graph whose each component is a rooted tree, is the maximum height of its components. The *depth* of a rooted tree is the maximum number of edges on a path from the root to a leaf, and the *depth* of a rooted forest is the maximum depth of its components. Note that the height and the depth of a rooted tree always differ by one; we use both notions to avoid cumbersome way of expressing that would otherwise require adding or subtracting one. The *closure* $\mathrm{cl}(F)$ of a rooted forest F is the graph obtained by adding edges from each vertex to all its descendants. Finally, the *tree-depth* $\mathrm{td}(G)$ of a graph G is the minimum height of a rooted forest F such that the closure $\mathrm{cl}(F)$ of the rooted forest F contains G as a subgraph. See Fig. 1 for an example. It can be shown that the path-width, and so the tree-width, of any graph is at most its tree-depth decreased by one;

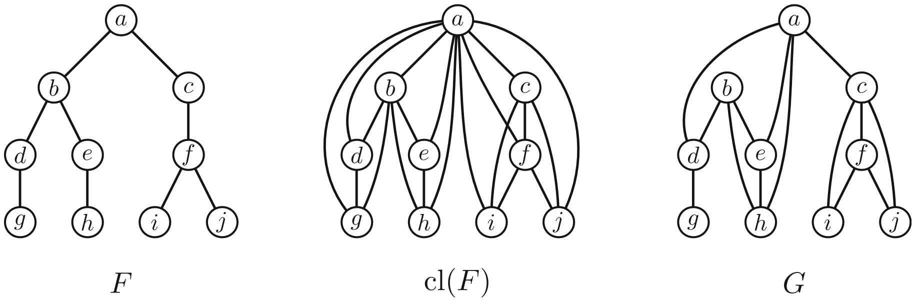


Fig. 1 A rooted forest F consisting of a single tree and its closure $\mathrm{cl}(F)$, which shows that tree-depth of the depicted graph G is (at most) four

see e.g. [14] for a more detailed discussion of the relation of tree-depth, path-width and tree-width, and their algorithmic applications.

2.2 Matroids

We next review basic definitions from matroid theory; we refer to the book of Oxley [46] for detailed exposition. A *hereditary* collection of subsets of a set is a collection closed under taking subsets; in particular, every non-empty hereditary collection of subsets contains the empty set. A *matroid* M is a pair $(X, \mathcal{I})$, where $\mathcal{I}$ is a non-empty hereditary collection of subsets of X that satisfies the *augmentation axiom*, i.e., if $X' \in \mathcal{I}$, $X'' \in \mathcal{I}$ and $|X'| < |X''|$, then there exists an element $x \in X'' \setminus X'$ such that $X' \cup \{x\} \in \mathcal{I}$. The set X is the *ground set* of M and the sets contained in $\mathcal{I}$ are referred to as *independent*. We often refer to elements of the ground set of M to as elements of the matroid M, and if e is an element of (the ground set of) M, we also write $e \in M$. Two important examples of matroids are vector matroids and graphic matroids. A *vector matroid* is a matroid whose ground set is formed by vectors and independent sets are precisely sets of linearly independent vectors (note that the augmentation axiom follows from the Steinitz exchange lemma). A *graphic matroid* is a matroid whose ground set is formed by edges of a graph and independent sets are precisely acyclic sets of edges, i.e., sets not containing a cycle.

The *rank* of a subset X' of the ground set X, which is denoted by $r_M(X')$ or simply by $r(X')$ if M is clear from the context, is the maximum size of an independent subset of X' (it can be shown that all maximal independent subsets of X' have the same cardinality); the *rank* of the matroid M, which is denoted by $r(M)$, is the rank of its ground set. Note that in the case of vector matroids, the rank of X' is exactly the dimension of linear space generated by X'. A *basis* of a matroid M is a maximal independent subset of the ground set of M and a *circuit* is a minimal subset of the ground set of M that is not independent. In particular, if X' is a circuit of M, then $r(X') = |X'| - 1$ and every proper subset of X' is independent. An element x of a matroid M is a *loop* if $r(\{x\}) = 0$, an element x is a *bridge* if it is contained in every basis of M, and two elements x and x' are *parallel* if $r(\{x\}) = r(\{x'\}) = r(\{x, x'\}) = 1$. Note that in the case of vector matroids, two non-loop elements are parallel if and only if they are non-zero multiple of each other. If M is a matroid with ground set X, the *dual matroid*, which is denoted by M^*, is the matroid with the same ground set X such that $X' \subseteq X$ is independent in M^* if and only if $r_M(X \setminus X') = r(M)$; in particular, $r_{M^*}(X') = r_M(X \setminus X') + |X'| - r(M)$ for every $X' \subseteq X$.

For a field $\mathbb{F}$, we say that a matroid M is $\mathbb{F}$-*representable* if every element of M can be assigned a vector from $\mathbb{F}^{r(M)}$ in such a way that a subset of the ground set of M is independent if and only if the set of assigned vectors is linearly independent. In particular, an element of M is a loop if and only if it is assigned the zero vector and two non-loop elements of M are parallel if and only if they are non-zero multiples of each other. Such an assignment of vectors of $\mathbb{F}^{r(M)}$ to the elements of M is an $\mathbb{F}$-*representation* of M. Clearly, a matroid M is $\mathbb{F}$-representable if and only if it is isomorphic to the vector matroid given by its $\mathbb{F}$-representation. Matroids representable over the 2-element field are referred to as *binary* matroids. We say that a matroid M is $\mathbb{F}$-*represented* if the matroid M is given by its $\mathbb{F}$-representation. If a particular field $\mathbb{F}$ is not relevant in the context, we just say that a matroid M is *represented* to express that it is given by its representation.

Let M be a matroid with a ground set X. The matroid kM for $k \in \mathbb{N}$ is the matroid obtained from M by introducing $k-1$ parallel elements to each non-loop element and $k-1$ additional loops for each loop; informally speaking, every element of M is "cloned" to k copies. Note that a subset X' of the elements of kM is independent if and only if it does not contains two clones of the same element and the set of the elements of M corresponding to those contained in X' is independent. Observe that if M is a vector matroid, then kM is the vector matroid obtained by adding $k-1$ copies of each vector forming M. Similarly, if M is a graphic matroid associated with a graph G, then kM is the graphic matroid obtained from the graph G by duplicating each edge $k-1$ times.

If $X' \subseteq X$, then the *restriction* of M to X', which is denoted by $M\left[X'\right]$, is the matroid with the ground set X' such that a subset of X' is independent in $M\left[X'\right]$ if and only if it is independent in M. In particular, the rank of $M\left[X'\right]$ is $r_M(X')$. For example, if M is a graphic matroid associated with a graph G, then the restriction of M to X' is the graphic matroid associated with the spanning subgraph of G with edge set X'. The matroid obtained from M by *deleting* X' is the restriction of M to $X \setminus X'$ and is denoted by $M \setminus X'$.

The *contraction* of M by X', which is denoted by M/X', is the matroid with the ground set $X \setminus X'$ such that a subset X'' of $X \setminus X'$ is independent in M/X' if and only if $r_M(X'' \cup X') = |X''| + r_M(X')$. If X' is a single element set and e is its only element, we write $M \setminus e$ and M/e instead of $M \setminus \{e\}$ and $M/\{e\}$, respectively. If M is a graphic matroid associated with a graph G and e is an edge of G, then M/e is the graphic matroid associated with the graph obtained from G by contracting the edge e (while keeping all resulting loops and parallel edges). If an $\mathbb{F}$-representation of M is given and X' is a subset of the ground set of M, then an $\mathbb{F}$-representation of M/X' can be obtained from the $\mathbb{F}$-representation of M by considering the representation in the quotient space by the linear hull of the vectors representing the elements of X'. This leads us to the following definition: if M is an $\mathbb{F}$-represented matroid and A is a linear subspace of $\mathbb{F}^{r(M)}$, then the matroid M/A is the $\mathbb{F}$-represented matroid with the representation of M in the quotient space by A. Note that the ground sets of M and M/A are the same, in particular, M and M/A have the same number of elements.

A matroid M is *connected* if every two distinct elements of M are contained in a common circuit. We remark that the property of being contained in a common circuit is transitive [46, Proposition 4.1.2], i.e., if the pair of elements e and e' is contained in a common circuit and the pair e' and e'' is also contained in a common circuit, then the pair e and e'' is also contained in a common circuit. If M is an $\mathbb{F}$-represented matroid with at least two elements, then M is connected if and only if M has no loops and there do not exist two non-trivial linear spaces A and B of $\mathbb{F}^{r(M)}$ such that $A \cap B$ contains the zero vector only and every element of M is contained in A or B (the linear space $\mathbb{F}^{r(M)}$ would be the direct sum of the linear spaces A and B). Also observe that if M is a graphic matroid associated with a graph G, then the matroid M is connected if and only if the graph G is 2-connected.

A *component* of a matroid M is an inclusion-wise maximal connected restriction of M; a component is *trivial* if it consists of a single loop, and it is *non-trivial* otherwise. If M is a vector matroid, then each non-trivial component of M can be associated with a linear space such that each element of M is contained in one of the linear spaces and the linear hull of all elements of M is the direct sum of the linear spaces. We often identify components of a matroid M with their element sets. Using this identification, it holds that a subset X' of a ground set of a matroid M is a component of M if and only if X' is a component of M^* (we use this equivalence to prove some of our hardness results in Sect. 6). We remark that $(M^*)^* = M$ for every matroid M, and if e is an element of a matroid M, then $(M/e)^* = M^* \setminus e$ and $(M\setminus e)^* = M^*/e$.

2.3 Matrices

In this section, we define notation related to matrices. If $\mathbb{F}$ is a field, we write $\mathbb{F}^{m\times n}$ for the set of matrices with m rows and n columns over the field $\mathbb{F}$. If A is a rational matrix, the entry complexity $\mathrm{ec}(A)$ is the maximum length of a binary encoding of its entries, i.e., the maximum of $\lceil \log_2(|p|+1)\rceil + \lceil(\log_2 |q|+1)\rceil$ taken over all entries p/q of A (where p and q are always assumed to be coprime). If A is an integral matrix, then $\mathrm{ec}(A) = \Theta(\log \|A\|_\infty)$. Throughout the paper, we use the entry complexity rather than the ℓ_∞-norm of matrices as this permits formulating our results for rational matrices rather than integral matrices only.

A rational matrix A is *z-integral* for $z \in \mathbb{Q}$ if every entry of A is an integral multiple of z. We say that two matrices A and A' are *row-equivalent* if one can be obtained from another by *elementary row operations*, i.e., by repeatedly adding a multiple of one row to another and multiplying a row by a non-zero element. Observe that if A and A' are row-equivalent matrices, then their kernels are the same. For a matrix A, we define $M(A)$ to be the represented matroid whose elements are the columns of A. Again, if matrices A and A' are row-equivalent, then the matroids $M(A)$ and $M(A')$ are the same.

If A is a matrix, the *primal graph* of A is the graph whose vertices are columns of A and two vertices are adjacent if there exists a row having non-zero elements in the two columns associated with the vertices; the *dual graph* of A is the graph

$$\begin{array}{c} \\ \alpha \\ \beta \\ \gamma \\ \delta \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 3 & 2 & 3 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

matrix A

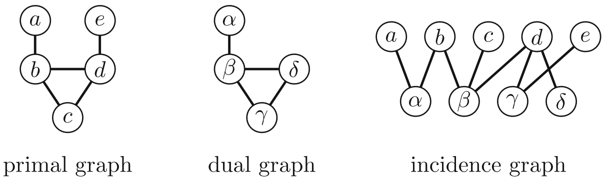


Fig. 2 The primal graph, the dual graph and the incidence graph of the depicted matrix A

whose vertices are rows of A and two vertices are adjacent if there exists a column having non-zero elements in the two associated rows; the *incidence graph* of A is the bipartite graph with one part formed by rows of A and the other part by columns of A and two vertices are adjacent if the entry in the associated row and in the associated column is non-zero. See Fig. 2 for an example. The *primal tree-depth* of A, denoted by $\mathrm{td}_P(A)$, is the tree-depth of the primal graph of A, the *dual tree-depth* of A, denoted by $\mathrm{td}_D(A)$, is the tree-depth of the dual graph of A, and the *incidence tree-depth* of A, denoted by $\mathrm{td}_I(A)$, is the tree-depth of the incidence graph of A. Finally, $\mathrm{td}^*_P(A)$ is the smallest primal tree-depth of a matrix row-equivalent to A, $\mathrm{td}^*_D(A)$ is the smallest dual tree-depth of a matrix row-equivalent to A, and $\mathrm{td}^*_I(A)$ is the smallest incidence tree-depth of a matrix row-equivalent to A.

