# Characterization of Matrices with Bounded Graver Bases and Depth Parameters and Applications to Integer Programming

**Marcin Briański**
Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

**Martin Koutecký**
Computer Science Institute, Charles University, Prague, Czech Republic

**Daniel Král'**
Faculty of Informatics, Masaryk University, Brno, Czech Republic

**Kristýna Pekárková**
Faculty of Informatics, Masaryk University, Brno, Czech Republic

**Felix Schröder**
Institute of Mathematics, Technische Universität, Berlin, Germany

## 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 an equivalent sparse matrix if it exists and provide structural results characterizing the existence of a sparse equivalent matrix in terms of the structural properties of the associated column matroid. In particular, our results imply that the $\ell_1$-norm of the Graver basis is bounded by a function of the maximum $\ell_1$-norm of a circuit of $A$. We use our results to design a parameterized algorithm that constructs a matrix equivalent to an input matrix $A$ that has small primal/dual tree-depth and entry complexity if such an equivalent matrix exists.

Our results yield parameterized algorithms for integer programming when parameterized by the $\ell_1$-norm of the Graver basis of the constraint matrix, when parameterized by the $\ell_1$-norm of the circuits of the constraint matrix, when parameterized by the smallest primal tree-depth and entry complexity of a matrix equivalent to the constraint matrix, and when parameterized by the smallest dual tree-depth and entry complexity of a matrix equivalent to the constraint matrix.

**2012 ACM Subject Classification** Mathematics of computing → Combinatorial optimization; Mathematics of computing → Matroids and greedoids; Mathematics of computing → Combinatorial algorithms

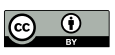
**Keywords and phrases** Integer programming, width parameters, matroids, Graver basis, tree-depth, fixed parameter tractability

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2022.29

**Category** Track A: Algorithms, Complexity and Games

**Related Version** *Full Version*: https://arxiv.org/abs/2202.05299

**Funding** 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. The second author was partially supported by Charles University project UNCE/SCI/004 and by the project by the project 19-27871X of GA ČR. First author was partially supported by the Polish National Science Center grant (BEETHOVEN; UMO-2018/31/G/ST1/03718).

**Acknowledgements** All five authors would like to thank the Schloss Dagstuhl – Leibniz-Zentrum für Informatik 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.

## 1 Introduction

Integer programming is a problem of fundamental importance in combinatorial optimization with many theoretical and practical applications. It is known to be computationally very hard and is one of the 21 NP-complete problems in the original paper on NP-completeness by Karp [34]; the problem is known to be NP-complete even when the entries of the constraint matrix are zero and one only. On the positive side, Kannan and Lenstra [32, 41] 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 unimodular, i.e., all determinants of its submatrices are equal to 0 or $\pm 1$, in which case all vertices of the feasible region are integral and so linear programming algorithms can be applied.

Integer programming is known to be tractable for instances where the constraint matrix of an input integer program (IP) enjoys a certain block structure. The two most important cases are the cases of 2-stage IPs due to Hemmecke and Schultz [23], further investigated in particular in [1, 12, 27, 36, 37, 40], and $n$-fold IPs introduced by De Loera et al. [13] and further investigated in particular in [10, 11, 16, 22, 31, 40]. IPs of this kind appear in various contexts, see e.g. [29, 38, 39, 44]. These (theoretical) tractability results complement well a vast number of empirical results demonstrating tractability of instances with a block structure, e.g. [2–4, 18, 19, 35, 45–47].

A more general approach to tractability of IPs with sparse constraint matrices involves depths/widths of graphs defined on columns or rows of the constraint matrices. Ganian and Ordyniak [20] 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 and right hand sides $\|A, b\|_\infty$ can be solved efficiently. Levin, Onn and the second author [40] widely generalized this result by showing that IPs with bounded coefficients $\|A\|_\infty$ 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. The existence of efficient algorithms in the case 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 $A$, which is formed by minimal integer vectors of $\ker A$ in an orthant (see Section 2 for the rigorous definition). The maximum $\ell_1$-norm and $\ell_\infty$-norm of an element of the Graver basis of $A$ is denoted by $g_1(A)$ and $g_\infty(A)$, respectively. In particular, the following holds [40], where $\mathrm{ec}(A)$ denotes the entry complexity of a matrix $A$ defined as the maximum number of bits needed to represent any of the entries of $A$.

▶ **Theorem 1.** *There exist functions $f_P, f_D : \mathbb{N}^2 \to \mathbb{N}$ such that the following holds for every matrix $A$: $g_\infty(A) \leq f_P(\mathrm{td}_P(A), \mathrm{ec}(A))$ and $g_1(A) \leq f_D(\mathrm{td}_D(A), \mathrm{ec}(A))$.*

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 a sparse instance, for example, the matrix in the left, whose dual tree-depth is 5, can be transformed by row operations to the matrix with dual tree-depth 2 given in the right.

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

Such a transformation of an input matrix is a preconditioner as it changes an input matrix to an equivalent one that is computationally more tractable.

Preconditioning a matrix to make the problem computationally simpler, e.g., to make the feasible region not too flat in any direction, is a ubiquitous preprocessing step in mathematical programming solvers. In this paper, we are concerned with the existence and efficient computability of preconditioners to sparsity of matrices. Chan and Cooper together with the second, third and fourth authors [7,8] gave a structural characterization of matrices that are equivalent, i.e., can be transformed by row operations, to a matrix with small dual tree-depth, and used their structural results to design a fixed parameter algorithm to find such a matrix if it exists; we denote the smallest dual tree-depth of a matrix equivalent to $A$ by $\mathrm{td}^*_D(A)$.

▶ **Theorem 2.** *There exists an algorithm parameterized by $d$ and $e$ that for an input matrix $A$ with entry complexity $e$*

- *either outputs that $\mathrm{td}^*_D(A) > d$, or*
- *outputs a matrix $A'$ equivalent to $A$ such that the dual tree-depth of $A'$ is $\mathrm{td}^*_D(A)$ and its entry complexity is $\mathcal{O}(d^2 2^{2d} e)$.*

The structural characterization given in [7,8] exhibits an interesting link to matroid theory: an equivalent matrix with small dual tree-depth exists if and only if the column matroid of the matrix, which is invariant under row operations, has small contraction*-depth (see Theorem 3 below). We remark that the term branch-depth was used in [7,8] following the terminology from [33] but as there is a competing notion of branch-depth [14], we decided to use a different name for this notion throughout the paper to avoid confusion.

In this paper, we provide a structural characterization of matrices that are equivalent to a matrix with small primal tree-depth or small incidence tree-depth (Theorems 4 and 5). We also study corresponding preconditioners and design fixed parameter algorithms for constructing an equivalent matrix with small primal tree-depth and small entry complexity and for constructing an equivalent matrix with small dual tree-depth and small entry complexity, if such a matrix exists (Theorems 7 and 8). Finally, we employ our structural results to resolve an open problem whether integer programming is fixed-parameter tractable parameterized by the largest $\ell_1$-norm of the Graver basis element of a matrix $A$. Additionally, we show that the $\ell_1$-norm of each element of the Graver basis of a matrix $A$ is bounded by a function of the largest $\ell_1$-norm of a circuit of the matrix $A$ (Theorem 6). All these results are stated more precisely in Subsection 1.1.

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 $\ell_1$-norm of the Graver basis of the constraint matrix,
- $c_1(A)$, i.e., the $\ell_1$-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 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 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 [22], and, about a decade later, their applications has become ubiquitous, see e.g. [5, 6, 9, 10, 24, 28, 30, 39].

## 1.1 Our contribution

We now describe the results presented in this paper in detail; we refer the reader for the definitions of the notions not yet rigorously introduced to Section 2.

### 1.1.1 Characterization of depth parameters

The main structural result of [7, 8] is the following structural characterization of the existence of an equivalent matrix with small dual tree-depth in terms of the structural parameter of the column matroid, which is invariant under row operations.

▶ **Theorem 3.** *For every non-zero matrix $A$, it holds that the smallest dual tree-depth of a matrix 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 an equivalent matrix with small primal tree-depth and the existence of an equivalent matrix with small incidence tree-depth.

▶ **Theorem 4.** *For every matrix $A$, it holds that the smallest primal tree-depth of a matrix equivalent to $A$ is equal to the deletion-depth of $M(A)$, i.e.,* $\mathrm{td}^*_P(A) = \mathrm{dd}(A)$.

▶ **Theorem 5.** *For every matrix $A$, it holds that the smallest incidence tree-depth of a matrix 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

As mentioned earlier, Graver bases play an essential role in designing efficient algorithms for integer programming. A *circuit* of a matrix $A$ is a support-wise minimal integral vector contained in the kernel of $A$ such that all its entries are coprime. Hence, every circuit of a matrix $A$ is an element of the Graver basis of $A$ and so the maximum $\ell_1$-norm of an element of the Graver basis, which is denoted by $g_1(A)$, is an upper bound on the maximum $\ell_1$-norm of a circuit of $A$, which is denoted by $c_1(A)$.

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 $g_1(A)$, the $\ell_1$-norm of an element of the Graver basis of the constraint matrix $A$. An affirmative answer to this question follows from Theorems 6 and 8 below. We actually show that *the maximum $\ell_1$-norm $g_1(A)$ of an element of the Graver basis of a matrix $A$ is small if and only if $A$ is equivalent to a matrix with a small dual tree-depth and small entry complexity*; the implication from left to right is given

in Theorem 1 and the other implication in Theorem 12 (recall that $c_1(A) \leq g_1(A)$ for every matrix). We summarize the relation between the the maximum $\ell_1$-norm of a circuit of a matrix $A$, the maximum $\ell_1$-norm of an element of the Graver basis of $A$, and the existence of an equivalent matrix with small dual tree-depth and small entry complexity in the next theorem (the second part of the theorem is given in Corollary 13).

▶ **Theorem 6.** *There exist a function $f_1 : \mathbb{N} \to \mathbb{N}$ such that the following holds for every matrix $A$ with $\dim \ker A > 0$:*

- *the matrix $A$ is equivalent to a matrix $A'$ with $\mathrm{td}_D(A') \leq c_1(A)^2$ and $\mathrm{ec}(A') \leq 2\lceil c_1(A) \rceil$, and*
- *$c_1(A) \leq g_1(A) \leq f_1(c_1(A))$.*

Hence, informally speaking, the following statements are equivalent for every matrix $A$:

- The matrix $A$ is equivalent to a matrix with bounded dual tree-depth and bounded entry complexity.
- The contraction*-depth of the matroid $M(A)$ is bounded.
- The $\ell_1$-norm of every circuit of $A$ is bounded.
- The $\ell_1$-norm of every element of the Graver basis of $A$ is bounded.

### 1.1.3 Algorithms to compute matrices with small depth parameters

We design a parameterized algorithm for computing an equivalent matrix with small primal tree-depth and small entry complexity if one exists.

▶ **Theorem 7.** *There exists a function $f : \mathbb{N}^2 \to \mathbb{N}$ and an FPT algorithm for the parameterization by $d$ and $e$ that, for a given rational matrix $A$ with $m$ rows and $n$ columns, $m \leq n$:*

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

We also design a parameterized algorithm for computing an equivalent matrix with small dual tree-depth and small entry complexity if one exists. Note that the algorithm described in Theorem 2 for computing an equivalent matrix with small dual tree-depth is parameterized by the dual tree-depth of the to be constructed matrix and the entry complexity of the *input* matrix while the algorithm given below is parameterized by the entry complexity of the to be constructed matrix and so the algorithm can be applied to a wider set of input matrices.