A *circuit* of a rational matrix A is a support-wise minimal integral vector contained in the kernel of A such that all its entries are coprime; the set of circuits of A is denoted by $\mathcal{C}(A)$. Note that a set X of columns is a circuit in the matroid $M(A)$ if and only if $\mathcal{C}(A)$ contains a vector with the support exactly equal to X. We write $c_1(A)$ for the maximum ℓ_1-norm of a circuit of A and $c_\infty(A)$ for the maximum ℓ_∞-norm of a circuit of A. Note if A and A' are row-equivalent rational matrices, then $\mathcal{C}(A) = \mathcal{C}(A')$ and so the parameters $c_1(\cdot)$ and $c_\infty(\cdot)$ are invariant under elementary row operations. Following the notation from [21], we write $\dot{\kappa}_A$ for the least common multiple of the entries of the circuits of A. Observe that there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that $\dot{\kappa}_A \le f(c_\infty(A))$ for every matrix A.

If $\mathbf{x}$ and $\mathbf{y}$ are two d-dimensional vectors, we write $\mathbf{x} \sqsubseteq \mathbf{y}$ if $|\mathbf{x}_i| \le |\mathbf{y}_i|$ for all $i \in [d]$ and $\mathbf{x}$ and $\mathbf{y}$ are in the same orthant, i.e., $\mathbf{x}_i$ and $\mathbf{y}_i$ have the same sign (or one or both are zero) for all $i \in [d]$. The *Graver basis* of a matrix A, denoted by $\mathcal{G}(A)$, is the set of the $\sqsubseteq$-minimal non-zero elements of the integer kernel $\ker_{\mathbb{Z}}(A)$. We use $g_1(A)$ and $g_\infty(A)$ for the Graver basis of A analogously to the set of circuits, i.e., $g_1(A)$ is the maximum ℓ_1-norm of a vector in $\mathcal{G}(A)$ and $g_\infty(A)$ is the maximum ℓ_∞-norm of a vector in $\mathcal{G}(A)$. Again, the parameters $g_1(\cdot)$ and $g_\infty(\cdot)$ are invariant under elementary row operations as the Graver bases of row-equivalent matrices are the same. Note that every circuit of a matrix A belongs to the Graver basis of A, i.e., $\mathcal{C}(A) \subseteq \mathcal{G}(A)$, and so it holds that $c_1(A) \le g_1(A)$ and $c_\infty(A) \le g_\infty(A)$ for every matrix A.

The existence of efficient algorithms for integer programming with of constraint matrices A with bounded primal and dual tree-depth is closely linked to bounds on the norm of elements of the Graver basis of A. In particular, Koutecký, Levin and Onn [44] established the following.

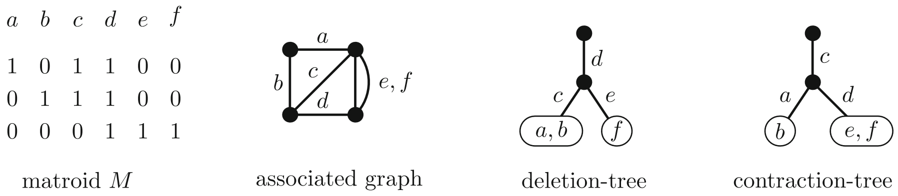


Fig. 3 A deletion-tree and a contraction-tree of the depicted binary matroid M, which is also the graphic matroid associated with the depicted graph

Theorem 9 *There exist functions* $f_P, f_D : \mathbb{N}^2 \to \mathbb{N}$ *such that the following holds for every rational matrix A:* $g_\infty(A) \leq f_P(\mathrm{td}_P(A), \mathrm{ec}(A))$ *and* $g_1(A) \leq f_D(\mathrm{td}_D(A), \mathrm{ec}(A))$.

2.4 Matroid depth parameters

We now define matroid depth parameters that will be of importance further. We start with the notion of deletion-depth and contraction-depth, which were introduced by DeVos, Kwon and Oum [16].

The *deletion-depth* of a matroid M, denoted by $\mathrm{dd}(M)$, is defined recursively as follows:

- If M has a single element, then $\mathrm{dd}(M) = 1$.
- If M is not connected, then $\mathrm{dd}(M)$ is the maximum deletion-depth of a component of M.
- Otherwise, $\mathrm{dd}(M) = 1 + \min_{e \in M} \mathrm{dd}(M \setminus e)$, i.e., $\mathrm{dd}(M)$ is 1 plus the minimum deletion-depth of $M \setminus e$ where the minimum is taken over all elements e of M.

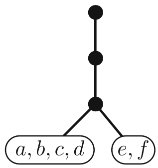
A sequence of deletions of elements witnessing that the deletion-depth of a matroid M is $\mathrm{dd}(M)$ can be visualized by a rooted tree, which we call a *deletion-tree*, defined as follows. If M has a single element, then the deletion-tree of M consists of a single vertex labeled with the single element of M. If M is not connected, then the deletion-tree is obtained by identifying the roots of deletion-trees of the components of M. Otherwise, there exists an element e of the matroid M such that $\mathrm{dd}(M) = \mathrm{dd}(M \setminus e) + 1$ and the deletion-tree of M is obtained from the deletion-tree of $M \setminus e$ by adding a new vertex adjacent to the root of the deletion-tree of $M \setminus e$, changing the root of the tree to the newly added vertex and labeling the edge incident with it with the element e. See Fig. 3 for an example. Observe that the height of the deletion-tree is equal to the deletion-depth of M. In what follows, we consider deletion-trees that need not to be of optimal height, i.e., its edges can be labeled by a sequence of elements that decomposes a matroid M in a way described in the definition of the deletion-depth but its height is larger than $\mathrm{dd}(M)$. In this more general setting, the deletion-depth of a matroid M is the smallest height of a deletion-tree of M.

The *contraction-depth* of a matroid M, denoted by $\mathrm{cd}(M)$, is defined recursively as follows:

- If M has a single element, then $\mathrm{cd}(M) = 1$.

$$\begin{array}{cccccc} a & b & c & d & e & f \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

matroid M



tree T

Fig. 4 A binary matroid M of rank four given by its representation and a tree T as in the definition of the contraction*-depth

- If M is not connected, then $\mathrm{cd}(M)$ is the maximum contraction-depth of a component of M.
- Otherwise, $\mathrm{cd}(M) = 1 + \min_{e \in M} \mathrm{cd}(M/e)$, i.e., $\mathrm{cd}(M)$ is 1 plus the minimum contraction-depth of M/e where the minimum is taken over all elements e of M.

It is not hard to show that $\mathrm{dd}(M) = \mathrm{cd}(M^*)$ and $\mathrm{cd}(M) = \mathrm{dd}(M^*)$ for every matroid M. We define a *contraction-tree* analogously to a deletion-tree; the contraction-depth of a matroid M is the smallest height of a contraction-tree of M (an example is given in Fig. 3).

We next introduce the contraction-deletion-depth; this parameter was studied under the name *type* in [17], however, we decided to adopt the name contraction-deletion-depth from [16], which we find to better fit the context considered here. The *contraction-deletion-depth* of a matroid M, denoted by $\mathrm{cdd}(M)$, is defined recursively as follows:

- If M has a single element, then $\mathrm{cdd}(M) = 1$.
- If M is not connected, then $\mathrm{cdd}(M)$ is the maximum contraction-deletion-depth of a component of M.
- Otherwise, $\mathrm{cdd}(M) = 1 + \min_{e \in M} \min\{\mathrm{cdd}(M \setminus e), \mathrm{cdd}(M/e)\}$, i.e., $\mathrm{cdd}(M)$ is 1 plus the smaller among the minimum contraction-deletion-depth of the matroid $M \setminus e$ and the minimum contraction-deletion-depth of the matroid M/e where both minima are taken over all elements e of M.

Observe that it holds that $\mathrm{cdd}(M) = \mathrm{cdd}(M^*)$, $\mathrm{cdd}(M) \leq \mathrm{dd}(M)$ and $\mathrm{cdd}(M) \leq \mathrm{cd}(M)$ for every matroid M.

One of the key parameters in our setting is that of contraction*-depth; this parameter was introduced under the name branch-depth in [36] and further studied in [8] but we decided to use a different name to avoid a possible confusion with the notion of branch-depth introduced in [16]. We first introduce the parameter for general matroids, and then present an equivalent definition for represented matroids, which is more convenient to work in our setting. The *contraction**-*depth* of a matroid M, denoted by $\mathrm{c}^{*}\!\mathrm{d}(M)$, is the smallest depth of a rooted tree T with exactly $r(M)$ edges with the following property: there exists a function f from the ground set of M to the leaves of T such that for every subset X of the ground set of M the total number of edges contained in paths from the root to vertices of X is at least $r(X)$. An example is given in Fig. 4.

There is an alternative definition of the parameter for represented matroids, which also justifies the name that we use for the parameter. The *contraction***-depth* of a represented matroid M can be defined recursively as follows:

- If M has rank zero, then $\mathrm{c\overset{*}{d}}(M) = 0$.
- If M is not connected, then $\mathrm{c\overset{*}{d}}(M)$ is the maximum contraction*-depth of a component of M.
- Otherwise, $\mathrm{c\overset{*}{d}}(M)$ is 1 plus the minimum contraction*-depth of a matroid obtained from the matroid M by factoring along an arbitrary one-dimensional subspace.

As the contraction in the definition is allowed to be by an arbitrary one-dimensional subspace, not only by a subspace generated by an element of M, it follows that $\mathrm{c\overset{*}{d}}(M) \le \mathrm{cd}(M)$.

The sequence of such contractions can be visualized by a *contraction***-tree* that is defined in the same way as a contraction-tree except that one-vertex trees are associated with matroids of rank zero (rather than matroids consisting of a single element) and the edges are labeled by one-dimensional subspaces. If each one-dimensional subspace that is the label of an edge of the tree is generated by an element of the matroid M, we say that the contraction*-tree is *principal* and we view the edges of the tree as labeled by the corresponding elements of M. Note that the minimum depth of a principal contraction*-tree of a matroid M is an upper bound on its contraction*-depth, however, in general, the contraction*-depth of a matroid M can be smaller than the minimum depth of a principal contraction*-tree of M. We point out that the notions of principal contraction*-trees and contraction-trees differ in a subtle but important way. For example, if M is a vector matroid of rank one containing a single element e, its only contraction-tree consists of a root labeled by e while its only contraction*-tree has a root and a leaf adjacent to it and the edge joining them is labeled with (the subspace generated by) e. However, if M is a vector matroid of rank one containing two parallel elements e and e', any contraction-tree and any contraction*-tree of M has depth one. Still, the minimum depth of a principal contraction*-tree of M is either $\mathrm{cd}(M) - 1$ or $\mathrm{cd}(M)$.

Kardoš et al. [36] established the connection between the contraction*-depth and the existence of a long circuit, which is described in Theorem 10 below; Theorem 10 implies that $\mathrm{cd}(M) \le k^2 + 1$ where k is the size of the largest circuit of M.

Theorem 10 *Let M be a matroid and k the size of its largest circuit. It holds that $\log_2 k \le \mathrm{c\overset{*}{d}}(M) \le k^2$. Moreover, there exists a polynomial-time algorithm that for an input oracle-given matroid M outputs a principal contraction*-tree of depth at most k^2.*

We next introduce the parameter of contraction*-deletion-depth, which we believe to have not been yet studied previously, but which is particularly relevant in our context. To avoid unnecessary technical issues, we introduce the parameter for represented matroids only. The *contraction***-deletion-depth* of a represented matroid M, denoted by $\mathrm{c\overset{*}{d}d}(M)$, is defined recursively as follows:

- If M has rank zero, then $\mathrm{c\overset{*}{d}d}(M) = 0$;
- if M has a single non-loop element, then $\mathrm{c\overset{*}{d}d}(M) = 1$.

– If M is not connected, then $\mathrm{c\overset{*}{d}d}(M)$ is the maximum contraction*-deletion-depth of a component of M.
– Otherwise, $\mathrm{c\overset{*}{d}d}(M)$ is 1 plus the smaller among the minimum contraction*-deletion-depth of the matroid $M \setminus e$, where the minimum is taken over all elements of M, and the minimum contraction*-deletion-depth of a matroid obtained from M by factoring along an arbitrary one-dimensional subspace.

Observe that $\mathrm{c\overset{*}{d}d}(M) \le \mathrm{cdd}(M)$ and $\mathrm{c\overset{*}{d}d}(M) \le \mathrm{c\overset{*}{d}}(M)$ for every matroid M.

Finally, if A is a matrix, the *deletion-depth*, *contraction-depth*, etc. of A is the corresponding parameter of the vector matroid $M(A)$ formed by the columns of A, and we write $\mathrm{dd}(A)$, $\mathrm{cd}(A)$, etc. for the deletion-depth, contraction-depth, etc. of the matrix A. Observe that the deletion-depth, contraction-depth etc. of a matrix A is invariant under elementary row operations as elementary row operations preserve the matroid $M(A)$.

3 Structural results

In this section, we prove our structural results concerning optimal primal tree-depth and optimal incidence tree-depth of a matrix. We start with presenting an algorithm, which uses a deletion-tree of the matroid associated with a given matrix to construct a row-equivalent matrix with small primal tree-depth.

Lemma 1 *There exists a polynomial-time algorithm that for an input matrix A and a deletion-tree of $M(A)$ with height d outputs a matrix A' row-equivalent to A such that* $\mathrm{td}_P(A') \le d$.

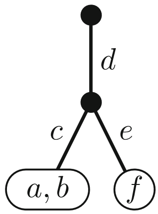
Proof We establish the existence of the algorithm by proving that $\mathrm{td}_P(A') \le d$ in a constructive (algorithmic) way. Fix a matrix A and a deletion-tree T of $M(A)$ with height d.

Let X be the set of non-zero columns that are labels of the vertices of T. We show that the columns contained in X form a basis of the column space of the matrix A. As the matroid obtained from $M(A)$ by deleting the labels of the edges of T has no component of size two or more, the columns contained in X are linearly independent. Suppose that there exists a non-zero column x that is not a linear combination of the columns contained in X, and choose among all such columns the label of an edge e as far from the root of T as possible. Since the element x does not form its own component in the matroid obtained from M by deleting the labels of all edges on the path from the root to e (excluding e), x is a linear combination of the labels of the vertices and edges of the subtree of T delimited by e. This implies that either x is a linear combination of the columns in X or there is a label of an edge of this subtree that is not a linear combination of the columns in X contrary to the choice of x. We conclude that X is a basis of the column space of A. In particular, unless A is the zero matrix, the set X is non-empty.

Let A' be the matrix obtained from A by elementary row operations such that the submatrix of A' induced by the columns of X is the unit matrix and with some additional zero rows; note that the set X is determined by the input deletion-tree and

$$\begin{array}{cccccc} a & b & c & d & e & f \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{array}$$

matrix A



deletion-tree

$$\begin{array}{cccccc} a & b & c & d & e & f \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

matrix A'

Fig. 5 A binary matrix A, a deletion-tree of the matroid $M(A)$ and the matrix A' as in the proof of Lemma 1. Note that $X = \{a, b, f\}$

so the matrix A' depends on the deletion-tree. See Fig. 5 for an example. This finishes the construction of A' and, in particular, the description of the algorithm to produce the output matrix A'. To complete the proof, it remains to show that the primal tree-depth of A' is at most d, i.e., the correctness of the algorithm. This will be proven by induction on the number of columns of an input matrix A.

The base of the induction is the case when A has a single column. In this case, the primal tree-depth of A' is one and the tree T is a single vertex labeled with the only column of A, and so its height is one. We next present the induction step. First observe that every label of a vertex of T is either in X or a loop in $M(A)$ (recall that only non-zero columns of A are included to X), and every label of an edge e is a linear combination of labels of the vertices in the subtree delimited by e.

Suppose that the root of T has a label and let x be one of its labels; note that x is either a loop or a bridge in the matroid $M(A)$. Let B be the matrix obtained from A by deleting the column x, and let T' be the deletion-tree of $M(B)$ obtained from T by removing the label x from the root. Let B' be the matrix produced by the algorithm described above for B and T'. If x is a loop, then the matrix A' is the matrix B' extended by a zero column with possibly permuted rows, and if x is a bridge (and so $x \in X$), then the matrix A' is, possibly after permuting rows, the matrix B' extended by a unit vector such that its non-zero entry is the only non-zero entry in its row. In either case, the vertex associated with the column x is isolated in the primal graph of A', and it follows that $\mathrm{td}_P(A') = \mathrm{td}_P(B') \leq d$ (the inequality holds by the induction hypothesis). Hence, we can assume that the root of T has no label.

We next analyze the case that the root of T has a single child and no label. Let x be the label of the single edge incident with the root of T. Let B be the matrix obtained from A by deleting the column x, and let T' be the deletion-tree of $M(B)$ obtained from T by deleting the edge incident with the root and rooting the tree at the remaining vertex of the deleted edge. Let B' be the matrix produced by the algorithm described above for B and T'; note that the primal tree-depth of B' is at most $d - 1$ by the induction hypothesis. Since B' is the submatrix of A' formed by the columns different from x (possibly after permuting rows), the primal tree-depth of A' is at most $\mathrm{td}_P(B') + 1 = d$.

The final case to analyze is the case when the root of T has $k \geq 2$ children (in addition to having no label). Let $T_1, \ldots, T_k$ be the k subtrees of T delimited by the k

edges incident with the root of T, let $Y_1, \ldots, Y_k$ be the labels of the vertices and edges of these subtrees, and let $B_1, \ldots, B_k$ be the submatrices of A formed by the columns contained in $Y_1, \ldots, Y_k$. Observe that T_i is a deletion-tree of the matroid $M(B_i)$ for $i = 1, \ldots, k$, and the matrix B_i' produced by the algorithm described above for B_i and T_i is the submatrix of A' formed by the columns contained in Y_i (possibly after permuting rows). Since the support of the columns contained in Y_i contains only the unit entries of the columns of A' contained in $X \cap Y_i$, the primal graph of A' contains no edge joining a column of Y_i and a column of Y_j for $i \neq j$. It follows that the primal tree-depth of A' is at most the maximum primal tree-depth of B_i, which is at most d by the induction hypothesis. It follows that $\mathrm{td}_P(A') \leq d$ as desired. The proof of the correctness of the algorithm is now completed and so is the proof of the lemma. □

We are now ready to establish the link between the optimal primal tree-depth and the deletion-depth of the matroid associated with the matrix.

Proof of Theorem 2 Fix a matrix A. By Lemma 1, it holds that $\mathrm{td}^*_P(A) \leq \mathrm{dd}(A)$ as there exists a deletion-tree of the matroid $M(A)$ with height $\mathrm{dd}(A)$. So, we focus on proving that $\mathrm{dd}(A) \leq \mathrm{td}^*_P(A)$. We will show that every matrix B satisfies that $\mathrm{dd}(B) \leq \mathrm{td}_P(B)$; this implies that $\mathrm{dd}(A) \leq \mathrm{td}_P(A')$ for every matrix A' row-equivalent to A as $\mathrm{dd}(A) = \mathrm{dd}(A')$ and so implies that $\mathrm{dd}(A) \leq \mathrm{td}^*_P(A)$.

The proof that $\mathrm{dd}(B) \leq \mathrm{td}_P(B)$ proceeds by induction on the number of columns. If B has a single column, then both $\mathrm{dd}(B)$ and $\mathrm{td}_P(B)$ are equal to one. We next present the induction step. We first consider the case when the matroid $M(B)$ is not connected. Let $B_1, \ldots, B_k$ be the submatrices of B formed by columns corresponding to the components of $M(B)$; note that some of the submatrices may consist of a single zero column (if $M(B)$ has a loop). The definition of the deletion-depth implies that $\mathrm{dd}(B)$ is the maximum among $\mathrm{dd}(B_1), \ldots, \mathrm{dd}(B_k)$. On the other hand, the primal tree-depth of each of the matrices B_i is at most the primal tree-depth of the matrix B as the primal graph of B_i is a subgraph of the primal graph of B. It follows that $\mathrm{dd}(B_i) \leq \mathrm{td}_P(B)$, which implies that $\mathrm{dd}(B) \leq \mathrm{td}_P(B)$.

We next assume that the matroid $M(B)$ is connected and claim that the primal graph of B must also be connected. Suppose that the primal graph of B is not connected, i.e., there exists a partition of rows of B into R_1 and R_2 and a partition of the columns into C_1 and C_2, such that for each $i = 1, 2$, the support of each column in C_i is contained in R_i. Therefore, for any dependent set of columns of B, either its subset formed by columns contained in C_1 is dependent, or its subset formed by by columns contained in C_2 is dependent, or both these subsets are dependent. It follows that the support of every circuit of $M(B)$ is fully contained in either C_1 or C_2; in particular, no there is no circuit of $M(B)$ containing a column from C_1 and a column from C_2. This implies that $M(B)$ is not connected. Hence, the primal graph of B must be connected.

Since the primal graph of B is connected, there exists a column such that the matrix B' obtained by deleting this column satisfies that $\mathrm{td}_P(B') = \mathrm{td}_P(B) - 1$. The induction assumption yields that $\mathrm{dd}(B') \leq \mathrm{td}_P(B) - 1$ and the definition of the deletion-depth yields that the deletion-depth of $M(B)$ is at most the deletion-depth of $M(B')$ increased by one. This implies that $\mathrm{dd}(B) = \mathrm{dd}(M(B)) \leq \mathrm{td}_P(B)$ as desired. □

Before proceeding with our structural result concerning incidence tree-depth, we use the structural results presented in Lemma 1 and Theorem 2 to get a parameterized algorithm for computing an optimal primal tree-depth of a matrix over a finite field.

Corollary 1 *There exists a fixed parameter algorithm for the parameterization by a finite field $\mathbb{F}$ and an integer d that for an input matrix A over the field $\mathbb{F}$,*

- *either outputs that* $\mathrm{td}^*_P(A) > d$*, or*
- *computes a matrix A' row-equivalent to A with* $\mathrm{td}_P(A') \le d$ *and also outputs the associated deletion-tree of $M(A)$ with height* $\mathrm{td}_P(A')$.

Proof We first show that the property that a matroid M has deletion depth at most d can be expressed in monadic second order logic. Recall that monadic second order formulas for matroids may contain quantification over elements and subsets of elements of a matroid, and the predicate $\psi_I(\cdot)$ used to test whether a particular subset is independent in addition to logic connectives and the equality $=$, the set inclusion $\in$ and the subset inclusion $\subseteq$. In the formulas that we present, small letters are used to denote elements of a matroid and capital letters subsets of the elements. We next present a monadic second order formula $\psi_d(X)$ that describes whether the deletion-depth of the restriction of the matroid M to a subset X of the elements of M is at most d, which would imply the statement. The following formula $\psi_c(\cdot,\cdot)$ describes the existence of a circuit containing two distinct elements:

$$\psi_c(x,y) \equiv (x \neq y) \wedge (\exists X : x \in X \wedge y \in X \wedge \neg\psi_I(X) \wedge \forall z \in X : \psi_I(X \setminus z))\,.$$

The next formula $\psi_C(\cdot)$ describes whether a subset X is a component of a matroid (recall that the binary relation of two matroid elements being contained in a common circuit is transitive):

$$\psi_C(X) \equiv (\forall x, y \in X : x \neq y \Rightarrow \psi_c(x,y)) \wedge (\forall x \in X, y \notin X : \neg\psi_c(x,y))\,.$$

The sought formula $\psi_d(\cdot)$ is defined inductively as follows (we remark that $\psi_d(\emptyset)$ is true for all d):

$$\begin{aligned} &\psi_1(X) \equiv \forall x, y \in X : x \neq y \Rightarrow \neg\psi_c(x,y) && \text{and} \\ &\psi_d(X) \equiv \forall X' \subseteq X : \psi_C(X') \Rightarrow \exists x \in X' : \psi_{d-1}(X' \setminus \{x\}) && \text{for } d \geq 2. \end{aligned}$$

Hliněný [29, 30] proved that all monadic second order logic properties can be tested in a fixed parameter way for matroids represented over a finite field $\mathbb{F}$ with branch-width at most d when parameterized by the property, the field $\mathbb{F}$ and the branch-width d. Since the branch-width of a matroid M is at most its deletion-depth, it follows that testing whether the deletion-depth of an input matroid represented over a finite field $\mathbb{F}$ is at most d is fixed parameter tractable when parameterized by the field $\mathbb{F}$ and the integer d. This establishes the existence of a fixed parameter algorithm deciding whether $\mathrm{td}^*_P(A) = \mathrm{dd}(M(A)) \le d$ (the equality follows from Theorem 2). To obtain the algorithm claimed to exist in the statement of the corollary, we need to extend the algorithm for testing whether the deletion-depth of an input matroid M represented over $\mathbb{F}$ is at most d to an algorithm that also outputs a deletion-tree of M with height at most d; this would yield an algorithm for computing A' by Lemma 1.