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

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

We remark that if a matrix $A$ has entry complexity $e$ and is equivalent to a matrix with dual tree-depth $d$, then there exists an equivalent matrix with dual tree-depth $d$ and entry complexity bounded by a function of $d$ and $e$ (as implied by Theorem 2). 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 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 Hardness results

As our algorithmic results involve computing depth decompositions of matroids for various depth parameters in a parameterized way, we establish the computational hardness of these parameters in Theorem 18, 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 and if $B$ is a set of vectors, we write $\mathcal{L}(B)$ for the linear hull of the vectors contained in $B$. If $A$ is a linear space and $K$ 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. [21] 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 $w$.

## 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; the number of vertices and edges of $G$ is denoted by $|G|$ and $\|G\|$, 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). 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.

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. 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.

## 2.2 Matroids

We next review basic definitions from matroid theory; for detailed information, we refer to the book of Oxley [42]. 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*. 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. 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 $r(\{x\}) = r(\{x'\}) = r(\{x, x'\}) = 1$. 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 elements of $M$ are parallel if and only if they are assigned non-zero multiples of the same non-zero vector. Such an assignment of vectors of $\mathbb{F}^{r(M)}$ to the elements of $M$ is an $\mathbb{F}$-*representation* of $M$. Observe that the rank of a subset $X'$ of the ground set is the dimension of the linear hull of the vectors assigned to the elements of $X'$. 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. If $X' \subseteq X$, then the *restriction* of $M$ to $X'$, which is denoted by $M[X']$, is the matroid with the ground set $X'$ such that a subset of $X'$ is independent in $M[X']$ if and only if it is independent in $M$. In particular, the rank of $M[X']$ is $r_M(X')$. 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 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 it 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.

A matroid $M$ is *connected* if every two distinct elements of $M$ are 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 vector 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 $B$. 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. 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 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). 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 *equivalent* if one can be obtained from another by (equivalent) *row operations*, i.e., adding (a non-zero multiple of) one row to another and multiplying a row by a non-zero element. Observe that if $A$ and $A'$ are 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 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 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. 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 equivalent to $A$, $\mathrm{td}^*_D(A)$ is the smallest dual tree-depth of a matrix equivalent to $A$, and $\mathrm{td}^*_I(A)$ is the smallest incidence tree-depth of a matrix 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 $\ell_1$-norm of a circuit of $A$ and $c_\infty(A)$ for the maximum $\ell_\infty$-norm of a circuit of $A$. Note if $A$ and $A'$ are equivalent rational matrices, then $\mathcal{C}(A) = \mathcal{C}(A')$ and so the parameters $c_1(\cdot)$ and $c_\infty(\cdot)$ are invariant under row operations. Following the notation from [17], 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 $|\mathbf{x}_i| \leq |\mathbf{y}_i|$ 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 they 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 $\ell_1$-norm of a vector in $\mathcal{G}(A)$ and $g_\infty(A)$ is the maximum $\ell_\infty$-norm of a vector in $\mathcal{G}(A)$. Again, the parameters $g_1(\cdot)$ and $g_\infty(\cdot)$ are invariant under row operations as the Graver bases of 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 $c_1(A) \leq g_1(A)$ and $c_\infty(A) \leq g_\infty(A)$ for every matrix $A$.

## 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 in [14].

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, $\mathrm{dd}(M)$ is 1 plus the minimum deletion-depth of $M \setminus e$ where the minimum is taken over all elements $e$ of $M$.

The 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-decomposition tree*, defined as follows. If $M$ has a single element, then the deletion-decomposition tree of $M$ consists of a single vertex labeled with the single element of $M$. If $M$ is not connected, then the deletion-decomposition tree is obtained by identifying the roots of deletion-decomposition trees of the components of $M$. Otherwise, there exists an element $e$ such that $\mathrm{dd}(M) = \mathrm{dd}(M \setminus e) + 1$ and the deletion-decomposition tree of $M$ is obtained from the deletion-decomposition tree of $M \setminus e$ by adding a new vertex adjacent to the root of the deletion-decomposition 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 $e$. Observe that the height of the deletion-decomposition tree is equal to the deletion-depth of $M$. In what follows, we consider deletion-decomposition 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-decomposition 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 $\mathrm{cd}(M) = 1$. If $M$ is not connected, then $\mathrm{cd}(M)$ is the maximum contraction-depth of a component of $M$. Otherwise, $\mathrm{cd}(M)$ 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-decomposition tree* analogously to a deletion-decomposition tree; the contraction-depth of a matroid $M$ is the smallest height of a contraction-decomposition tree of $M$.

We next introduce the contraction-deletion-depth, which was studied under the name of type in [15], but we decided to use the name from [14], which we find to describe the parameter in the context considered here better. The *contraction-deletion-depth* of 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, $\mathrm{cdd}(M)$ 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 $M$. Clearly, 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 [33] 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 [14]. A *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 $r(X)$. There is an alternative equivalent definition of the parameter for represented matroids. The *contraction*-depth* of a represented matroid $M$ can be defined recursively as follows. If $M$ has rank zero, then $\mathrm{c}^*\mathrm{d}(M) = 0$. If $M$ is not connected, then $\mathrm{c}^*\mathrm{d}(M)$ is the maximum contraction*-depth of a component of $M$. Otherwise, $\mathrm{c}^*\mathrm{d}(M)$ 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 $\mathrm{c}^*\mathrm{d}(M) \leq \mathrm{cd}(M)$.

Kardoš et al. [33] established the connection between the contraction*-depth and the existence of a long circuit, which is described in Theorem 9.