We now describe the extension of the algorithm from testing to constructing a deletion-decomposition tree as a recursive algorithm. Let d be the computed deletion-depth of an input matroid M. The deletion-depth of an input matroid M is one if and only if every element of M is a loop or a bridge. Since the latter is easy to algorithmically test, if $d = 1$, then the deletion-tree of height one consists of a single vertex labeled with all elements of M. If $d \geq 2$ and the matroid M is not connected, we first identify its components, which can be done in polynomial time even in the oracle model, then proceed recursively with each component of M and eventually merge the roots of all deletion-trees obtained recursively.

Finally, we discuss the case when $d \geq 2$ and the matroid M is connected. We loop over all elements e of M and test using the monadic second order checking algorithm of Hliněný [29, 30] whether $\mathrm{dd}(M \setminus e) \leq d-1$, i.e., whether $\psi_{d-1}(M \setminus e)$ is true. Such an element e must exist (otherwise, the deletion-depth of M cannot be d) and when e is found, we recursively apply the algorithm to $M \setminus e$ to obtain a deletion-tree T of $M \setminus e$ with height $d-1$. Note that there is a single recursive call as we invoke recursion for a single element e of the matroid M. The tree T returned by the recursive call is extended to a deletion-tree of M by introducing a new vertex, joining it by an edge to the root of T, rerooting the tree to the new vertex, and labeling the new edge with the element e. This completes the description of the algorithm for constructing a deletion-tree of height at most d if it exists. Observe that the running time of the described procedure for constructing a deletion-tree is bounded by the product of the number of elements of M and the running time of the test whether an input matroid is connected and the test whether an input matroid satisfies $\psi_1(\cdot), \ldots, \psi_d(\cdot)$; in particular, it is fixed parameter when parameterized by a finite field $\mathbb{F}$ and an integer d. ◻

We conclude this section by establishing a link between the optimal incidence tree-depth and the contraction*-deletion-depth of the matroid associated with the matrix.

Proof of Theorem 3 We prove the equality as two inequalities starting with the inequality $\overset{*}{\mathrm{c}}\mathrm{dd}(A) \leq \mathrm{td}_I^*(A) - 1$. To prove this inequality, we show that $\overset{*}{\mathrm{c}}\mathrm{dd}(A) \leq \mathrm{td}_I(A) - 1$ holds for every matrix A with m rows and n columns by induction on $m + n$. The base of the induction is formed by the cases when all entries of A are zero, $n = 1$ or $m = 1$. If all entries of A are zero, then $\overset{*}{\mathrm{c}}\mathrm{dd}(A) = 0$ and $\mathrm{td}_I(A) = 1$. If $n = 1$ and A is non-zero, then $M(A)$ has a single non-loop element and so $\overset{*}{\mathrm{c}}\mathrm{dd}(A) = 1$ while the incidence graph of A is formed by a star and possibly some isolated vertices and so $\mathrm{td}_I(A) = 2$. Finally, if $m = 1$ and A is non-zero, then $M(A)$ has rank 1, so contracting any non-loop element of $M(A)$ yields a matroid of rank zero and so $\overset{*}{\mathrm{c}}\mathrm{dd}(A) = 1$. On the other hand, the incidence graph of A is formed by a star and possibly some isolated vertices and so $\mathrm{td}_I(A) = 2$.

We now establish the induction step, i.e., we assume that the matrix A is non-zero, $m \geq 2$ and $n \geq 2$. First suppose that the matroid $M(A)$ is not connected. Let $X_1, \ldots, X_k$ be the components of $M(A)$ and let $A_1, \ldots, A_k$ be the submatrices of A formed by the columns $X_1, \ldots, X_k$, respectively. The definition of the contraction*-deletion-depth implies that $\overset{*}{\mathrm{c}}\mathrm{dd}(A)$ is the maximum of $\overset{*}{\mathrm{c}}\mathrm{dd}(A_i)$. The induction hypothesis yields that $\overset{*}{\mathrm{c}}\mathrm{dd}(A_i) \leq \mathrm{td}_I(A_i) - 1$. Since the incidence graph of

A_i is a subgraph of the incidence graph of A, it follows that $\mathrm{td}_I(A_i) \leq \mathrm{td}_I(A)$ and so $\mathrm{c}^*\mathrm{dd}(A_i) \leq \mathrm{td}_I(A) - 1$. We conclude that $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{td}_I(A) - 1$.

Next suppose that the matroid $M(A)$ is connected but the incidence graph of A is not connected. As the columns associated with vertices contained in different components of the incidence graph of A have disjoint supports, such columns cannot be contained in the same component of the matroid $M(A)$. Hence, if the incidence graph of A is not connected despite the matroid $M(A)$ being connected, then the incidence graph of A consists of a single non-trivial component and isolated vertices associated with zero rows of A. Let x be one such row and let A' be the matrix obtained from A by deleting the row x. Since the matroids $M(A)$ and $M(A')$ are the same, it holds that $\mathrm{c}^*\mathrm{dd}(A) = \mathrm{c}^*\mathrm{dd}(A')$, and since the incidence graph of A is the incidence graph of A' with an isolated vertex added, it holds $\mathrm{td}_I(A) = \mathrm{td}_I(A')$. Hence, the induction hypothesis yields that $\mathrm{c}^*\mathrm{dd}(A') \leq \mathrm{td}_I(A') - 1$, which implies that $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{td}_I(A) - 1$.

Finally, suppose that the matroid $M(A)$ is connected and the incidence graph of A is also connected. The definition of the tree-depth implies that there exists a vertex w of the incidence graph whose deletion decreases the tree-depth of the incidence graph by one. Let A' be the matrix obtained from A by deleting the row or the column associated with the vertex w and note that $\mathrm{td}_I(A') = \mathrm{td}_I(A) - 1$. If the vertex w is associated with a column x, the matroid $M(A')$ is the matroid obtained from $M(A)$ by deleting the element x. If the vertex w is associated with a row x, the matroid $M(A')$ is the matroid obtained from $M(A)$ by contracting by the subspace generated by the unit vector with the entry in the row x. In either case, the definition of the contraction*-deletion-depth implies that $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{c}^*\mathrm{dd}(A') + 1$. The induction hypothesis applied to A' yields that $\mathrm{c}^*\mathrm{dd}(A') \leq \mathrm{td}_I(A') - 1$, which yields that $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{td}_I(A') = \mathrm{td}_I(A) - 1$.

To complete the proof of the theorem, it remains to show that the inequality $\mathrm{td}_I^*(A) \leq \mathrm{c}^*\mathrm{dd}(A) + 1$ holds for every matrix A. The proof proceeds by induction on the number n of columns of A. If $n = 1$ and the only column of A is zero, then the incidence graph of A is formed by isolated vertices and so $\mathrm{td}_I(A) = 1$ while $\mathrm{c}^*\mathrm{dd}(A) = 0$ since the rank of $M(A)$ is zero. If $n = 1$ and the only column of A is not zero, then the incidence graph of A is formed by a star and possibly some isolated vertices and so $\mathrm{td}_I(A) = 2$ while $\mathrm{c}^*\mathrm{dd}(A) = 1$. In either case, it holds that $\mathrm{td}_I^*(A) \leq \mathrm{td}_I(A) = \mathrm{c}^*\mathrm{dd}(A) + 1$.

We now establish the induction step. First suppose that the matroid $M(A)$ is not connected. Let $X_1, \ldots, X_k$ be the sets of columns forming the components of $M(A)$ and let A' be the matrix row-equivalent to A such that there exist sets of rows $Y_1, \ldots, Y_k$ such that $|Y_i| = r_{M(A)}(X_i)$ for $i = 1, \ldots, k$ and the only columns with non-zero entries in the rows Y_i are those of X_i and all rows not contained in $Y_1 \cup \cdots \cup Y_k$ are zero (such a matrix A' exists since the matroid $M(A)$ is union of its restrictions to $X_1, \ldots, X_k$). Let A'_i be the submatrix of A' formed by the rows of Y_i and the columns of X_i. Observe that all entries of the matrix A not contained in one of the matrices $A'_1, \ldots, A'_k$ are zero. By the induction hypothesis, for every $i = 1, \ldots, k$, there exists a matrix A''_i row-equivalent to A'_i such that $\mathrm{td}_I(A''_i) \leq \mathrm{c}^*\mathrm{dd}(M(A)\,[X_i]) + 1$, in particular, $\mathrm{td}_I(A''_i) \leq \mathrm{c}^*\mathrm{dd}(A) + 1$. Let A'' be the matrix obtained from A' by replacing A'_i with A''_i for $i = 1, \ldots, k$. Observe that A'' is row-equivalent to A' and so to A. Since the incidence graph of A'' is the union of the incidence graphs of A''_i, $i = 1, \ldots, k$, and possibly some isolated vertices (which correspond to zero rows), it follows that $\mathrm{td}_I(A'') \le \overset{*}{\mathrm{c}}\mathrm{dd}(A) + 1$. Hence, it holds that $\mathrm{td}^*_I(A) \le \overset{*}{\mathrm{c}}\mathrm{dd}(A) + 1$.

To complete the proof, we need to consider the case that the matroid $M(A)$ is connected. The definition of the contraction*-deletion-depth implies that there exists an element x of $M(A)$ such that $\overset{*}{\mathrm{c}}\mathrm{dd}(M(A)\backslash x) = \overset{*}{\mathrm{c}}\mathrm{dd}(M(A)) - 1 = \overset{*}{\mathrm{c}}\mathrm{dd}(A) - 1$ or there exists a one-dimensional subspace such that the contraction by this subspace yields a matroid M' such that $\overset{*}{\mathrm{c}}\mathrm{dd}(M') = \overset{*}{\mathrm{c}}\mathrm{dd}(M(A)) - 1 = \overset{*}{\mathrm{c}}\mathrm{dd}(A) - 1$. In the former case, let A' be the matrix obtained from A by deleting the column x. By the induction hypothesis, there exists a matrix A'' row-equivalent to A' such that $\mathrm{td}_I(A'') \le \overset{*}{\mathrm{c}}\mathrm{dd}(A') + 1 = \overset{*}{\mathrm{c}}\mathrm{dd}(A)$, and let A''' be the matrix obtained from A by the same elementary row operations that A'' is obtained from A'. Observe that the incidence graph of A'' can be obtained from the incidence graph of A''' by deleting the vertex associated with the column x. Hence, $\mathrm{td}_I(A''') \le \mathrm{td}_I(A'') + 1$. Since A''' is row-equivalent to A, it follows that

$$\mathrm{td}^*_I(A) \le \mathrm{td}_I(A''') \le \mathrm{td}_I(A'') + 1 \le \overset{*}{\mathrm{c}}\mathrm{dd}(A) + 1.$$

We now analyze the latter case, i.e., the case that there exists a one-dimensional subspace such that the contraction by this subspace yields a matroid M' with $\overset{*}{\mathrm{c}}\mathrm{dd}(M') = \overset{*}{\mathrm{c}}\mathrm{dd}(A) - 1$. Let A' be the matrix obtained from A by elementary row operations such that the contracted subspace used to obtain M' is generated by the unit vector with the non-zero entry being its first entry, and let B be the matrix obtained from A' by deleting the first row. By the induction hypothesis, there exists a matrix B' row-equivalent to B such that $\mathrm{td}_I(B') \le \overset{*}{\mathrm{c}}\mathrm{dd}(A') + 1 = \overset{*}{\mathrm{c}}\mathrm{dd}(A)$, and let A'' be the matrix consisting of the first row of A and the matrix B'. Observe that A'' is row-equivalent to A. Since the incidence graph of B' can be obtained from the incidence graph of A'' by deleting the vertex associated with the first row, it holds that $\mathrm{td}_I(A'') \le \mathrm{td}_I(B') + 1$. Hence, it follows that

$$\mathrm{td}^*_I(A) \le \mathrm{td}_I(A'') \le \mathrm{td}_I(B') + 1 \le \overset{*}{\mathrm{c}}\mathrm{dd}(A) + 1.$$

The proof of the theorem is now completed. □

4 Primal tree-depth

In this section, we present a parameterized algorithm for computing a row-equivalent matrix with small primal tree-depth and bounded entry complexity if such a matrix exists.