▶ **Theorem 9.** *Let $M$ be a matroid and $k$ the size of its largest circuit. It holds that $\log_2 k \leq \mathrm{c}^*\mathrm{d}(M) \leq k^2$. Moreover, there exists a polynomial-time algorithm that for an input oracle-given matroid $M$ outputs a contraction*-decomposition tree of depth at most $k^2$.*

All contractions used in the proof of the inequality $\mathrm{c}^*\mathrm{d}(M) \leq k^2$ are contractions of elements of a matroid $M$, i.e., the one-dimensional subspaces as in the definition of the contraction*-depth are all generated by elements of $M$. We remark that this implies that $\mathrm{cd}(M) \leq k^2 + 1$. The sequence of such contractions can be visualized by a *contraction*-decomposition tree* that is defined in the same way as a contraction-decomposition tree except that one-vertex trees are associated with matroids of rank zero (rather than matroids consisting of a single element), however, the edges of the tree are still labeled by some of the elements of $M$. Note that the minimum depth of a contraction*-decomposition 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 contraction*-decomposition tree of $M$ as the definition of the contraction*-depth in the case of represented matroids permits contractions by arbitrary one-dimensional subspaces.

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 $\mathrm{c}^*\mathrm{dd}(M)$, is defined recursively as follows. If $M$ has rank zero, then $\mathrm{c}^*\mathrm{dd}(M) = 0$; if $M$ has a single non-loop element, then $\mathrm{c}^*\mathrm{dd}(M) = 1$. If $M$ is not connected, then $\mathrm{c}^*\mathrm{dd}(M)$ is the maximum contraction*-deletion-depth of a component of $M$. Otherwise, $\mathrm{c}^*\mathrm{dd}(M)$ 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 $\mathrm{c}^*\mathrm{dd}(M) \leq \mathrm{cdd}(M)$ and $\mathrm{c}^*\mathrm{dd}(M) \leq \mathrm{c}^*\mathrm{d}(M)$ 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 row operations as 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-decomposition tree of the matroid associated with a given matrix to construct a matrix with small primal tree-depth that is equivalent to the given matrix.

▶ **Lemma 10.** *There exists a polynomial-time algorithm that for an input matrix $A$ and a deletion-decomposition tree of $M(A)$ with height $d$ outputs a matrix $A'$ equivalent to $A$ such that $\mathrm{td}_P(A') \leq d$.*

**Proof.** We establish the existence of the algorithm by proving that $\mathrm{td}_P(A') \leq d$ in a constructive (algorithmic) way. Due to space constraints, we omit some details in the arguments that follow.

Fix a matrix $A$ and a deletion-decomposition tree $T$ of $M(A)$ with height $d$. Let $X$ be the set of non-zero columns that are labels of the vertices of $T$. It can be shown that the columns contained in $X$ form a basis of the column space of the matrix $A$. In particular, unless $A$ is the zero matrix, the set $X$ is non-empty. Let $A'$ be the matrix obtained from $A$ by row operations such that the submatrix of $A'$ induced by the columns of $X$ is the unit matrix with possibly some additional zero rows. We will prove by induction on the number of columns of an input matrix $A$ that the primal tree-depth of $A'$ is at most $d$.

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 $M(A)$, and every label of an edge $e$ is a linear combination of labels of the vertices in the subtree delimited by $e$. It follows that if $x$ is a label of the root of $T$, then $x$ 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-decomposition tree of $M(B)$ obtained from $T$ by removing the label $x$ from the root. Since the matrix $B'$ produced by the algorithm described above for $B$ and $T'$ is the submatrix of $A'$ formed by the columns different from $x$ (possibly after permuting rows) and the vertex associated with the column $x$ is isolated in the primal graph of $A'$, 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 now 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 $T$, and let $B'$ be the matrix obtained from $A'$ by deleting the column $x$. By induction, the primal tree-depth of $B'$ is at most $d-1$, which implies that 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 $Y_1, \ldots, Y_k$ be the labels of the vertices and edges of the $k$ subtrees of $T$ delimited by the $k$ edges incident with the root of $T$, and let $B_1, \ldots, B_k$ be the submatrices of $A$ formed by the columns contained in $Y_1, \ldots, Y_k$. 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. ◀

We are now ready to establish the link between the optimal primal tree-depth of a matrix and the deletion-depth of the matroid associated with the matrix. Due to space constraints, the proof of Theorem 4 is sketched only.

**Sketch of the proof of Theorem 4.** Fix a matrix $A$. By Lemma 10, it holds that $\mathrm{td}^*_P(A) \leq \mathrm{dd}(A)$. So, we focus on proving that $\mathrm{dd}(A) \leq \mathrm{td}^*_P(A)$. We will show that every matrix $B$ satisfies that $\mathrm{dd}(B) \leq \mathrm{td}_P(B)$, which 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. If the matroid $M(B)$ is not connected, we apply induction to each of its components and derive that each component of $M(B)$ has deletion depth at most $\mathrm{td}_P(B)$. This implies that $\mathrm{dd}(B) \leq \mathrm{td}_P(B)$.

We next assume that the matroid $M(B)$ is connected. It can be shown that the primal graph of $B$ is also connected, which yields that 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 10 and Theorem 4 to get a parameterized algorithm for computing an optimal primal tree-depth of a matrix over a finite field. Due to space constraints, several steps of the proof of Corollary 11 are sketched only.

▶ **Corollary 11.** *There exists an FPT 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 $A'$ equivalent to $A$ with $\mathrm{td}_P(A') \leq d$ and also outputs the associated deletion-decomposition tree of $M(A)$ with height $\mathrm{td}_P(A')$.*

**Proof.** The property that a matroid $M$ has deletion depth at most $d$ can be expressed in monadic second order logic. Specifically, there exists a monadic second order formula $\psi_d(X)$ that describes whether the deletion-depth of a restriction of the matroid $M$ to a subset $X$ of the elements of $M$ is at most $d$. In the formula that we present, small letters are used to denote elements of a matroid and capital letters subsets of the elements. Assuming that $\psi_c(x, y)$ is a monadic second order formula describing the existence of a circuit containing two elements $x$ and $y$ and $\psi_C(X)$ is a monadic second order formula describing whether a subset $X$ is a component of a matroid, The sought formula $\psi_d(\cdot)$ is defined inductively (note that $\psi_d(\emptyset)$ is true for all $d$):

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

Hliněný [25, 26] 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, this establishes the existence of a fixed parameter algorithm deciding whether $\mathrm{td}^*_P(A) = \mathrm{dd}(M(A)) \leq d$ (the equality follows from Theorem 4).