Proof of Theorem 6 We first find a bound on $\dot{\kappa}_B$ when B is an integer matrix with $\mathrm{td}_P(B) \le d$ and $\mathrm{ec}(B) \le e$ (recall that $\dot{\kappa}_B$ is the least common multiple of the entries of the circuits of B). Consider such a square invertible matrix C with $\mathrm{td}_P(C) \le d$ and $\mathrm{ec}(C) \le e$. By the result of Brand, Ordyniak and the second author [5], the maximum denominator appearing over all entries of C^{-1} can be bounded by a function of d and e; in particular, there exists $k_0 \le (2^e)^{d!}(d!)^{d!/2}$ such that every entry of C^{-1} is

$1/k$-integral for some $k \le k_0$. Set κ_0 to be the least common multiple of the integers $1, \dots, k_0$. By [21, Theorem 3.8], $\dot{\kappa}_B$ is the smallest integer such that every denominator of the inverse of every invertible square submatrix of B divides $\dot{\kappa}_B$. Since the primal tree-depth of any square submatrix of B at most the primal tree-depth of B, $\dot{\kappa}_B$ divides κ_0 for every matrix B with $\mathrm{td}_P(B) \le d$ and $\mathrm{ec}(B) \le e$. In particular, $\dot{\kappa}_B \le \kappa_0$ for every matrix B with $\mathrm{td}_P(B) \le d$ and $\mathrm{ec}(B) \le e$.

We next describe the algorithm from the statement of the theorem. Without loss of generality, we can assume that the rank of the input matrix A is equal to the number of its rows, in particular, A is non-zero. The algorithm starts with diagonalizing the square submatrix of the input matrix A formed by an arbitrary basis of the column space, i.e., performing elementary row operations so that the selected columns form the identity matrix. The resulting matrix is denoted by A_I. If the numerator or the denominator of any (non-zero) entry of A_I does not divide κ_0, the algorithm arrives at the first conclusion of the theorem as every (non-zero) entry of A_I is a fraction that can be obtained by dividing two entries of a circuit of A (indeed, consider the circuit of the matrix A with support formed by a column x and some of the columns of the chosen basis, and observe that each entry in the column x is equal to the x-entry of the circuit divided by one of its other entries). Hence, we assume that both numerator and denominator of each (non-zero) entry of A_I divides κ_0 in the rest. The algorithm multiplies A_I by κ_0, which yields an integral matrix A_0 with entries between $-\kappa_0^2$ and κ_0^2.

Let $M_{\mathbb{Q}}$ be the column matroid of A_0 when viewed as a matrix over rationals and let M_p be the column matroid of A_0 when viewed as a matrix over a p-element field $\mathbb{F}_p$ for a prime $p > \kappa_0^2$; note that such a prime p can be found algorithmically as the algorithm is parameterized by d and e and κ_0 depends on d and e only. Note that the elements of both matroids $M_{\mathbb{Q}}$ and M_p are the columns of the matrix A_0, i.e., we can assume that their ground sets are the same, and the matroid $M_{\mathbb{Q}}$ is the column matroid of A, which is a matrix over rationals.

We now establish the following claim: *if A is row-equivalent to a matrix with primal tree-depth at most d and entry complexity at most e, then the matroids $M_{\mathbb{Q}}$ and M_p are the same*. If a set X of columns forms a circuit in $M_{\mathbb{Q}}$, then there exists a linear combination of the columns of X that has all coefficients integral and coprime, i.e., not all are divisible by p, and that is equal to the zero vector (in fact, there exist such coefficients that they all divide κ_0 by the definition of κ_0); it follows that the set X is also dependent in M_p. If a set X of columns is independent in $M_{\mathbb{Q}}$, let B_I be an invertible square submatrix of A_I formed by the columns X and $|X|$ rows, and let B_0 be the square submatrix of A_0 formed by the same rows and columns. By [21, Lemma 3.3], the matrix B_I^{-1} is $1/\dot{\kappa}_A$-integral and the absolute value of both numerator and denominator of each entry of B_I^{-1} is at most $\dot{\kappa}_A$. Note that $\dot{\kappa}_A$ divides κ_0 (here, we use the definition of κ_0 and the assumption that A is row-equivalent to a matrix with primal tree-depth at most d and entry complexity at most e) and so the matrix B_I^{-1} is $1/\kappa_0$-integral and the absolute value of both numerator and denominator of each entry of B_I^{-1} is at most κ_0. Let B' be the matrix obtained from B_I^{-1} by multiplying each entry by κ_0; note that B' is an integer matrix with entries between $-\kappa_0^2$ and κ_0^2. The definitions of the matrices B_I, B_0 and B' yield that the product of the matrix B'

when viewed as over $\mathbb{F}_p$ and the matrix B_0 has numerically the same entries as the matrix $(\kappa_0 B_I^{-1})(\kappa_0 B_I)$, which is a diagonal matrix with all diagonal entries equal to κ_0^2. Hence, the matrix B_0 is full rank (over the field $\mathbb{F}_p$) and so the set X is independent in M_p.

We now continue the description of the algorithm that is asserted to exist in the statement of the theorem. As the next step, we apply the algorithm from Corollary 1 to the matrix A_0 viewed as over the field $\mathbb{F}_p$. If the algorithm determines that the deletion-depth of A_0 is larger than d, we arrive at the first conclusion of the theorem: either the matroids $M_{\mathbb{Q}}$ and M_p are different (and so A is not row-equivalent to a matrix with primal tree-depth at most d and entry complexity at most e), or the matroids $M_{\mathbb{Q}}$ and M_p are the same but $\mathrm{dd}(A) = \mathrm{td}^*_P(A) > d$. If the algorithm determines that the deletion-depth of A_0 is at most d, it also outputs a deletion-tree of M_p with height at most d. If the output deletion-tree is not valid for the matroid $M_{\mathbb{Q}}$, which can be verified in polynomial time, the matroids $M_{\mathbb{Q}}$ and M_p are different and we again arrive at the first conclusion of the theorem. If the output deletion-tree is also a deletion-tree of the matroid $M_{\mathbb{Q}}$, we use the algorithm from Lemma 1 to obtain a matrix A' row-equivalent to A such that the primal tree-depth of A' at most the height of the deletion-tree, i.e., $\mathrm{td}_P(A') \leq d$. As the matrix A' contains a unit submatrix formed by m rows and m columns, each (non-zero) entry of A' is a fraction that can be obtained by dividing two entries of a circuit of A (as argued earlier). If the absolute value of the numerator or the denominator of any of these fractions exceeds κ_0, then $c_\infty(A) > \kappa_0$ and we again arrive at the first conclusion of the theorem. Otherwise, we output the matrix A'. Note that the primal tree-depth of A' is at most d and its entry complexity is at most $2\lceil\log_2(\kappa_0 + 1)\rceil$. As κ_0 depends on d and e only, the matrix A' has the properties given in the second conclusion of the theorem. □

5 Dual tree-depth, circuit complexity and Graver basis

In this section, we link minimum dual tree-depth of a matrix to its circuit complexity, and also present related algorithmic results. We start with proving Theorem 5; in fact, we show that a matrix row-equivalent to an input matrix A such that both its dual tree-depth and entry complexity bounded by a function of $c_1(A)$ can be found efficiently. Note that the algorithm presented in the next theorem is not a fixed parameter algorithm, i.e., its running time is polynomial in the size of an input matrix.

Theorem 11 *There exists a polynomial-time algorithm that for a given rational matrix A with* $\dim\ker A > 0$ *outputs a row-equivalent matrix A' such that* $\mathrm{td}_D(A') \leq c_1(A)^2$ *and* $\mathrm{ec}(A') \leq 2\lceil\log_2(c_1(A) + 1)\rceil$.

Proof We start with the description of the algorithm from the statement of the theorem. Let A be the input matrix. We apply the algorithm from Theorem 10 to the matroid $M(A)$ given by the columns of the matrix A. Let T be the principal contraction*-tree output by the algorithm and let X be the set of columns of A that are the labels of the edges of T, i.e., the elements of $M(A)$ used in the contractions. Observe that the definition of the contraction*-depth and the principal contraction*-tree yields that the labels of the edges of T form a basis X of $M(A)$ and for every element z of $M(A)$

there is a leaf v of T such that z is contained in the linear hull of the labels of the edges on the path from v to the root of T. Next perform row-operations on the matrix A in a way that the submatrix formed by the columns of X is an identity matrix (with additional zero rows if the rank of A is smaller than the number of its rows); let A' be the resulting matrix; see Fig. 6 for an example. The algorithm outputs the matrix A'. Note that the running time of the algorithm is indeed polynomial in the size of the input matrix A.

We next analyze the matrix A' that is output by the algorithm. Since $\dim \ker A > 0$, the matrix A has at least one circuit. Recall that for every circuit C of the matroid $M(A)$, there exists a circuit of A whose support is exactly formed by the elements of C. This implies that every circuit of $M(A)$ contains at most $c_1(A)$ elements and so Theorem 10 implies that the depth of the principal contraction*-tree T is at most $c_1(A)^2$.

Let F be a rooted forest obtained from the tree T as follows. For each edge e, the vertex of e farther from the root is identified with the (unique) row of A' that is non-zero in the column that is the label of the edge e (recall that the columns of X form an identity matrix), and then remove the root of T; also add an isolated vertex for each zero row of A'. In this way, we obtain a rooted forest F with vertex set formed by the rows of A'. Note that the height of F is at most $c_1(A)^2$. We will show that the dual graph of A' is a subgraph of $\mathrm{cl}(F)$. As no column of X contributes any edges to the dual graph of A', it is enough to consider columns not contained in X. Let z be a column of A' that is not contained in X and let v be a leaf of T such that the column z of A, which is an element of $M(A)$, is contained in the linear hull of the labels of the edges on the path from v to the root of T. Hence, the column z of A' contains non-zero entries only in the rows with non-zero entries in the columns that are labels of the edges on the path from v to the root of T. Consequently, all edges contained in the dual graph of A' because of non-zero entries in the column z are between vertices on the path from the vertex v in F to the root of the corresponding tree of F. It follows that the dual graph of A' is a subgraph of $\mathrm{cl}(F)$ and so its tree-depth is at most the height of F, i.e., it is at most $c_1(A)^2$. We conclude that $\mathrm{td}_D(A') \leq c_1(A)^2$.

It remains to analyze the entry complexity of A'. The entries of A' in the columns of X are zero or one. Next consider a column z of A' that is not contained in X and consider a circuit c of A' (and so of A) whose support contains z and some elements of X (such a circuit exists as the columns of X form a basis of the column space of A'). Observe that the entries in the column z are equal to $-c_x/c_z$ (otherwise, c would not be a circuit of A'). We conclude that the entry complexity of A' is at most $2\lceil\log_2(c_1(A)+1)\rceil$. $\square$

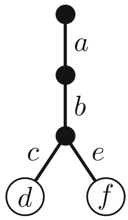
We are now ready to prove Theorem 4. Note that the condition $\dim \mathrm{Ker} A > 0$ in the statement of the theorem is necessary as otherwise A has no circuit and so $c_1(A)$ is not defined.

Proof of Theorem 4 Consider a rational matrix A with $\dim \mathrm{Ker} A > 0$. Note that $c_1(A) \leq g_1(A)$ as every circuit of A is also an element of the Graver basis of A. To prove the existence of the function f_1, let f_D be the function from Theorem 9 and note that Theorem 11 implies that $g_1(A) \leq f_D(c_1(A)^2, 2\lceil\log_2(c_1(A)+1)\rceil)$. $\square$

We next combine the algorithms from Theorems 7 and 11.

$$\begin{array}{cccccc} a & b & c & d & e & f \\ 2 & 0 & 0 & 2 & 0 & 1 \\ -2 & 1 & 0 & 0 & 0 & -1 \\ 2 & 0 & 3 & 5 & 0 & 1 \\ 0 & 1 & 0 & 2 & 1 & 1 \end{array}$$

matrix A



principal contraction*-tree

$$\begin{array}{cccccc} a & b & c & d & e & f \\ 1 & 0 & 0 & 1 & 0 & 1/2 \\ 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

matrix A'

Fig. 6 A rational matrix A, a principal contraction*-tree T of the matroid $M(A)$ and the matrix A' as in the proof of Theorem 11

Corollary 2 *There exists a function $f : \mathbb{N} \to \mathbb{N}$ and a fixed parameter algorithm for the parameterization by k that for a given rational matrix A:*

- *either outputs that $c_1(A) > k$, or*
- *outputs a matrix A' that is row-equivalent to A, its dual tree-depth is $\mathrm{td}^*_D(A)$ and its entry complexity is at most $f(k)$.*

Proof If $\dim \mathrm{Ker} A = 0$ for an input matrix A, i.e., the rank of A is equal to the number of the columns, which can be easily verified in polynomial time, then A is row-equivalent to the unit matrix possibly with some zero rows added, i.e., to a matrix with dual tree-depth one and entry complexity one. If $\dim \mathrm{Ker} A > 0$, we apply the algorithm from Theorem 11 to get a matrix A' row-equivalent to A that has properties given in the statement of Theorem 11. If the dual tree-depth of A' is larger than k^2 or the entry complexity of A' is larger than $2\lceil \log_2(k+1)\rceil$, then $c_1(A) > k$ (by Theorem 11) and we arrive at the first conclusion. Otherwise, we apply the algorithm from Theorem 7 with parameters $d = k^2$ and $e = 2\lceil \log_2(k+1)\rceil$ to compute a matrix A'' row-equivalent to A' and so to A such that the dual tree-depth of A'' is $\mathrm{td}^*_D(A)$ and the entry complexity of A'' is bounded by a function of k only. □

Finally, the previous corollary together with Theorem 9 yields the parameterized algorithm for testing whether an input matrix is row-equivalent to a matrix with small dual tree-depth and small entry complexity as given in Theorem 8.

Proof of Theorem 8 Let f_D be the function from the statement of Theorem 9 and set $k = f_D(d, e)$; note $f_D(d, e) \le 2^{2^{(d \log e)^{O(1)}}}$ by Eisenbrand et al. [20]. Apply the algorithm from Corollary 2 with the parameter k to an input matrix A. If the algorithm reports that $c_1(A) > k$, then A is not row-equivalent to a matrix with dual tree-depth at most d and entry complexity at most e. If the algorithm outputs a matrix A' and $\mathrm{td}_D(A') > d$, then $\mathrm{td}^*_D(A) > d$ and so the matrix A is not row-equivalent to a matrix with dual tree-depth at most d. Otherwise, the dual tree-depth of A' is at most d and its entry complexity is bounded by $f(k) = f(f_D(d, e))$ where $f(\cdot)$ is the function from Corollary 2, i.e., the entry complexity of A' is bounded by a function of d and e only as required. □

6 Computational hardness of depth parameters

In this section, we complement our algorithmic results by establishing computational hardness of matroid depth parameters that we have discussed in this paper. The hardness results apply even when the input matroid is given by its representation over a fixed (finite or infinite) field.

We start with defining a matroid $M_{\mathbb{F}}(G)$ derived from a graph G. Fix a field $\mathbb{F}$. For a graph G, we define an $\mathbb{F}$-represented matroid $M_{\mathbb{F}}(G)$ as follows. The matroid $M_{\mathbb{F}}(G)$ contains $|V(G)| + |E(G)|$ elements, each corresponding to a vertex or an edge of G. We associate each element of $M_{\mathbb{F}}(G)$ with a vector of $\mathbb{F}^{V(G)}$. An element of $M_{\mathbb{F}}(G)$ corresponding to a vertex w of G is represented by e_w and an element of $M_{\mathbb{F}}(G)$ corresponding to an edge ww' of G is represented by $e_w - e_{w'}$ or $e_{w'} - e_w$ (an arbitrary of the two vectors can be chosen as the choice does not affect the matroid).

We next define a graph G/A for a graph G and a linear subspace A of $\mathbb{F}^{V(G)}$. Let W be the subset of vertices of $V(G)$ such that $e_w \in A$ for $w \in W$, and let F be the set of edges ww' of G such that neither w nor w' is contained in W and A contains a vector $e_w + \alpha e_{w'}$ for a non-zero element $\alpha \in \mathbb{F}$. The graph G/A is obtained by deleting all vertices of W and then contracting a maximal acyclic subset of edges contained in F (the remaining edges of F become loops and so get removed); note that we use an acyclic subset of edges for contraction so that the resulting graph is well-defined.

The next lemma relates the number of components of the matroid $M_{\mathbb{F}}(G)/A$ and the number of components of the graph G/A for a graph G and a linear subspace A of $\mathbb{F}^{V(G)}$.

Lemma 2 *Let G be a graph, $\mathbb{F}$ a field and A a linear subspace of $\mathbb{F}^{V(G)}$. The number of components of $M_{\mathbb{F}}(G)/A$ is at most the number of components of the graph G/A.*

Proof Fix a graph G, a field $\mathbb{F}$ and a linear subspace of $\mathbb{F}^{V(G)}$. Let W and F be the subsets of vertices and edges of G as in the definition of G/A, respectively. Let A' be the linear subspace of $\mathbb{F}^{V(G)}$ generated by the vectors e_w, $w \in W$, and the vectors $e_w + \alpha e_{w'} \in A$ for $ww' \in F$; clearly, A' is a subspace of A. Note that the sets W and F defined with respect to A would be the same if defined with respect to A', and so the graphs G/A and G/A' are the same. We next describe an $\mathbb{F}$-representation of the matroid $M_{\mathbb{F}}(G)/A'$ using vectors of $\mathbb{F}^{V(G/A')}$. The matroid $M_{\mathbb{F}}(G)$ contains elements corresponding to vertices and to edges of G. Consider a vertex w of $V(G)$. If $w \in W$, then the element corresponding to w is a loop in $M_{\mathbb{F}}(G)/A'$ and so represented by the zero vector. If $w \notin W$, then the element corresponding to w is represented by the vector e_u where u is the vertex of G/A' that the vertex w was contracted to. Next consider an edge ww' of G.

- If both w and w' belong to W, then the element corresponding to ww' is a loop in $M_{\mathbb{F}}(G)/A'$ and so represented by the zero vector.
- If exactly one of w and w' belong to W, say $w \in W$ and $w' \notin W$, then the element corresponding to ww' is represented by the vector e_u where u is the vertex of G/A' that the vertex w' was contracted to.
- If neither w nor w' belongs to W and $e_w - e_{w'} \in A'$ (and so $ww' \in F$), then the element corresponding to ww' is a loop in $M_{\mathbb{F}}(G)/A'$ and so represented by the zero vector.

- If neither w nor w' belongs to W, $e_w - e_{w'} \notin A'$ but $ww' \in F$, then the element corresponding to ww' is represented by the vector e_u where u is the vertex of G/A' that the vertex w (and so the vertex w') was contracted to.
- Finally, if neither w nor w' belongs to W and $ww' \notin F$, the element corresponding to ww' is the vector $\alpha e_u + \alpha' e_{u'}$ where u is the vertex of G/A' that the vertex w was contracted to, u' is the vertex of G/A' that the vertex w' was contracted to, and the vector $\alpha e_u + \alpha' e_{u'}$ corresponds to the vector $e_w - e_{w'}$ in $\mathbb{F}^{V(G)}/A'$; note that uu' is an edge of G/A' and both coefficients α and α' are non-zero.

It is straightforward to verify that the just described representation is indeed a representation of the matroid $M_{\mathbb{F}}(G)/A'$; note that each non-loop element of $M_{\mathbb{F}}(G)/A'$ is associated to a vertex or an edge of G/A' and each vertex or an edge of G/A' with at least one (but possibly more) non-loop element of $M_{\mathbb{F}}(G)/A'$.

We now analyze the matroid $M_{\mathbb{F}}(G)/A$. The matroid $M_{\mathbb{F}}(G)/A$ can be viewed as the matroid $(M_{\mathbb{F}}(G)/A')/(A/A')$. Observe that the definition of A' implies that any non-loop element of $M_{\mathbb{F}}(G)/A'$ is non-loop in $M_{\mathbb{F}}(G)/A$. Also observe that the space A/A' viewed as a subspace of $\mathbb{F}^{V(G/A')}$ does not contain a vector e_w for $w \in V(G/A')$ or a vector $e_w + \alpha e_{w'}$ for a non-zero $\alpha \in \mathbb{F}$ such that ww' is an edge of $E(G/A')$. In particular, the support of every vector of A/A' is at least two and if the support has size two, then it does not correspond to an edge of $E(G/A')$. It follows that if ww' is an edge of $E(G/A')$, x is an element of $M_{\mathbb{F}}(G)/A'$ associated with the vertex w, x' is an element of $M_{\mathbb{F}}(G)/A'$ associated with the vertex w', and x'' is an element of $M_{\mathbb{F}}(G)/A'$ associated with the edge ww', then the elements x, x' and x'' form a circuit of $(M_{\mathbb{F}}(G)/A')/(A/A')$. Since the relation of being contained in a common circuit is transitive, it follows that all elements of $M_{\mathbb{F}}(G)/A'$ corresponding to the vertices and the edges of the same component of G/A' are contained in the same component of $(M_{\mathbb{F}}(G)/A')/(A/A')$. In particular, the number of components of the matroid $(M_{\mathbb{F}}(G)/A')/(A/A')$ is at most the number of components of the graph G/A'. Since the graphs G/A and G/A' are the same and the matroids $M_{\mathbb{F}}(G)/A$ and $(M_{\mathbb{F}}(G)/A')/(A/A')$ are also the same, the lemma follows. □

We next link the existence of a balanced independent set in a bipartite graph to the contraction*-depth of a suitably defined matroid. We remark that the idea of using a bipartite graph with cliques added between the vertices of its parts was used in [47] to establish that computing tree-depth of a graph is NP-complete.

Lemma 3 *Let G be a bipartite graph with parts X and Y, let $\mathbb{F}$ be a field, and let k be an integer. Let G' be the graph obtained from G by adding all edges between the vertices of X and between the vertices of Y. The following three statements are row-equivalent.*

- *The graph G has an independent set containing k elements of X and k elements of Y.*
- *The contraction*-depth of $M_{\mathbb{F}}(G')$ is at most $|X| + |Y| - k$.*
- *The contraction-depth of the matroid $2M_{\mathbb{F}}(G')$ is at most $|X| + |Y| - k + 1$.*

Proof Fix a bipartite graph G with parts X and Y, a field $\mathbb{F}$ and an integer k. We first show that if G has an independent set containing k elements of X and k elements of

Y, then the contraction*-depth of $M_{\mathbb{F}}(G')$ is at most $|X|+|Y|-k$ and the contraction-depth of $2M_{\mathbb{F}}(G')$ is at most $|X|+|Y|-k+1$. Let W be such an independent set and let W' be the set containing all elements e_w of $M_{\mathbb{F}}(G')$ such that $w \notin W$. Note that $|W'| = |X| + |Y| - 2k$. The matroid $M_{\mathbb{F}}(G')/W'$ is the matroid obtained from $M_{\mathbb{F}}(G'[W])$ by adding

- a loop for every edge with both end vertices not contained in W, and
- an element represented by e_w for every edge joining a vertex $w \in W$ to a vertex not contained in W.

In particular, the matroid $M_{\mathbb{F}}(G')/W'$ has two non-trivial components, each of rank k, and so the contraction*-depth of $M_{\mathbb{F}}(G')$ is at most $|W'| + k = |X| + |Y| - k$. Similarly, the matroid $(2M_{\mathbb{F}}(G'))/W'$ is the matroid obtained from $2M_{\mathbb{F}}(G'[W])$ by adding

- a loop for every vertex not contained in W,
- two loops for every edge with both end vertices not contained in W, and
- two elements represented by e_w for every edge joining a vertex $w \in W$ to a vertex not contained in W.

Since the matroid $(2M_{\mathbb{F}}(G'))/W'$ has two non-trivial components, each of rank k, its contraction-depth is at most $|W'| + k + 1 = |X| + |Y| - k + 1$ (note that any rank r matroid has contraction-depth at most $r + 1$).