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-decomposition tree of $M$ with height at most $d$. The deletion-depth of an input matroid $M$ is one if and only if every element of $M$ is a loop or a bridge. Hence, if the deletion-depth of $M$ is $d = 1$, then the deletion-depth decomposition tree of height one consists of a single vertex labeled

with all elements of $M$. If the deletion-depth of $M$ is at most $d \geq 2$ and $M$ 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-depth decomposition trees obtained recursively. Finally, if the deletion-depth of $M$ is at most $d \geq 2$ and $M$ is connected, we loop over all elements $e$ of $M$ and test whether $\mathrm{dd}(M \setminus e) \leq d-1$. Such an element $e$ must exist and when it is found, we recursively apply the algorithm to $M \setminus e$ to obtain a deletion-depth decomposition tree $T$ of $M \setminus e$ with height $d-1$, which is then extended to a deletion-depth decomposition tree of $M$ of height $d$. ◀

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. Due to space constraints, the proof of Theorem 5 is sketched only.

**Proof of Theorem 5.** We prove the equality as two inequalities starting with the inequality $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{td}^*_I(A) - 1$. To prove this inequality, we show that $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{td}_I(A) - 1$ 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 $m = 1$. We focus on describing the induction step, which considers the case when $A$ 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 component 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 $\mathrm{c}^*\mathrm{dd}(A)$ is the maximum of $\mathrm{c}^*\mathrm{dd}(A_i)$. The induction hypothesis yields that $\mathrm{c}^*\mathrm{dd}(A_i) \leq \mathrm{td}_I(A_i) - 1$. 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 $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{td}_I(A) - 1$.

The core of the inductive argument showing that $\mathrm{c}^*\mathrm{dd}(A) \leq \mathrm{td}_I(A) - 1$ is formed by the case when 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 $A$. The core of the argument is the inductive step when the matroid $M(A)$ is connected, which we present next. The definition of the contraction*-deletion-depth implies that there exists an element $x$ of $M(A)$ such that $\mathrm{c}^*\mathrm{dd}(M(A)\setminus x) = \mathrm{c}^*\mathrm{dd}(M(A)) - 1 = \mathrm{c}^*\mathrm{dd}(A) - 1$ or there exists a one-dimensional subspace such that the contraction by this subspace yields a matroid $M'$ such that $\mathrm{c}^*\mathrm{dd}(M') = \mathrm{c}^*\mathrm{dd}(M(A)) - 1 = \mathrm{c}^*\mathrm{dd}(A) - 1$. 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 $A''$ equivalent to $A'$ such that $\mathrm{td}_I(A'') \leq \mathrm{c}^*\mathrm{dd}(A') + 1 = \mathrm{c}^*\mathrm{dd}(A)$, and let $A'''$ be the matrix obtained from $A$ by the same row operations using 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, $\mathrm{td}_I(A''') \leq \mathrm{td}_I(A'') + 1$. Since $A'''$ is equivalent to $A$, it follows that

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

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 $M'$ such that $\mathrm{c}^*\mathrm{dd}(M') = \mathrm{c}^*\mathrm{dd}(A) - 1$. Let $A'$ be the matrix obtained from $A$ by row operations such that the contracted subspace used to obtain $M'$ is generated by the unit vector with the non-zero entry being the first entry, and let $B$ be the matrix obtained from $A'$ by deleting the first row. By the induction hypothesis, there exists a matrix $B'$ equivalent to $B$ such that $\mathrm{td}_I(B') \leq \mathrm{c}^*\mathrm{dd}(A') + 1 = \mathrm{c}^*\mathrm{dd}(A)$, and let $A''$ be the matrix consisting of the first row of $A$ and the matrix $B'$. Observe that $A''$ is 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 $\mathrm{td}_I(A'') \leq \mathrm{td}_I(B') + 1$. Hence, it follows that

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

The proof of the theorem is now completed. ◀

## 4 Primal tree-depth

In this section, we present a parameterized algorithm for computing an equivalent matrix with small primal tree-depth and bounded entry complexity if such a matrix exists. Due to space constraints, details of some of the arguments used in the proof are omitted.

**Proof of Theorem 7.** Let $f_P(\cdot,\cdot)$ be the function from the statement of Theorem 1 and set $\kappa_0$ to be the least common multiple of the integers $1, \ldots, f_P(d,e)$. Observe that the entry of every circuit of a matrix $B$ with $\mathrm{td}_P(B) \leq d$ and $\mathrm{ec}(B) \leq e$ divides $\kappa_0$ as $c_\infty(B) \leq f_P(d,e)$.

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 input matrix $A$ by selecting an arbitrary basis of the column space and performing row-operations 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 $\kappa_0$, the algorithm arrives at the first conclusion of the theorem. The algorithm next multiplies each row of $A_I$ by $\kappa_0$, This yields an integral matrix $A_0$ with entries between $-\kappa_0^2$ and $\kappa_0^2$.

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 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 $e$. 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.

We argue that *if $A$ is 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.* As this is the key step in the proof, we present it in full detail. 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 $\kappa_0$ by the definition of $\kappa_0$); 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 a square submatrix of $A_I$ formed by the columns $X$ and some of the rows such that $B_I$ is non-singular, and let $B_0$ be the square submatrix of $A_0$ formed by the same rows and columns. By [17, Lemma 3.3], the matrix $B_I^{-1}$ is $1/\dot{\kappa}_A$-integral and the absolute values of its entries are between $1/c_\infty(A)$ and $c_\infty(A)$. Note that both $c_\infty(A)$ and $\dot{\kappa}_A$ divide $\kappa_0$ (here, we use the definition of $\kappa_0$ and the assumption that $A$ is equivalent to a matrix with primal tree-depth at most $d$ and entry complexity at most $e$) and so this the matrix $B_I^{-1}$ is $1/\kappa_0$-integral and the absolute values of its entries are between $1/\kappa_0$ and $\kappa_0$. Let $B'$ be the matrix obtained from $B_I^{-1}$ by

multiplying each entry by $\kappa_0$; note that $B'$ is an integer matrix with entries between $-\kappa_0^2$ and $\kappa_0^2$. The definitions of the matrices $B_I$, $B_0$ and $B'$ yield that $B'B_0$ is a diagonal matrix with all diagonal entries equal to $\kappa_0^2$. It follows that the columns $X$ form a set independent in $M_p$.

We now continue the description of the algorithm. As the next step, we apply the algorithm from Corollary 11 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. If it determines that the deletion-depth of $A_0$ is at most $d$, it also outputs a deletion-decomposition tree of $M_p$ with height at most $d$. If the output deletion-decomposition tree is not valid for the matroid $M_{\mathbb{Q}}$, we again arrive at the first conclusion of the theorem. If the output deletion-decomposition tree is also a deletion-decomposition tree of the matroid $M_{\mathbb{Q}}$, we use the algorithm from Lemma 10 to obtain a matrix $A'$ equivalent to $A$ such that the primal tree-depth of $A'$ at most the height of the deletion-decomposition 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 obtained by dividing two entries of a circuit of $A$. If the absolute value of the numerator or the denominator of any these fractions exceeds $\kappa_0$, 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 $2\lceil\log_2(\kappa_0+1)\rceil$ and $\kappa_0$ depends on $d$ and $e$ only. ◀

## 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. Due to space constraints, details of some of the arguments used in the proof are omitted.

▶ **Theorem 12.** *There exists a polynomial-time algorithm that for a given matrix $A$ whose rank is smaller than the number of its columns outputs an equivalent matrix $A'$ such that $\mathrm{td}_D(A') \leq c_1(A)^2$ and $\mathrm{ec}(A') \leq 2\lceil\log_2 c_1(A)\rceil$.*

**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 9 to the matroid $M(A)$ given by the columns of the matrix $A$. Let $T$ be the contraction*-decomposition tree output by the algorithm, let $d$ be its depth 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. Since the columns contained in $X$ form a base of the column space of $A$, we can perform row-operations on the matrix $A$ in a way that the submatrix formed by the columns of $X$ is an identity matrix; let $A'$ be the resulting matrix. The algorithm outputs the matrix $A'$.

We now analyze the matrix $A'$ that is output by the algorithm. Since the rank of $A$ is smaller than the number of its columns, the matrix $A$ has at least one circuit and so $c_1(A) \geq 2$. Observe that every circuit of $M(A)$ contains at most $c_1(A)$ elements and so it holds that $d \leq c_1(A)^2$, i.e., the depth of $T$ is at most $c_1(A)^2$. Let $F$ be a rooted forest obtained from $T$ by removing the root and for each edge $e$, associating the vertex of $e$ farther from the root of $T$ with the (unique) row of $A'$ that is non-zero in the column that is the label of $e$. Hence, the vertex set of $F$ is formed by the rows of $A'$. Since it can be shown that the dual graph of $A'$ is a subgraph of $\mathrm{cl}(F)$, it follows that that $\mathrm{td}_D(A') \leq c_1(A)^2$ (note that the height of $F$ is at most $c_1(A)^2$).

It remains to analyze the entry complexity of $A'$. The entries of $A'$ in the columns of $X$ are zero or one. Every column $z$ of $A'$ that is not contained in $X$ forms a circuit with some of the columns of $X$. The entries of $A'$ in the column $z$ are equal to $-c_x/c_z$ where $c_x$, $x \in X$, and $c_z$ are the corresponding integer entries of this circuit of $A'$ (and so of $A$). We conclude that the entry complexity of $A'$ is at most $2\lceil\log_2 c_1(A)\rceil$. ◀

Theorem 12 implies that $g_1(A) \leq f_D\left(c_1(A)^2, 2\lceil \log_2 c_1(A)\rceil\right)$ for every matrix $A$ with $\mathcal{C}(A) \neq \emptyset$, where $f_D$ is the function from Theorem 1. Note that the condition $\mathcal{C}(A) \neq \emptyset$ is necessary as otherwise $A$ has no circuit and so $c_1(A)$ is not defined. We state this bound as a corollary.

▶ **Corollary 13.** *There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that $g_1(A) \leq f(c_1(A))$ for every matrix $A$ with $\mathcal{C}(A) \neq \emptyset$.*

We next combine the algorithms from Theorems 2 and 12.

▶ **Corollary 14.** *There exists a function $f : \mathbb{N} \to \mathbb{N}$ and an FPT 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 equivalent to $A$, its dual tree-depth is $\mathrm{td}^*_D(A)$ and its entry complexity is at most $f(k)$.*

**Proof of Corollary14.** For an input matrix $A$, we apply the algorithm from Theorem 12 to get a matrix $A'$ equivalent to $A$. If the dual tree-depth of $A'$ is larger than $k^2$ or the entry complexity of $A'$ is larger than $2\lceil \log_2 k\rceil$, then $c_1(A) > k$ and we arrive at the first conclusion. Otherwise, we apply the algorithm from Theorem 2 with parameters $d = k^2$ and $e = 2\lceil \log_2 k\rceil$ to compute a matrix $A''$ 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 $O(k^4 2^{2k^2} \log k)$. ◀

Finally, the previous corollary together with Theorem 1 yields the parameterized algorithm for testing whether an input matrix is 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 1 and set $k = f_D(d, e)$. Apply the algorithm from Corollary 14 with the parameter $k$ to an input matrix $A$. If the algorithm reports that $c_1(A) > k$, then $A$ is not 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 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 $2^{O(k^2)} = 2^{O\left(f_D(d,e)^2\right)}$, i.e., it 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 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 $|G| + \|G\|$ elements, which correspond to the vertices and the edges of $G$. We next 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 $e_w - e_{w'}$.

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 a maximal subset of edges $ww' \in E(G)$ such that

- $A$ contains a vector $e_w + \alpha e_{w'}$ for a non-zero element $\alpha \in \mathbb{F}$, and
- the set containing all vectors $e_w$, $w \in W$, and all vectors $e_w - e_{w'}$, $ww' \in F$, is linearly independent.

The graph $G/A$ is obtained by deleting all vertices of $W$ and then contracting all edges contained in $F$. The next lemma asserts that $G/A$ is well-defined. We remark that the proof of the next lemma as well as the proofs of Lemmas 16 and 17 are omitted due to space constraints.

▶ **Lemma 15.** *Let $G$ be a graph, $\mathbb{F}$ a field and $A$ a linear subspace of $\mathbb{F}^{V(G)}$. The graph $G/A$ is well-defined, i.e., the graph $G/A$ does not depend on the choice of the set $F$ in its definition.*

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 16.** *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 $G/A$.*

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 [43] to establish that computing tree-depth of a graph is NP-complete.

▶ **Lemma 17.** *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 equivalent.*

- *There exists 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 $|X| + |Y| - k + 1$.*

We are now ready to state and prove our hardness result.

▶ **Theorem 18.** *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 [43]. 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 $(|G'| + 1)M_{\mathbb{F}}(G')$ has contraction-deletion-depth at most $|X| + |Y| - k + 1$.
- The matroid $(|G'| + 1)M_{\mathbb{F}}(G')$ has contraction*-deletion-depth at most $|X| + |Y| - k$.

The equivalence of the first and second statements follow directly from Lemma 17. Since the rank of the matroid $(|G'|+1)M_{\mathbb{F}}(G')$ 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 $(|G'|+1)M_{\mathbb{F}}(G')$ is parallel to (at least) $|G'|$ 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 17 now implies the equivalence of the third and fourth statements. As the matroids $2M_{\mathbb{F}}(G')$, $M_{\mathbb{F}}(G')$ and $(|G'|+1)M_{\mathbb{F}}(G')$ can be easily constructed from the input graph $G$ in time polynomial in $|G|$, 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 $M$ [42, 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. ◀

## References


1 Matthias Aschenbrenner and Raymond Hemmecke. Finiteness theorems in stochastic integer programming. *Foundations of Computational Mathematics*, 7(2):183–227, 2007.

2 Cevdet Aykanat, Ali Pinar, and Ümit V. Çatalyürek. Permuting sparse rectangular matrices into block-diagonal form. *SIAM Journal on Scientific Computing*, 25(6):1860–1879, 2004.

3 Martin Bergner, Alberto Caprara, Alberto Ceselli, Fabio Furini, Marco E Lübbecke, Enrico Malaguti, and Emiliano Traversi. Automatic Dantzig–Wolfe reformulation of mixed integer programs. *Mathematical Programming*, 149(1-2):391–424, 2015.

4 Ralf Borndörfer, Carlos E Ferreira, and Alexander Martin. Decomposing matrices into blocks. *SIAM Journal on Optimization*, 9(1):236–269, 1998.

5 Robert Bredereck, Aleksander Figiel, Andrzej Kaczmarczyk, Dušan Knop, and Rolf Niedermeier. High-multiplicity fair allocation made more practical. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 260–268. International Foundation for Autonomous Agents and Multiagent Systems, 2021.

6 Robert Bredereck, Andrzej Kaczmarczyk, Dušan Knop, and Rolf Niedermeier. High-multiplicity fair allocation: Lenstra empowered by n-fold integer programming. In *Proceedings of the 20th ACM Conference on Economics and Computation*, pages 505–523. Association for Computing Machinery, 2019.

7 Timothy F. N. Chan, Jacob W. Cooper, Martin Koutecký, Daniel Král', and Kristýna Pekárková. Matrices of optimal tree-depth and row-invariant parameterized algorithm for integer programming. To appear in SIAM Journal on Computing.

8 Timothy F. N. Chan, Jacob W. Cooper, Martin Koutecký, Daniel Král', and Kristýna Pekárková. Matrices of optimal tree-depth and row-invariant parameterized algorithm for integer programming. In *47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:19, 2020.

9 Hua Chen, Lin Chen, and Guochuan Zhang. Fpt algorithms for a special block-structured integer program with applications in scheduling. *preprint arXiv:2107.01373*, 2021.

10 Lin Chen and Dániel Marx. Covering a tree with rooted subtrees–parameterized and approximation algorithms. In *29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2801–2820. SIAM, 2018.

11 Jana Cslovjecsek, Friedrich Eisenbrand, Christoph Hunkenschröder, Lars Rohwedder, and Robert Weismantel. Block-structured integer and linear programming in strongly polynomial and near linear time. In *32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1666–1681. SIAM, 2021.

12 Jana Cslovjecsek, Friedrich Eisenbrand, Michal Pilipczuk, Moritz Venzin, and Robert Weismantel. Efficient sequential and parallel algorithms for multistage stochastic integer programming using proximity. In *29th Annual European Symposium on Algorithms (ESA)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:14, 2021.

13 Jesús A. De Loera, Raymond Hemmecke, Shmuel Onn, and Robert Weismantel. $N$-fold integer programming. *Discrete Optimization*, 5(2):231–241, 2008.

14 Matt DeVos, O-joung Kwon, and Sang-il Oum. Branch-depth: Generalizing tree-depth of graphs. *European Journal of Combinatorics*, 90:103186, 2020.

15 G. Ding, B. Oporowski, and J. Oxley. On infinite antichains of matroids. *Journal of Combinatorial Theory Series B*, 63(1):21–40, 1995.

16 Friedrich Eisenbrand, Christoph Hunkenschröder, and Kim-Manuel Klein. Faster Algorithms for Integer Programs with Block Structure. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:13, 2018.

17 Farbod Ekbatani, Bento Natura, and László A. Végh. Circuit imbalance measures and linear programming. *preprint arXiv:2108.03616*, 2021.

18 Michael C. Ferris and Jeffrey D. Horn. Partitioning mathematical programs for parallel solution. *Mathematical Programming*, 80(1):35–61, 1998.

19 Gerald Gamrath and Marco E. Lübbecke. Experiments with a generic Dantzig–Wolfe decomposition for integer programs. *Experimental Algorithms*, 6049:239–252, 2010.

20 Robert Ganian and Sebastian Ordyniak. The complexity landscape of decompositional parameters for ILP. *Artificial Intelligence*, 257:61–71, 2018.

21 P.R. Halmos. *Finite-Dimensional Vector Spaces*. Undergraduate Texts in Mathematics. Springer, 1993.

22 Raymond Hemmecke, Shmuel Onn, and Lyubov Romanchuk. $N$-fold integer programming in cubic time. *Mathematical Programming*, 137:325–341, 2013.

23 Raymond Hemmecke and Rüdiger Schultz. Decomposition of test sets in stochastic integer programming. *Mathematical Programming*, 94:323–341, 2003.

24 Danny Hermelin, Hendrik Molter, Rolf Niedermeier, and Dvir Shabtay. Equitable scheduling for the total completion time objective. *preprint arXiv:2112.13824*, 2021.

25 Petr Hliněný. Branch-width, parse trees, and monadic second-order logic for matroids. In *20th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of *LNCS*, pages 319–330, 2003.

26 Petr Hliněný. Branch-width, parse trees, and monadic second-order logic for matroids. *Journal of Combinatorial Theory Series B*, 96(3):325–351, 2006.

27 Klaus Jansen, Kim-Manuel Klein, and Alexandra Lassota. The double exponential runtime is tight for 2-stage stochastic ILPs. In *Integer Programming and Combinatorial Optimization - 22nd International Conference (IPCO)*, volume 12707 of *Lecture Notes in Computer Science*, pages 297–310. Springer, 2021.

28 Klaus Jansen, Kim-Manuel Klein, Marten Maack, and Malin Rau. Empowering the configuration-ip: new ptas results for scheduling with setup times. To appear in Mathematical Programming.

29 Klaus Jansen, Kim-Manuel Klein, Marten Maack, and Malin Rau. Empowering the configuration-IP – new PTAS results for scheduling with setups times. *preprint arXiv:1801.06460*, 2018.

30 Klaus Jansen, Alexandra Lassota, and Marten Maack. Approximation algorithms for scheduling with class constraints. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 349–357. Association for Computing Machinery, 2020.

31 Klaus Jansen, Alexandra Lassota, and Lars Rohwedder. Near-linear time algorithm for $n$-fold ILPs via color coding. *SIAM Journal on Discrete Mathematics*, 34(4):2282–2299, 2020.

32 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, August 1987.

33 František Kardoš, Daniel Král', Anita Liebenau, and Lukáš Mach. First order convergence of matroids. *European Journal of Combinatorics*, 59:150–168, 2017.

34 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations, The IBM Research Symposia Series*, pages 85–103. Springer, 1972.

35 Taghi Khaniyev, Samir Elhedhli, and Fatih Safa Erenay. Structure detection in mixed-integer programs. *INFORMS Journal on Computing*, 30(3):570–587, 2018.

36 Kim-Manuel Klein. About the complexity of two-stage stochastic IPs. To appear in Mathematical Programming.

37 Kim-Manuel Klein and Janina Reuter. Collapsing the tower - on the complexity of multistage stochastic ips. In *33rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 348–358. SIAM, 2022.

38 Dušan Knop and Martin Koutecký. Scheduling kernels via configuration LP. *preprint arXiv:2003.02187*, 2018.

39 Dušan Knop and Martin Koutecký. Scheduling meets $n$-fold integer programming. *Journal of Scheduling*, 21(5):493–503, 2018.

40 Martin Koutecký, Asaf Levin, and Shmuel Onn. A parameterized strongly polynomial algorithm for block structured integer programs. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107, pages 85:1–85:14, 2018.

41 Hendrik W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

42 James Oxley. *Matroid Theory*. Oxford Graduate Texts in Mathematics. Oxford University Press, 2011.

43 Alex Pothen. The complexity of optimal elimination trees. *Technical Report CS-88-13, Pennsylvania State University*, 1988.

44 Rüdiger Schultz, Leen Stougie, and Maarten H. van der Vlerk. Solving stochastic programs with integer recourse by enumeration: A framework using gröbner basis reductions. *Mathematical Programming*, 83:229–252, 1998.

45 François Vanderbeck and Laurence A Wolsey. Reformulation and decomposition of integer programs. In *50 Years of Integer Programming 1958-2008*, pages 431–502. Springer, 2010.

46 Jiadong Wang and Ted Ralphs. Computational experience with hypergraph-based methods for automatic decomposition in discrete optimization. In *Proceedings of the 10th International Conference on Integration of Constraint Programming*, pages 394–402. Springer, 2013.

47 Roman L. Weil and Paul C. Kettler. Rearranging matrices to block-angular form for decomposition (and other) algorithms. *Management Science*, 18(1):98–108, 1971.