We next argue that if the contraction*-depth of $M_{\mathbb{F}}(G')$ is at most $|X| + |Y| - k$ or the contraction-depth of $2M_{\mathbb{F}}(G')$ is at most $|X| + |Y| - k + 1$, then there exists a subset W of $V(G) = V(G')$ that is independent in G, $|W \cap X| \geq k$ and $|W \cap Y| \geq k$. To do so, we first show that there is no linear subspace A such that $M_{\mathbb{F}}(G')/A$ would have more than two components. Consider a linear subspace A of $\mathbb{F}^{V(G')}$ such that the matroid $M_{\mathbb{F}}(G')/A$ is not connected. By Lemma 2, the graph G'/A is disconnected. Since the graph G'/A cannot have more than two components (one is formed by some of the vertices of X and another by some of the vertices of Y), it follows that the graph G'/A has exactly two components and so the matroid $M_{\mathbb{F}}(G')/A$ has exactly two components, too.

If the contraction*-depth of $M_{\mathbb{F}}(G')$ is at most $|X| + |Y| - k$, there exists a linear subspace A of $\mathbb{F}^{V(G')}$ such that the matroid $M_{\mathbb{F}}(G')/A$ is not connected and the rank of each of its two components is at most $|X| + |Y| - k - \dim A$. We will prove that the existence of such A is also implied by the assumption that the contraction-depth of $2M_{\mathbb{F}}(G')$ is at most $|X|+|Y|-k+1$. We next use that the matroid $(2M_{\mathbb{F}}(G'))/F$ has at most two non-trivial components for every subset F of the elements of $2M_{\mathbb{F}}(G')$. If the contraction-depth of $2M_{\mathbb{F}}(G')$ is at most $|X|+|Y|-k+1$, then there exists a subset F of the elements of $2M_{\mathbb{F}}(G')$ such that the matroid $(2M_{\mathbb{F}}(G'))/F$ is not connected and the rank of each of its two components is at most $|X| + |Y| - k - \operatorname{rank} F$ (as the contraction-depth of each of its two components is the rank of the component increased by one because each element is parallel to at least one other element). It follows that there exists a linear subspace A of $\mathbb{F}^{V(G')}$, which is the hull of the vectors representing the elements of the set F as above, such that the matroid $M_{\mathbb{F}}(G')/A$ is not connected and the rank of each of its two components is at most $|X| + |Y| - k - \dim A$. We conclude that if the contraction*-depth of $M_{\mathbb{F}}(G')$ is at most $|X| + |Y| - k$ or if the contraction-depth of $2M_{\mathbb{F}}(G')$ is at most $|X| + |Y| - k + 1$, then there exists a linear subspace A of $\mathbb{F}^{V(G')}$ such that the matroid $M_{\mathbb{F}}(G')/A$ is not connected and the rank of each of its two components is at most $|X| + |Y| - k - \dim A$.

It remains to show that *the existence of a subspace A of $\mathbb{F}^{V(G')}$ such that the matroid $M_{\mathbb{F}}(G')/A$ is not connected and the rank of each of its two components is at most $|X| + |Y| - k - \dim A$ implies that there exists an independent set containing k elements of X and k elements of Y*. Fix such a subspace A. Let W be the set of vertices w such that e_w is contained in A and let A_W be the subspace of A generated by the vectors e_w, $w \in W$. Since G'/A is not connected, the graph $G' \setminus W$ is also not connected (recall that G'/A is obtained by removing the vertices of W and then contracting some edges). By Lemma 2 the matroid $M_{\mathbb{F}}(G')/A_W$ is also not connected. Since the space A_W is a subspace of A, the rank of each component of $M_{\mathbb{F}}(G')/A_W$ is larger by at most $\dim A - \dim A_W$ compared to the corresponding component of $M_{\mathbb{F}}(G')/A$. Hence, the rank of each of the two components of $M_{\mathbb{F}}(G')/A_W$ is at most $|X| + |Y| - k - \dim A_W = |X| + |Y| - k - |W|$. It follows that each component of the graph $G'/A_W = G' \setminus W$ contains at most $|X| + |Y| - k - |W|$ vertices. Since the sum of the sizes of the two components of $G' \setminus W$ is $|X| + |Y| - |W|$, each component of $G' \setminus W$ has at least k vertices. In addition, the vertex set of each component of $G' \setminus W$ is either a subset of X or a subset of Y, which implies that there is no edge joining a vertex of $X \setminus W$ and a vertex of $Y \setminus W$ and both sets $X \setminus W$ and $Y \setminus W$ have at least k vertices. Hence, the graph G has an independent set containing k elements of X and k elements of Y (such an independent set is a subset of $V(G) \setminus W$). □

We are now ready to state our hardness result.

Theorem 12 *For every field $\mathbb{F}$, each of the following five decision problems, whose input is an $\mathbb{F}$-represented matroid M and an integer d, is NP-complete:*

- *Is the contraction-depth of M at most d?*
- *Is the contraction*-depth of M at most d?*
- *Is the contraction-deletion-depth of M at most d?*
- *Is the contraction*-deletion-depth of M at most d?*
- *Is the deletion-depth of M at most d?*

Proof It is NP-complete to decide for a bipartite graph G with parts X and Y and an integer k whether there exist k-element subsets $X' \subseteq X$ and $Y' \subseteq Y$ such that $X' \cup Y'$ is independent [47]. For an input bipartite graph G, let G' be the graph obtained from G by adding all edges between the vertices of X and between the vertices of Y. We claim that the existence of such subsets X' and Y' is equivalent to each of the following four statements:

- The matroid $2M_{\mathbb{F}}(G')$ has contraction-depth at most $|X| + |Y| - k + 1$.
- The matroid $M_{\mathbb{F}}(G')$ has contraction*-depth at most $|X| + |Y| - k$.
- The matroid $(|V(G')| + 1)M_{\mathbb{F}}(G')$ has contraction-deletion-depth at most $|X| + |Y| - k + 1$.
- The matroid $(|V(G')| + 1)M_{\mathbb{F}}(G')$ has contraction*-deletion-depth at most $|X| + |Y| - k$.

The equivalences to the first and second statements follow directly from Lemma 3. Since the rank of the matroid $(|V(G')|+1)M_{\mathbb{F}}(G')$ is $|G'|$, its contraction-deletion-depth is at most $|G'|+1$ and its contraction*-deletion-depth is at most $|G'|$. As each element of the matroid $(|V(G')|+1)M_{\mathbb{F}}(G')$ is parallel to (at least) $|V(G')|$ elements of the matroid, it follows that the contraction-deletion-depth of $M_{\mathbb{F}}(G')$ is the same as its contraction-depth and its contraction*-deletion-depth is the same as its contraction*-depth. Lemma 3 now implies the equivalence of the third and fourth statements. As the matroids $2M_{\mathbb{F}}(G')$, $M_{\mathbb{F}}(G')$ and $(|V(G')|+1)M_{\mathbb{F}}(G')$ can be easily constructed from the input graph G in time polynomial in $|V(G)|$, the NP-completeness of the first four problems listed in the statement of the theorem follows.

For an $\mathbb{F}$-represented matroid M, it is easy to construct an $\mathbb{F}$-represented matroid M^* that is dual to M in time polynomial in the number of the elements of M [46, Chapter 2]. Since the contraction-depth of M is equal to the deletion-depth of M^*, it follows that the fifth problem listed in the statement of the theorem is also NP-complete. □

7 Concluding remarks

We would like to conclude with addressing three natural questions related to the work presented in this paper.

In Sect. 5, we have given a structural characterization of matrices A with $g_1(A)$ bounded by showing that $g_1(A)$ is bounded if and only if A is row-equivalent to a matrix with small dual tree-depth and small entry complexity. Unfortunately, a similar (if and only if) characterization of matrices A with $g_\infty(A)$ bounded does not seem to be in our reach.

Problem 1 Find a structural characterization of matrices A with $g_\infty(A)$ bounded.

In view of Theorem 9, it may be tempting to think that such a characterization can involve matrices with bounded incidence tree-depth as if a matrix A has bounded primal tree-depth or it has bounded dual tree-depth, then $g_\infty(A)$ is bounded. However, the following matrix A has incidence tree-depth equal to 4 and yet $g_\infty(A)$ grows with the number t of its columns; in particular, the vector $(t-1, 1, 1, \dots, 1)$ is an element of its Graver basis as it can be readily verified. We remark that a similar matrix was used by Eiben et al. [18] in their NP-completeness argument.

$$\begin{pmatrix} 1 & -1 & -1 & -1 & \cdots & -1 & -1 \\ 0 & 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \ddots & & \vdots \\ 0 & 1 & 0 & 0 & \cdots & -1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & -1 \end{pmatrix}$$

In Sect. 3, we have given structural characterizations of matrices that are row-equivalent to a matrix with small primal tree-depth or small incidence tree-depth, which complements the characterization of matrices row-equivalent to a matrix with small dual tree-depth from [8, 9]. We have also presented fixed parameter algorithms (Theorems 6 and 8) for finding such a row-equivalent matrix with bounded entry complexity if one exists; both of these algorithms are based on fixed parameter algorithms for finding deletion-depth decompositions and contraction*-depth decompositions of matroids over finite fields, which are presented in Corollary 1 in the case of deletion-depth and in [8, 9] in the case of contraction*-depth. We believe that similar techniques would lead to a fixed parameter algorithm for contraction*-deletion-depth decompositions of matroids represented over a finite field (note that contraction*-deletion-depth does not have an obvious description in monadic second order logic and so the algorithmic results of Hliněný [29, 30] do not readily apply in this setting). However, it is unclear whether such an algorithm would yield a fixed parameter algorithm for rational matrices as we do not have structural results on the circuits of rational matrices with small incidence tree-depth, which would reduce the case of rational matrices to those over finite fields.

Another natural question is whether the upper bound on the depth of the principal contraction*-tree given in Theorem 10, which is quadratic in the length of the longest circuit of a represented matroid, can be improved. However, this turns out to be impossible as we now argue. Since the minimum depth of a principal contraction*-tree of a matroid M differs from $\mathrm{cd}(M)$, i.e. the minimum height of a contraction-tree of M, by at most one, it is enough to construct a sequence of matroids M_n such that

- the length of the longest circuit of M_n is is at most $\mathcal{O}(n)$, and
- the contraction-depth of M_n is at least $\Omega(n^2)$.

Hence, the quadratic dependence of the minimum depth in Theorem 10 is optimal up to a constant factor. Still, it can be the case that the bound on the contraction*-depth can be improved.

The matroids M_n are the graphic matroids of graphs G_n, which are constructed inductively. To facilitate the induction we will require slightly stronger properties. Each of the graphs G_n contains two distinguished vertices, denoted by r_n and b_n, and the following holds:

1. The length of any path in G_n between the vertices r_n and b_n is between n and $2n$.
2. The length of any circuit in M_n is at most $4n$.
3. The contraction-depth of M_n is at least $\binom{n}{2}$.

If $n = 1$, we set G_1 to be the two-vertex graph formed by two parallel edges, and r_1 and b_1 are chosen as the two vertices of G_1. Note that the graph G_1 and the matroid $M_1 = M(G_1)$ has the properties (7), (7), and (7). To obtain G_n, we start with a cycle of length $2n$ and choose any two vertices at distance n to be r_n and b_n. This cycle containing the vertices r_n and b_n will be referred to as the *root cycle*. We then add n copies of G_{n-1}, connect the vertex r_{n-1} in each copy to the vertex r_n, and connect the vertex b_{n-1} in each copy to b_n. The construction is illustrated in Fig. 7.

Assuming that the matroid M_{n-1} and the graph G_{n-1} have the properties (7), (7), and (7), and we will show that the matroid M_n and the graph G_n also have these properties.

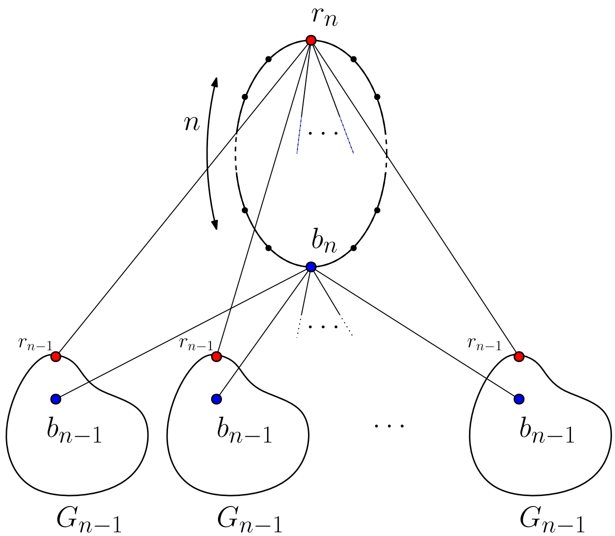


Fig. 7 The construction of the graph G_n. The vertices r_n and r_{n-1} are drawn red while b_n and b_{n-1} are drawn blue

We start with showing that G_n has the property (7). Indeed, a path from r_n to b_n is either contained in the root cycle or consists of a path between r_{n-1} and b_{n-1} in one of the copies of G_{n-1}, whose length is between $n-1$ and $2(n-1)$, together with two edges joining r_{n-1} to r_n and b_{n-1} to b_n. In either of the cases, the length of the path is between n and $2n$ as required.

Having established the property (7), we prove the property (7). Any circuit of the matroid M_n corresponds to a cycle in the graph G_n, thus we can simply investigate the lengths of cycles in G_n. First observe that a cycle of G_n contains either both vertices r_n and b_n or neither of them. Every cycle containing r_n and b_n consists of two paths between r_n and b_n and so its length is at most $4n$, and every cycle containing neither r_n nor b_n is contained entirely within a copy of G_{n-1} and so its length is at most $4(n-1) \leq 4n$.

Finally, we argue that contraction-depth of M_n is at least $\binom{n}{2}$. Recall that contracting an element of M_n corresponds to contracting the associated edge in the graph G_n, and components of a graphic matroid correspond to blocks, i.e., maximal 2-edge-connected components, of an associated graph. Since the length of any path between r_n to b_n is at least n, until at least n edge contractions are performed in graph G_n, the vertices r_n and b_n are distinct and are contained in the same block. Hence, after $n-1$ edge contractions followed by deleting all blocks not containing the vertices r_n and b_n (if such blocks appear), the graph still contains an intact copy G_{n-1}. It follows that $\mathrm{cd}(M_n) \geq (n-1) + \mathrm{cd}(M_{n-1})$, which implies that $\mathrm{cd}(M_n) \geq (n-1) + \binom{n-1}{2} = \binom{n}{2}$.

Acknowledgements All five authors would like to thank the Schloss Dagstuhl—Leibniz Center for Informatics for hospitality during the workshop "Sparsity in Algorithms, Combinatorics and Logic" in September 2021 where the work leading to the results contained in this paper was started. The authors are also indebted to the two anonymous reviewers for their detailed comments on the manuscript, which have helped to improve the presentation significantly and made the manuscript more accessible to a wider audience.

Funding Open access publishing supported by the National Technical Library in Prague.

References

1. Aschenbrenner, M., Hemmecke, R.: Finiteness theorems in stochastic integer programming. Found. Comput. Math. **7**(2), 183–227 (2007)
2. Aykanat, C., Pinar, A., Çatalyürek, Ü.V.: Permuting sparse rectangular matrices into block-diagonal form. SIAM J. Sci. Comput. **25**(6), 1860–1879 (2004)
3. Bergner, M., Caprara, A., Ceselli, A., Furini, F., Lübbecke, M.E., Malaguti, E., Traversi, E.: Automatic Dantzig–Wolfe reformulation of mixed integer programs. Math. Program. **149**(1–2), 391–424 (2015)
4. Borndörfer, R., Ferreira, C.E., Martin, A.: Decomposing matrices into blocks. SIAM J. Optim. **9**(1), 236–269 (1998)
5. Brand, C., Koutecký, M., Ordyniak, S.: Parameterized algorithms for MILPs with small treedepth. Proc. AAAI Conf. Artif. Intell. **35**(14), 12249–12257 (2021)
6. Bredereck, R., Figiel, A., Kaczmarczyk, A., Knop, D., Niedermeier, R.: High-multiplicity fair allocation made more practical. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS'21, pp. 260–268. International Foundation for Autonomous Agents and Multiagent Systems (2021)
7. Bredereck, R., Kaczmarczyk, A., Knop, D., Niedermeier, R.: High-multiplicity fair allocation: Lenstra empowered by n-fold integer programming. In: Proceedings of the 2019 ACM Conference on Economics and Computation, EC'19, pp. 505–523. Association for Computing Machinery (2019)
8. Chan, T.F.N., Cooper, J.W., Koutecký, M., Král', D., Pekárková, K.: Matrices of optimal tree-depth and row-invariant parameterized algorithm for integer programming. In: 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020), Leibniz International Proceedings in Informatics (LIPIcs), vol. 168, pp. 26:1–26:19 (2020)
9. Chan, T.F.N., Cooper, J.W., Koutecký, M., Král', D., Pekárková, K.: Matrices of optimal tree-depth and a row-invariant parameterized algorithm for integer programming. SIAM J. Comput. **51**(3), 664–700 (2022)
10. Chen, H., Chen, L., Zhang, G.: FPT algorithms for a special block-structured integer program with applications in scheduling. preprint arXiv:2107.01373 (2021)
11. Chen, L., Marx, D.: Covering a tree with rooted subtrees–parameterized and approximation algorithms. In: 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2018), pp. 2801–2820. SIAM (2018)
12. Cslovjecsek, J., Eisenbrand, F., Hunkenschröder, C., Rohwedder, L., Weismantel, R.: Block-structured integer and linear programming in strongly polynomial and near linear time. In: 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2021), pp. 1666–1681. SIAM (2021)
13. Cslovjecsek, J., Eisenbrand, F., Pilipczuk, M., Venzin, M., Weismantel, R.: Efficient Sequential and Parallel Algorithms for Multistage Stochastic Integer Programming Using Proximity. In: 29th Annual European Symposium on Algorithms (ESA 2021), Leibniz International Proceedings in Informatics (LIPIcs), vol. 204, pp. 33:1–33:14 (2021)
14. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Berlin (2015)
15. De Loera, J.A., Hemmecke, R., Onn, S., Weismantel, R.: N-fold integer programming. Discret. Optim. **5**(2), 231–241 (2008)
16. DeVos, M., Kwon, O., Oum, S.: Branch-depth: Generalizing tree-depth of graphs. Eur. J. Comb. **90**, 103186 (2020)

17. Ding, G., Oporowski, B., Oxley, J.: On infinite antichains of matroids. J. Comb. Theory Ser. B **63**(1), 21–40 (1995)
18. Eiben, E., Ganian, R., Knop, D., Ordyniak, S., Pilipczuk, M., Wrochna, M.: Integer programming and incidence tree depth. In: Integer Programming and Combinatorial Optimization—20th International Conference (IPCO), LNCS vol. 11480, pp. 194–204. Springer International Publishing (2019)
19. Eisenbrand, F., Hunkenschröder, C., Klein, K.: Faster algorithms for integer programs with block structure. In: 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018), Leibniz International Proceedings in Informatics (LIPIcs), pp. 49:1–49:13 (2018)
20. Eisenbrand, F., Hunkenschröder, C., Klein, K., Koutecký, M., Levin, A., Onn, S.: An algorithmic theory of integer programming. preprint arXiv:1904.01361 (2019)
21. Ekbatani, F., Natura, B., Végh, L.A.: Circuit imbalance measures and linear programming. preprint arXiv:2108.03616 (2021)
22. Ferris, M.C., Horn, J.D.: Partitioning mathematical programs for parallel solution. Math. Program. **80**(1), 35–61 (1998)
23. Gamrath, G., Lübbecke, M.E.: Experiments with a generic Dantzig–Wolfe decomposition for integer programs. Exp. Algorithms **6049**, 239–252 (2010)
24. Ganian, R., Ordyniak, S.: The complexity landscape of decompositional parameters for ILP. Artif. Intell. **257**, 61–71 (2018)
25. Halmos, P.: Finite-Dimensional Vector Spaces. Undergraduate Texts in Mathematics, Springer, Berlin (1993)
26. Hemmecke, R., Onn, S., Romanchuk, L.: N-fold integer programming in cubic time. Math. Program. **137**, 325–341 (2013)
27. Hemmecke, R., Schultz, R.: Decomposition of test sets in stochastic integer programming. Math. Program. **94**, 323–341 (2003)
28. Hermelin, D., Molter, H., Niedermeier, R., Shabtay, D.: Equitable scheduling for the total completion time objective. preprint arXiv:2112.13824 (2021)
29. Hliněný, P.: Branch-width, parse trees, and monadic second-order logic for matroids. In: H. Alt, M. Habib (eds.) 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS), LNCS, vol. 2607, pp. 319–330 (2003)
30. Hliněný, P.: Branch-width, parse trees, and monadic second-order logic for matroids. J. Comb. Theory Ser. B **96**(3), 325–351 (2006)
31. Jansen, K., Klein, K., Lassota, A.: The double exponential runtime is tight for 2-stage stochastic ILPs. In: M. Singh, D.P. Williamson (eds.) Integer Programming and Combinatorial Optimization—22nd International Conference (IPCO), LNCS vol. 12707, Lecture Notes in Computer Science, vol. 12707, pp. 297–310. Springer (2021)
32. Jansen, K., Klein, K., Maack, M., Rau, M.: Empowering the configuration-IP: new PTAS results for scheduling with setup times. Math. Program. (2021)
33. Jansen, K., Lassota, A., Maack, M.: Approximation algorithms for scheduling with class constraints. In: Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures, pp. 349–357. Association for Computing Machinery (2020)
34. Jansen, K., Lassota, A., Rohwedder, L.: Near-linear time algorithm for n-fold ILPs via color coding. SIAM J. Discret. Math. **34**(4), 2282–2299 (2020)
35. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
36. Kardoš, F., Kráľ, D., Liebenau, A., Mach, L.: First order convergence of matroids. Eur. J. Comb. **59**, 150–168 (2017)
37. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds.) Complexity of Computer Computations. The IBM Research Symposia Series, pp. 85–103. Springer, Berlin (1972)
38. Khaniyev, T., Elhedhli, S., Erenay, F.S.: Structure detection in mixed-integer programs. INFORMS J. Comput. **30**(3), 570–587 (2018)
39. Klein, K.: About the complexity of two-stage stochastic IPs. Math. Program. **192**(1), 319–337 (2022)
40. Klein, K., Reuter, J.: Collapsing the tower—on the complexity of multistage stochastic IPs. In: 33rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2022), pp. 348–358. SIAM (2022)
41. Knop, D., Koutecký, M.: Scheduling kernels via configuration LP. preprint arXiv:2003.02187 (2018)
42. Knop, D., Koutecký, M.: Scheduling meets n-fold integer programming. J. Sched. **21**(5), 493–503 (2018)

43. Knop, D., Koutecký, M., Mnich, M.: Combinatorial n-fold integer programming and applications. In: 25th Annual European Symposium on Algorithms (ESA), Leibniz International Proceedings in Informatics (LIPIcs), vol. 87, pp. 54:1–54:14 (2017)
44. Koutecký, M., Levin, A., Onn, S.: A parameterized strongly polynomial algorithm for block structured integer programs. In: 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018), Leibniz International Proceedings in Informatics (LIPIcs), vol. 107, pp. 85:1–85:14 (2018)
45. Lenstra, H.W., Jr.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
46. Oxley, J.: Matroid Theory. Oxford Graduate Texts in Mathematics, Oxford University Press, Oxford (2011)
47. Pothen, A.: The complexity of optimal elimination trees. Technical Report CS-88-13, Pennsylvania State University (1988)
48. Schultz, R., Stougie, L., van der Vlerk, M.H.: Solving stochastic programs with integer recourse by enumeration: a framework using gröbner basis reductions. Math. Program. **83**, 229–252 (1998)
49. Vanderbeck, F., Wolsey, L.A.: Reformulation and decomposition of integer programs. In: 50 Years of Integer Programming 1958–2008, pp. 431–502. Springer (2010)
50. Wang, J., Ralphs, T.: Computational experience with hypergraph-based methods for automatic decomposition in discrete optimization. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, pp. 394–402. Springer (2013)
51. Weil, R.L., Kettler, P.C.: Rearranging matrices to block-angular form for decomposition (and other) algorithms. Manag. Sci. **18**(1), 98–108 (1971)

Authors and Affiliations

Marcin Briański[1] · Martin Koutecký[2] · Daniel Král'[3] · Kristýna Pekárková[3] · Felix Schröder[4]

Marcin Briański
marcin.brianski@doctoral.uj.edu.pl

Martin Koutecký
koutecky@iuuk.mff.cuni.cz

Daniel Král'
dkral@fi.muni.cz

Felix Schröder
schroder@kam.mff.cuni.cz

[1] Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

[2] Computer Science Institute, Charles University, Prague, Czech Republic

[3] Faculty of Informatics, Masaryk University, Brno, Czech Republic

[4] Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic