Mathematical Programming (2023) 198:561–593
https://doi.org/10.1007/s10107-022-01783-x

FULL LENGTH PAPER
Series A

# On the optimality of pseudo-polynomial algorithms for integer programming

**Fedor V. Fomin**[1] · **Fahad Panolan**[2] · **M. S. Ramanujan**[3] · **Saket Saurabh**[4]

Received: 2 January 2020 / Accepted: 25 January 2022 / Published online: 14 February 2022

## Abstract

In the classic *Integer Programming Feasibility* (IPF) problem, the objective is to decide whether, for a given $m \times n$ matrix $A$ and an $m$-vector $b = (b_1, \ldots, b_m)$, there is a non-negative integer $n$-vector $x$ such that $Ax = b$. Solving (IPF) is an important step in numerous algorithms and it is important to obtain an understanding of the precise complexity of this problem as a function of natural parameters of the input. The classic pseudo-polynomial time algorithm of Papadimitriou [J. ACM 1981] for instances of (IPF) with a constant number of constraints was only recently improved upon by Eisenbrand and Weismantel [SODA 2018] and Jansen and Rohwedder [ITCS 2019]. Jansen and Rohwedder designed an algorithm for (IPF) with running time $\mathcal{O}(m\Delta)^m \log(\Delta) \log(\Delta + \|b\|_\infty) + \mathcal{O}(mn)$. Here, $\Delta$ is an upper bound on the absolute values of the entries of $A$. We continue this line of work and show that under the Exponential Time Hypothesis (ETH), the algorithm of Jansen and Rohwedder is nearly optimal, by proving a lower bound of $n^{o(\frac{m}{\log m})} \cdot \|b\|_\infty^{o(m)}$. We also prove that assuming ETH, (IPF) cannot be solved in time $f(m) \cdot (n \cdot \|b\|_\infty)^{o(\frac{m}{\log m})}$ for any computable function

A preliminary version of the paper appeared in the proceedings of 26th Annual European Symposium on Algorithms (ESA) 2018. This work is supported by the European Research Council (ERC) via grant LOPPRE, reference 819416, and the Norwegian Research Council via project MULTIVAL.

✉ Fahad Panolan
fahad@cse.iith.ac.in

Fedor V. Fomin
fomin@ii.uib.no

M. S. Ramanujan
R.Maadapuzhi-Sridharan@warwick.ac.uk

Saket Saurabh
saket@imsc.res.in

1 Department of Informatics, University of Bergen, Bergen, Norway

2 Department of Computer Science and Engineering, IIT Hyderabad, Sangareddy, India

3 University of Warwick, Coventry, United Kingdom

4 The Institute of Mathematical Sciences, HBNI, Chennai, India

$f$. This motivates us to pick up the line of research initiated by Cunningham and Geelen [IPCO 2007] who studied the complexity of solving (IPF) with non-negative matrices in which the number of constraints may be unbounded, but the branch-width of the column-matroid corresponding to the constraint matrix is a constant. We prove a lower bound on the complexity of solving (IPF) for such instances and obtain optimal results with respect to a closely related parameter, path-width. Specifically, we prove *matching* upper and lower bounds for (IPF) when the *path-width* of the corresponding column-matroid is a constant .

**Keywords** Integer programming · Algorithms and data structures

**Mathematics Subject Classification** 68W01 General topics in the theory of algorithms · 68Q25 Analysis of algorithms and problem complexity

## 1 Introduction

In the classic *Integer Programming* problem, the input is an $m \times n$ integer matrix $A$, and an $m$-vector $b = (b_1, \dots, b_m)$. We consider the feasibility version of the problem, where the objective is to find a non-negative integer $n$-vector $x$ (if one exists) such that $Ax = b$. Solving this problem, denoted by (IPF), is a fundamental step in numerous algorithms and it is important to obtain an understanding of the precise complexity of this problem as a function of natural parameters of the input .

(IPF) is known to be NP-hard [1]. However, there are two classic algorithms due to Lenstra [16] and Papadimitriou [20] solving (IPF) in polynomial or pseudo-polynomial time for two important cases when the number of variables and the number of constraints are bounded. These algorithms in some sense complement each other.

The algorithm of Lenstra shows that (IPF) is solvable in polynomial time when the number of variables is bounded. Actually, the result of Lenstra is even stronger: (IPF) is *fixed-parameter tractable* parameterized by the number of variables. However, the running time of Lenstra's algorithm is doubly exponential in $n$. Later, Kannan [14] provided an algorithm for (IPF) running in time $n^{\mathcal{O}(n)}$. Deciding whether the running time $n^{\mathcal{O}(n)}$ can be improved to $2^{\mathcal{O}(n)}$ is a long-standing open question.

Our work is motivated by the complexity analysis of the complementary case when the number of constraints is bounded. (IPF) is NP-hard already for $m = 1$ (the Knapsack problem) but solvable in pseudo-polynomial time. In 1981, Papadimitriou [20] extended this result by showing that (IPF) is solvable in pseudo-polynomial time on instances for which the number of constraints $m$ is a constant. The algorithm of Papadimitriou consists of two steps. The first step is combinatorial, showing that if the entries of $A$ and $b$ are from $\{0, \pm 1, \dots, \pm d\}$, and (IPF) has a solution, then there is also a solution which is in $\{0, 1, \dots, n(md)^{2m+1}\}^n$. The second, algorithmic step shows that if (IPF) has a solution with the maximum entry at most $B$, then the problem is solvable in time $\mathcal{O}((nB)^{m+1})$. Thus the total running time of Papadimitriou's algorithm is $\mathcal{O}(n^{2m+2} \cdot (md)^{(m+1)(2m+1)})$, where $d$ is an upper bound on the absolute values of the entries of $A$ and $b$. There was no algorithmic progress on this problem until the very recent breakthrough of Eisenbrand and Weismantel [6]. They proved the following result.

**Proposition 1** (Theorem 2.2,Eisenbrand and Weismantel [6]) *(IPF) with $m \times n$ matrix $A$ is solvable in time $(m \cdot \Delta)^{\mathcal{O}(m)} \cdot \|b\|_\infty^2$, where $\Delta$ is an upper bound on the absolute values of the entries of $A$.*

Then, Jansen and Rohwedder improved Proposition 1 and gave a matching lower bound very recently [12].

**Proposition 2** (Jansen and Rohwedder [12]) *(IPF) with $m \times n$ matrix $A$ is solvable in time $\mathcal{O}(m\Delta)^m \log(\Delta)\log(\Delta + \|b\|_\infty) + \mathcal{O}(mn)$, where $\Delta$ is an upper bound on the absolute values of the entries of $A$. Assuming the Strong Exponential Time Hypothesis (SETH), there is no algorithm for (IPF) running in time $n^{f(m)} \cdot \mathcal{O}(m(\Delta + \|b\|_\infty))^{m-\delta}$ for any $\delta > 0$, and any computable function $f$.*

Notice that the exponent in the running time of the algorithm in Proposition 2 is improved to $m$ from $\mathcal{O}(m)$ in Proposition 1.

SETH is the hypothesis that CNF-SAT cannot be solved in time $(2-\epsilon)^n m^{\mathcal{O}(1)}$ on $n$-variable $m$-clause formulas for any constant $\epsilon$. ETH is the hypothesis that 3-SAT cannot be solved in time $2^{o(n)}$ on $n$-variable formulas. Both ETH and SETH were first introduced in the work of Impagliazzo and Paturi [10], which built upon earlier work of Impagliazzo, Paturi and Zane [11].

Notice that it is safe to remove duplicate columns in the input matrix of (IPF). Thus we can easily get an upper bound of $n \leq (2\Delta + 1)^m$. By using the proximity theorem of Eisenbrand and Weismantel [6], one can show that given an instance $(A, b)$ of (IPF), one can construct an equivalent instance $(A, b')$ of (IPF) in polynomial time such that $||b'||_\infty \leq m\Delta \cdot (2m\Delta + 1)^m$. In this work we consider the case of (IPF) when both $n$ and $\|b\|_\infty$ are much smaller than the above mentioned upper bounds.

One of the natural question that arises from Proposition 2 is whether the exponential dependence of $\|b\|_\infty$ can be improved significantly at the cost of super polynomial dependence on $n$. Our first theorem provides a conditional lower bound indicating that any significant improvements are unlikely.

**Theorem 1** *Unless the Exponential Time Hypothesis (ETH) fails, (IPF) with $m \times n$ matrix $A$ cannot be solved in time $n^{o(\frac{m}{\log m})} \cdot \|b\|_\infty^{o(m)}$ even when the constraint matrix $A$ is non-negative and each entry in any feasible solution is at most 2.*

Let us note that since the bound in Theorem 1 holds for a non-negative matrix $A$, we can always reduce (in polynomial time) the original instance of the problem to an equivalent instance where the maximum value $\Delta$ in the constraint matrix $A$ does not exceed $\|b\|_\infty$. Thus Theorem 1 also implies the conditional lower bound $n^{o(\frac{m}{\log m})} \cdot (\Delta \cdot \|b\|_\infty)^{o(m)}$. When $m = \mathcal{O}(n)$, our bound also implies the lower bound $(n \cdot m)^{o(\frac{m}{\log m})} \cdot (\Delta \cdot \|b\|_\infty)^{o(m)}$. We complement Theorem 1 by turning our focus to the dependence of algorithms solving (IPF) on $m$ alone, and obtaining the following theorem.

**Theorem 2** *Unless the Exponential Time Hypothesis (ETH) fails, (IPF) with $m \times n$ matrix $A$ cannot be solved in time $f(m) \cdot (n \cdot \|b\|_\infty)^{o(\frac{m}{\log m})}$ for any computable function $f$. The result holds even when the constraint matrix $A$ is non-negative and each entry in any feasible solution is at most* 1.

The difference between our first two theorems is the following. Although Theorem 1 provides a better dependence on $\|b\|_\infty$, Theorem 2 provides much more information on how the complexity of the problem depends on $m$. Since several parameters are involved in this running time estimation, a natural objective is to study the possible tradeoffs between them. For instance, consider the

$\mathcal{O}(m\Delta)^m \log(\Delta)\log(\Delta + \|b\|_\infty)$ time algorithm (Proposition 2) for (IPF). A natural follow up question is the following. Could it be that by allowing a significantly worse dependence (a superpolynomial dependence) on $n$ and $\|b\|_\infty$ and an *arbitrary* dependence on $m$, one might be able to improve the dependence on $\Delta$ alone? Theorem 2 provides a strong argument against such an eventuality. Indeed, since the lower bound of Theorem 2 holds even for non-negative matrices, it rules out algorithms with running time $f(m) \cdot \Delta^{o(\frac{m}{\log m})} \cdot (n \cdot \|b\|_\infty)^{o(\frac{m}{\log m})}$. Therefore, obtaining a subexponential dependence of $\Delta$ on $m$ even at the cost of a superpolynomial dependence of $n$ and $\|b\|_\infty$ on $m$, and an arbitrarily bad dependence on $m$ is as hard as obtaining a subexponential algorithm for 3-SAT.

We now motivate our remaining results. We refer the reader to Fig. 1 for a summary of our main results. It is straightforward to see that when the matrix $A$ happens to be

| Upper Bounds | Lower bounds |
|---|---|
| | no $n^{o(\frac{m}{\log m})} \cdot \|b\|_\infty^{o(m)}$ time algorithm under ETH (Theorem 1) (*even* for non-negative matrix $A$ and solution entries bounded by 2) |
| $(m \cdot \Delta)^{\mathcal{O}(m)} \cdot \|b\|_\infty^{\mathcal{O}(1)}$ [6, 12] | no $n^{\mathcal{O}(1)} \cdot \mathcal{O}(m(\Delta + \|b\|_\infty))^{m-\delta}$ time algorithm for $\delta > 0$ under SETH [12] |
| | no $(n \cdot m)^{o(\frac{m}{\log m})}(\Delta \cdot \|b\|_\infty)^{o(m)}$ algorithm when $m = \mathcal{O}(n)$ under ETH (consequence of Theorem 1) (*even* for non-negative matrix $A$ and solution entries bounded by 2) |
| | no $f(m) \cdot (n \cdot \|b\|_\infty)^{o(\frac{m}{\log m})}$ under ETH (Theorem 2) (*even* for non-negative matrix $A$ and solution entries bounded by 1) |
| $\mathcal{O}((\|b\|_\infty + 1)^{\mathsf{pw}+1} mn + m^2 n)$ (non-negative matrix $A$) (Theorem 6) | no $f(\mathsf{pw})(\|b\|_\infty + 1)^{(1-\epsilon)\mathsf{pw}}(mn)^{\mathcal{O}(1)}$ algorithm under SETH (Theorem 4) (*even* for non-negative matrix $A$) <br> no $f(\|b\|_\infty)(\|b\|_\infty + 1)^{(1-\epsilon)\mathsf{pw}}(mn)^{\mathcal{O}(1)}$ algorithm under SETH (Theorem 5) (*even* for non-negative matrix $A$) |
| $\mathcal{O}((\|b\|_\infty + 1)^{2\mathsf{bw}} mn + m^2 n)$ (non-negative matrix $A$) [1] | no $f(\mathsf{bw})(\|b\|_\infty + 1)^{(1-\epsilon)\mathsf{bw}}(mn)^{\mathcal{O}(1)}$ or $f(\|b\|_\infty)(\|b\|_\infty + 1)^{(1-\epsilon)\mathsf{bw}}(mn)^{\mathcal{O}(1)}$ algorithm under SETH (Theorem 3) (*even* for non-negative matrix $A$) |

**Fig. 1** A summary of our lower bound results in comparison with the relevant known upper bound results. Here, $n$ and $m$ are the number of variables and constraints respectively, $pw$ and $bw$ denote the path-width and branch-width of the column matroid of $A$ and $\|b\|_\infty$ denotes a bound on the largest absolute value in $b$ while $\Delta$ denotes a bound on the largest absolute value in $A$

non-negative, the algorithm of Papadimitriou [20] runs in time $\mathcal{O}((n \cdot \|b\|_\infty)^{m+1})$. Due to Theorems 1 and 2, the dynamic programming step of the algorithm of Papadimitriou for (IPF) when the maximum entry in a solution as well as in the constraint matrix is bounded, is already close to optimal. Consequently, any quest for "faster" algorithms for (IPF) must be built around the use of additional structural properties of the matrix $A$. Cunningham and Geelen [1] introduced such an approach by considering the *branch decomposition* of the matrix $A$. They were motivated by the fact that the result of Papadimitriou can be interpreted as a result for matrices of constant *rank* and branch-width is a parameter which is upper bounded by rank plus one. For a matrix $A$, the *column-matroid* of $A$ denotes the matroid whose elements are the columns of $A$ and whose independent sets are precisely the linearly independent sets of columns of $A$. We postpone the formal definitions of branch decomposition and branch-width till the next section. For (IPF) with a *non-negative* matrix $A$, Cunningham and Geelen showed that when the branch-width of the column-matroid of $A$ is constant, (IPF) is solvable in pseudo-polynomial time [1, 18].

**Proposition 3** (Cunningham and Geelen [1]) *(IPF) with a non-negative $m \times n$ matrix $A$ given together with a branch decomposition of its column matroid of width $k$, is solvable in time $\mathcal{O}((\|b\|_\infty + 1)^{2k}mn + m^2n)$.*

We analyze the complexity of (IPF) parameterized by the branch-width of $A$ by making use of SETH

and obtain the following lower bounds.

**Theorem 3** *Unless SETH fails, (IPF) with a non-negative $m \times n$ constraint matrix $A$ cannot be solved in time $f(bw)(\|b\|_\infty + 1)^{(1-\epsilon)bw}(mn)^{\mathcal{O}(1)}$ or $f(\|b\|_\infty)(\|b\|_\infty + 1)^{(1-\epsilon)bw}(mn)^{\mathcal{O}(1)}$, for any computable function $f$. Here $bw$ is the branchwidth of the column matroid of $A$.*

In recent years, SETH has been used to obtain several tight conditional bounds on the running time of algorithms for various optimization problems on graphs of bounded treewidth [17]. However, in order to be able to use SETH to prove lower bounds for (IPF) in combination with the branch-width of matroids, we have to develop new ideas.

In fact, Theorem 3 follows from stronger lower bounds we prove using the path-width of $A$ as our parameter of interest instead of the branch-width.

The parameter *path-width* is closely related to the notion of *trellis-width* of a linear code, which is a parameter commonly used in coding theory [9]. For a matrix $A \in \mathbb{R}^{m \times n}$, computing the path-width of the column matroid of $A$ is equivalent to computing the trellis-width of the linear code generated by $A$. Roughly speaking, the path-width of the column matroid of $A$ is at most $k$, if there is a permutation of the columns of $A$ such that in the matrix $A'$ obtained from $A$ by applying this column-permutation, for every $1 \leq i \leq n - 1$, the *dimension* of the subspace of $\mathbb{R}^m$ obtained by taking the intersection of the subspace of $\mathbb{R}^m$ spanned by the first $i$ columns with the subspace of $\mathbb{R}^m$ spanned by the remaining columns, is at most $k - 1$.

The value of the parameter path-width is always at least the value of branch-width and thus Theorem 3 follows from the following theorems.

**Theorem 4** *Unless SETH fails, (IPF) with even a non-negative $m \times n$ constraint matrix $A$ cannot be solved in time $f(k)(\|b\|_\infty + 1)^{(1-\epsilon)k}(mn)^{\mathcal{O}(1)}$ for any computable function $f$ and $\epsilon > 0$, where $k$ is the path-width of the column matroid of $A$.*

**Theorem 5** *Unless SETH fails, (IPF) with even a non-negative $m \times n$ constraint matrix $A$ cannot be solved in time $f(\|b\|_\infty)(\|b\|_\infty + 1)^{(1-\epsilon)k}(mn)^{\mathcal{O}(1)}$ for any computable function $f$ and $\epsilon > 0$, where $k$ is the path-width of the column matroid of $A$.*

Although the proofs of both lower bounds have a similar structure, we believe that there are sufficiently many differences in the proofs to warrant stating and proving them separately.

Note that although there is still a gap between the upper bound of Cunningham and Geelen from Proposition 3 and the lower bound provided by Theorem 3, the lower bounds given in Theorems 5 and 4 are asymptotically tight in the following sense. The proof of Cunningham and Geelen in [1] actually implies the upper bound stated in Theorem 6. We provide a self-contained proof in this paper for the reader's convenience.

**Theorem 6** *(IPF) with non-negative $m \times n$ matrix $A$ given together with a path decomposition of its column matroid of width $k$ is solvable in time $\mathcal{O}((\|b\|_\infty + 1)^{k+1}mn + m^2n)$.*

Then by Theorem 4, we cannot relax the $(\|b\|_\infty + 1)^k$ factor in Theorem 6 even if we allow in the running time an arbitrary function depending on $k$, while Theorem 5 shows a similar lower bound in terms of $\|b\|_\infty$ instead of $k$. Put together the results imply that no matter how much one is allowed to compromise on either the path-width or the bound on $\|b\|_\infty$, it is unlikely that the algorithm of Theorem 6 can be improved.

The path-width of matrix $A$ does not exceed its rank and thus the number of constraints in (IPF). Hence, similar to Proposition 3, Theorem 6 generalizes the result of Papadimitriou when restricted to non-negative matrices. Also we note that the assumption of non-negativity is unavoidable (without any further assumptions such as a bounded domain for the variables) in this setting because (IPF) is NP-hard when the constraint matrix $A$ is allowed to have negative values (in fact even when restricted to $\{-1, 0, 1\}$) and the branchwidth of the column matroid of $A$ is at most 3. A close inspection of the instances constructed by Cunningham and Geelen [1] in their NP-hardness reduction shows that the column matroids of the resulting constraint matrices are in fact direct sums of circuits, implying that even their *path-width* is bounded by 3.

## 1.1 Other related works and future research directions

In the conference version of the paper we asked whether the lower bound in Theorem 1 can be improved and this is answered by Knop et al. [15]. They prove that unless the Exponential Time Hypothesis (ETH) fails, (IPF) with $m \times n$ matrix $A \in \{0, 1\}^{m \times n}$ cannot be solved in time $2^{o(m \log m)} \cdot (n + \|b\|_\infty)^{o(m)}$. We also note that Ganian et al. [8] studied the parameterized complexity of (IPF) when parameterized by various combinations of $tw$ and $\|b\|_\infty$, where $tw$ is the incident treewidth of the input matrix

$A$. They gave a complete characterization of parameterized complexity results for (IPF) with non-negative constraint matrix when parameterized by all combinations of $tw$ and $\|b\|_\infty$. In particular, they showed that (IPF) with non-negative constraint matrix is FPT when parameterized by $tw$ and $\|b\|_\infty$.

While our SETH-based lower bounds for (IPF) with non-negative constraint matrix are tight for the path-width parameterization, there is a "$(\|b\|_\infty+1)^k$ to $(\|b\|_\infty+1)^{2k}$" gap between lower and upper bounds for branch-width parameterization. Closing this gap is the first natural question.

The proof of Theorem 3 given by Cunningham and Geelen consists of two parts. The first part bounds the number of potential partial solutions corresponding to any edge of the branch decomposition tree by $(\|b\|_\infty+1)^k$. The second part is the dynamic programming over the branch decomposition using the fact that the number of potential partial solutions is bounded. The bottleneck in the algorithm of Cunningham and Geelen is the following subproblem. We are given two vector sets $A$ and $B$ of partial solutions, each set of size at most $(\|b\|_\infty + 1)^k$. We need to construct a new vector set $C$ of partial solutions, where the set $C$ will have size at most $(\|b\|_\infty + 1)^k$ and each vector from $C$ is the *sum* of a vector from $A$ and a vector from $B$.

Thus to construct the new set of vectors, one has to go through all possible pairs of vectors from both sets $A$ and $B$, which takes time roughly $(\|b\|_\infty + 1)^{2k}$.

A tempting approach towards improving the running time of this particular step could be the use of *fast subset convolution* or *matrix multiplication* tricks, which work very well for "join" operations in dynamic programming algorithms over tree and branch decompositions of graphs [4, 5, 22], see also [3, Chapter 11]. Unfortunately, we have reason to suspect that these tricks may *not* help for matrices: solving the above subproblem in time $(\|b\|_\infty + 1)^{(1-\epsilon)2k}n^{\mathcal{O}(1)}$ for any $\epsilon > 0$ would imply that 3-SUM is solvable in time $n^{2-\epsilon}$, which is believed to be unlikely[1]. Indeed, consider an equivalent version of 3-SUM, named 3-SUM$'$, which is defined as follows. Given 3 sets of integers $A$, $B$ and $C$ each of cardinality $n$, and the objective is to check whether there exist $a \in A$, $b \in B$ and $c \in C$ such that $a + b = c$. Then, 3-SUM is solvable in time $n^{2-\epsilon}$ if and only if 3-SUM$'$ is as well (see Theorem 3.1 in [7]). However, the problem 3-SUM$'$ is equivalent to the most time consuming step in the algorithm of Theorem 3, where the integers in the input of 3-SUM$'$ can be thought of as length-one vectors. While this observation does not *rule out* the existence of an algorithm solving (IPF) with constraint matrices of branch-width $k$ in time $(\|b\|_\infty + 1)^{(1-\epsilon)2k}n^{\mathcal{O}(1)}$, it indicates that any interesting improvement in the running time would require a completely different approach.

## 1.2 Organization of the paper

The rest of the paper is organized as follows. There are two main technical parts to this paper. The first part (Sect. 3) is devoted to proving Theorem 1 and Theorem 2 (our ETH based lower bounds) while the second part (Sect. 4) is devoted to proving Theorem 4 and Theorem 5 (our SETH based lower bounds), and consequently, Theorem 3. For all

[1] The 3-SUM problem asks whether a given set of $n$ integers contains three elements that sum to zero.

our reductions, we begin by giving an overview in order to help the reader (especially in the SETH based reductions) navigate the technical details in the reductions. We then prove Theorem 5 in Sect. 4.3 and Theorem 6 in Sect. 5 (completing the results for constant path-width).

## 2 Preliminaries

We assume that the reader is familiar with basic definitions from linear algebra, matroid theory and graph theory.

### 2.1 Notations

We use $\mathbb{Z}_{\geq 0}$ and $\mathbb{R}$ to denote the sets of non negative integers and real numbers, respectively. For a positive integer $n$ and a non-negative integer $m$, we use $[n]$ and $[m, n]$ to denote the sets $\{1, \ldots, n\}$ and $\{m, m+1, \ldots, n\}$, respectively. For convenience, we say that $[0] = \emptyset$. For any two vectors $b, b' \in \mathbb{R}^m$ and $i \in [m]$, we use $b[i]$ to denote the $i^{th}$ coordinate of $b$ and we write $b' \leq b$, if $b'[i] \leq b[i]$ for all $i \in [m]$. We often use 0 to denote the zero-vector whose length will be clear from the context. For a matrix $A \in \mathbb{R}^{m \times n}$, $I \subseteq [m]$ and $J \subseteq [n]$, $A[I, J]$ denote the submatrix of $A$ obtained by the restriction of $A$ to the rows indexed by $I$ and columns indexed by $J$. For an $m \times n$ matrix $A$ and $n$-vector $v$, we can write $Av = \sum_{i=1}^{n} A_i v[i]$, where $A_i$ is the $i^{th}$ column of $A$. Here we say that $v[i]$ is a multiplier of column $A_i$. For convenience, in this paper, we consider 0 as an even number .

### 2.2 Branch-width of matroids

The notion of the branch-width of graphs, and implicitly of matroids, was introduced by Robertson and Seymour in [21]. Let $M = (U, \mathcal{F})$ be a matroid with universe set $U$ and family $\mathcal{F}$ of independent sets over $U$. We use $r_M$ to denote the rank function of $M$. That is, for any $S \subseteq U$, $r_M(S) = \max_{S' \subseteq S, S' \in \mathcal{F}} |S'|$. For $X \subseteq U$, the *connectivity function* of $M$ is defined as

$$\lambda_M(X) = r_M(X) + r_M(U \setminus X) - r_M(U) + 1$$

For a matrix $A \in \mathbb{R}^{m \times n}$, we use $M(A)$ to denote the column-matroid of $A$. In this case the connectivity function $\lambda_{M(A)}$ has the following interpretation. For $E = \{1, \ldots, n\}$ and $X \subseteq E$, we define

$$S(A, X) = \text{span}(A|X) \cap \text{span}(A|E \setminus X),$$

where $A|X$ is the set of columns of $A$ restricted to $X$ and $\text{span}(A|X)$ is the subspace of $\mathbb{R}^m$ spanned by the columns $A|X$. It is easy to see that the dimension of $S(A, X)$ is equal to $\lambda_{M(A)}(X) - 1$.

A tree is *cubic* if its internal vertices all have degree 3. A *branch decomposition* of a matroid $M$ with universe set $U$ is a cubic tree $T$ and a mapping $\mu$ which maps elements of $U$ to leaves of $T$. Let $e$ be an edge of $T$. Then the forest $T - e$ consists of two connected components $T_1$ and $T_2$. Thus every edge $e$ of $T$ corresponds to the partitioning of $U$ into two sets $X_e$ and $U \setminus X_e$ such that $\mu(X_e)$ are the leaves of $T_1$ and $\mu(U \setminus X_e)$ are the leaves of $T_2$. The *width* of an edge $e$ is $\lambda_M(X_e)$ and the width of a branch decomposition $(T, \mu)$ is the maximum edge width, where the maximum is taken over all edges of $T$. Finally, the *branch-width* of $M$ is the minimum width taken over all possible branch decompositions of $M$.

The *path-width* of a matroid is defined as follows. Recall that a *caterpillar* is a tree which is obtained from a path by attaching leaves to some vertices of the path. Then the path-width of a matroid is the minimum width of a branch decomposition $(T, \mu)$, where $T$ is a cubic caterpillar. Let us note that every mapping of elements of a matroid to the leaves of a cubic caterpillar corresponds to an ordering of these elements. Jeong, Kim, and Oum [13] gave a constructive fixed-parameter tractable algorithm to construct a path decomposition of width at most $k$ for a column matroid of a given matrix.

### 2.3 ETH and SETH

For $q \geq 3$, let $\delta_q$ be the infimum of the set of constants $c$ for which there exists an algorithm solving $q$-SAT with $n$ variables and $m$ clauses in time $2^{cn} \cdot m^{\mathcal{O}(1)}$. The *Exponential-Time Hypothesis (ETH)* and *Strong Exponential-Time Hypothesis (SETH)* are then formally defined as follows. ETH conjectures that $\delta_3 > 0$ and SETH that $\lim_{q\to\infty} \delta_q = 1$.

## 3 ETH lower bounds on pseudopolynomial solvability of (IPF)

In this section we prove Theorems 1 and 2.

### 3.1 Proof of Theorem 1

**Theorem** 1 *Unless the Exponential Time Hypothesis (ETH) fails,* (IPF)*with $m \times n$ matrix $A$ cannot be solved in time $n^{o(\frac{m}{\log m})} \cdot \|b\|_\infty^{o(m)}$ even when the constraint matrix $A$ is non-negative and each entry in any feasible solution is at most* 2.

Our proof is by a reduction from 3- CNF SAT to (IPF). There are exactly 2 variables in the (IPF) instance for each variable (one for each literal) and clause. For each clause we define two constraints. For each variable in the 3-CNF formula, we have a constraint, which is a selection gadget.

We now proceed to the formal description of the reduction. From a 3- CNF formula $\psi$ on $n$ variables and $m$ clauses we create an equivalent (IPF) instance $A_\psi x = b_\psi, x \geq 0$, where $A_\psi$ is a non-negative integer $(2m + n) \times 2(m + n)$ matrix and the largest entry in $b_\psi$ is 3. Our reduction can be easily seen to be a polynomial time reduction and we do not give an explicit analysis. Let $\psi$ be the input of 3- CNF SAT. Let $X = \{x_1, \ldots, x_n\}$ be the set of variables in $\psi$ and $\mathcal{C} = \{C_1, \ldots, C_m\}$ be the set of clauses in $\psi$. First we define the set of variables in the in the (IPF) instance. For each $x_i \in X$, we have two variables $x_i$ and $\overline{x}_i$ in the (IPF) instance $A_\psi x = b_\psi, x \geq 0$. For each $C_i \in \mathcal{C}$, we have two variables $Y_i$ and $Z_i$.

Now we define the set of constraints of $A_\psi x = b_\psi, x \geq 0$. For each $C_i = x \vee y \vee z$, we define two constraints

$$x + y + z + Y_i = 3 \quad \text{and} \tag{1}$$
$$Y_i + Z_i = 2. \tag{2}$$
$$\text{For each } i \in [n], \qquad x_i + \overline{x}_i = 1 \tag{3}$$

This completes the construction of (IPF) instance $A_\psi x = b_\psi, x \geq 0$. See Fig. 2 for an illustration. We now argue that this reduction correctly maps satisfiable 3-CNF formulas to feasible instances of (IPF) and vice versa.

**Lemma 1** *The formula $\psi$ is satisfiable if and only if $A_\psi x = b_\psi, x \geq 0$ is feasible.*

***Proof*** Suppose that the formula $\psi$ is satisfiable and let $\phi$ be a satisfying assignment of $\psi$. We set values for the variables $\{x_i, \overline{x}_i : i \in [n]\} \cup \{Y_i, Z_i : i \in [m]\}$ and prove that $A_\psi x = b_\psi$. For any $i \in [n]$, if $\phi(x_i) = 1$ we set $x_i = 1$ and $\overline{x}_i = 0$. Otherwise, we set $x_i = 0$ and $\overline{x}_i = 1$.

For every $i \in [m]$, we define

$$Y_i = \begin{cases} 0 & \text{if the number of literals set to 1 in } C_i \text{ by } \phi \text{ is 3,} \\ 1 & \text{if the number of literals set to 1 in } C_i \text{ by } \phi \text{ is 2,} \\ 2 & \text{otherwise,} \end{cases} \tag{4}$$

and

$$Z_i = 2 - Y_i = \begin{cases} 2 & \text{if the number of literals set to 1 in } C_i \text{ by } \phi \text{ is 3,} \\ 1 & \text{if the number of literals set to 1 in } C_i \text{ by } \phi \text{ is 2,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

| | $x_1$ | $\bar{x}_1$ | $x_2$ | $\bar{x}_2$ | $x_3$ | $\bar{x}_3$ | $x_4$ | $\bar{x}_4$ | $Y_1$ | $Z_1$ | $Y_2$ | $Z_2$ | $Y_3$ | $Z_3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$ | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | | | | | |
| | | | | | | | | | 1 | 1 | | | | |
| $C_2$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | | | 1 | | | |
| | | | | | | | | | | | 1 | 1 | | |
| $C_3$ | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | | | | | 1 | |
| | | | | | | | | | | | | | 1 | 1 |
| $x_1$ | 1 | 1 | | | | | | | | | | | | |
| $x_2$ | | | 1 | 1 | | | | | | | | | | |
| $x_3$ | | | | | 1 | 1 | | | | | | | | |
| $x_4$ | | | | | | | 1 | 1 | | | | | | |

**Fig. 2** An illustration of the matrix $A_\psi$ corresponding to the 3-CNF formula $\psi = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_4 \vee \bar{x}_2 \vee \bar{x}_3)$. The unfilled cells have 0 as the entry

We now proceed to prove that the above substitution of values to the variables is indeed a feasible solution. Towards this, we need to show that (1), (2), and (3) are satisfied. First consider (1). Let $C_i = x \vee y \vee z$. Since $\phi$ is a satisfying assignment, we have that $1 \le x + y + z \le 3$. Thus, by (4), we conclude that $x + y + z + Y_i = 3$. Because of (4),

(2) is satisfied. Since the values for $\{x_i, \overline{x}_i : i \in [n]\}$ is derived from an assignment $\phi$, (3) is satisfied.

For the converse direction of the statement of the lemma, suppose that there exists non-negative values for the set of variables $\{x_i, \overline{x}_i : i \in [n]\} \cup \{Y_i, Z_i : i \in [m]\}$, such that (1), (2), and (3) are satisfied. Now we need to show that $\psi$ is satisfiable. Because of (3), we know that exactly one of $x_i$ and $\overline{x}_i$ is set to one and other is set to zero. Next, we define an assignment $\phi$ and prove that $\phi$ is a satisfying assignment for $\psi$. For $i \in [n]$ we define

$$\phi(x_i) = \begin{cases} 1 & \text{if } x_i = 1, \\ 0 & \text{if } \overline{x}_i = 1. \end{cases}$$

We claim that $\phi$ satisfies all the clauses. Consider a clause $C_j = x \vee y \vee z$ where $j \in [m]$. Since $Y_j + Z_j = 2$ (by (2)), we have that $Y_i \in \{0, 1, 2\}$. Since $Y_i \in \{0, 1, 2\}$, by (1), at least one of $x$, $y$ or $z$ is set to one. This implies that $\phi$ satisfies $C_j$. This completes the proof of the lemma. □

By (2) and (3), for any satisfying assignment $\phi$, any variable $w \in \{x_i, \overline{x}_i : i \in [n]\}$, and any variable $W \in \{Y_i, Z_i : i \in [m]\}$, we have that $\phi(w) \le 1$ and $W \le 2$. The following lemma completes the proof of the theorem.

**Lemma 2** *If there is an algorithm for (IPF) running in time $n^{o(\frac{m}{\log m})} \|b\|_\infty^{o(m)}$, then ETH fails.*

***Proof*** By the Sparsification Lemma [11], we know that 3- CNF SAT on $n'$ variables and $cn'$ clauses, where $c$ is a constant, cannot be solved in time $2^{o(n')}$ time. Suppose there is an algorithm ALG for (IPF) running in time $n^{o(\frac{m}{\log m})} \|b\|_\infty^{o(m)}$. Then for a 3-CNF formula $\psi$ with $n'$ variables and $m' = cn$ clauses we create an instance $A_\psi x = b_\psi$, $x \ge 0$ of (IPF) as discussed in this section, in polynomial time, where $A_\psi$ is a matrix of dimension $(2cn' + n') \times (2(n' + cn'))$ and the largest entry in $b_\psi$ is 3. Then by Lemma 1, we can run ALG to test whether $\psi$ is satisfiable or not. This takes time

$$(2(cn' + n'))^{o(\frac{2cn'+n'}{\log(2cn'+n')})} \cdot 3^{o(2cn'+n')} = 2^{o(n')},$$

hence refuting ETH. □

## 3.2 Proof of Theorem 2

In this section we prove the following theorem.

**Theorem** 2 Unless the Exponential Time *Hypothesis (ETH) fails,* (IPF)*with $m \times n$ matrix $A$ cannot be solved in time $f(m) \cdot (n \cdot \|b\|_\infty)^{o(\frac{m}{\log m})}$ for any computable function*

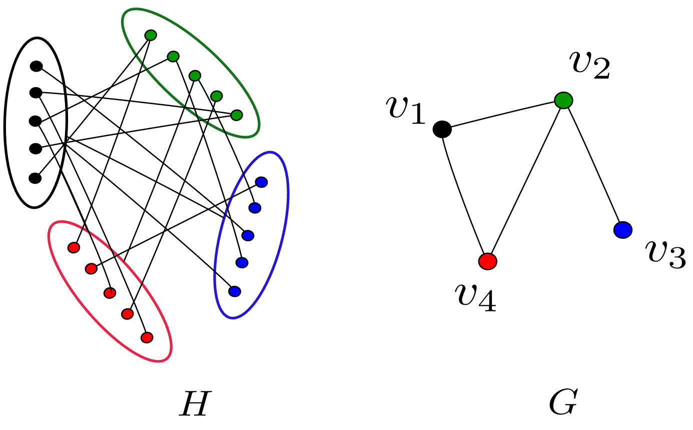


**Fig. 3** An illustration of an instance of Partitioned Subgraph Isomorphism

*$f$. The result holds even when the constraint matrix $A$ is non-negative and each entry in any feasible solution is at most* 1.

Towards proving Theorem 2 we use the ETH based lower bound result of Marx [19] for Partitioned Subgraph Isomorphism. For two graphs $G$ and $H$, a map $\phi \colon V(G) \mapsto V(H)$ is called a *subgraph isomorphism* from $G$ to $H$, if $\phi$ is injective and for any $\{u, v\} \in E(G)$, $\{\phi(u), \phi(v)\} \in E(H)$ (see Fig. 3 for an illustration).

Partitioned Subgraph Isomorphism
**Input:** Two graphs $G, H$, a bijection $c_G \colon V(G) \mapsto [\ell]$ and a function $c_H \colon V(H) \mapsto [\ell]$, where $\ell = |V(G)|$.
**Question:** Is there a subgraph isomorphism $\phi$ from $G$ to $H$ such that for any $v \in V(G)$, $c_G(v) = c_H(\phi(v))$?

**Lemma 3** (Corollary 6.3 [19]) If Partitioned Subgraph Isomorphism can be solved in time $f(G)n^{o(\frac{k}{\log k})}$, where $f$ is an arbitrary function, $n = |V(H)|$ and $k$ is the number of edges of the smaller graph $G$, then ETH fails.

To prove Theorem 2 we give a polynomial time reduction from Partitioned Subgraph Isomorphism to (IPF) such that for every instance $(G, H, c_G, c_H)$ of Partitioned Subgraph Isomorphism the reduction outputs an instance of (IPF) where the constraint matrix has dimension $\mathcal{O}(|E(G)|) \times \mathcal{O}(|E(H)|)$ and the largest value in the target vector is $\max\{|E(H)|, |V(H)|\}$.

Let $(G, H, c_G, c_H)$ be an instance of Partitioned Subgraph Isomorphism. Let $k = |E(G)|$ and $n = |V(H)|$. We construct an instance $Ax = b$ of (IPF) from $(G, H, c_G, c_H)$ in polynomial time. Without loss of generality we assume that $[n] = V(H)$ and that there are no isolated vertices in $G$. Hence, the number of vertices in $G$ is at most $2k$. Let $m = |E(H)|$. For each $e \in E(H)$ we assign a unique integer from $[m]$. Let $\alpha \colon E(H) \mapsto [m]$ be the bijection which represents the assignment mentioned above. For any $i, j \in [\ell]$, we use $E_H(i, j)$ as a shorthand for the set of edges of $H$ between $c_H^{-1}(i)$ and $c_H^{-1}(j)$. Finally, for ease of presentation we let $\{v_1, \ldots, v_\ell\} = V(G)$ and $c_G(v_i) = i$ for all $i \in [\ell]$, where $\ell = |V(G)|$.

For illustrative purposes, before proceeding to the formal construction, we give an informal description of the (IPF) instance we obtain from a specific instance of Partitioned Subgraph Isomorphism. Let $H$ and $G$ be the graphs in Fig. 3 and consider the graph $\widehat{H}$ obtained from $H$ as depicted in Fig. 4.

For every color $i \in [\ell]$ we have a column in $\widehat{H}$ and for every pair of distinct colors $i, j \in [\ell]$ such that $\{v_i, v_j\} \in E(G)$, we have a copy of $c_H^{-1}(i)$ in Column $i$ and Row $i$ and a copy of $c_H^{-1}(i)$ in Column $i$ and Row $j$. Thus, Column $i$ comprises at most $\ell$ copies of the vertices of $H$ whose image under $c_H$ is $i$ and Row $i$ comprises a copy of $c_H^{-1}(i)$ and additionally, a copy of every vertex $u$ of $H$ such that $v_{c_H(u)}$ is adjacent to $v_i$ in $G$. That is, the color of $u$ is "adjacent" to the color $i$ in $G$.

For a vertex $u \in V(H)$, we refer to the unique copy of $u$ in the $i^{th}$ row as the $i^{th}$ copy of $u$ in $\widehat{H}$. For every edge $e = \{a, b\} \in E(H)$ where $c_H(a) = i, c_H(b) = j$, and $\{v_i, v_j\} \in E(G)$, we have two copies of $e$ in $\widehat{H}$. The first copy of $e$ has as its endpoints, the $i^{th}$ copy of $a$ and the $i^{th}$ copy of $b$ and the second copy of $e$ has as its endpoints, the $j^{th}$ copy of $a$ and the $j^{th}$ copy of $b$. We now rephrase the Partitioned Subgraph Isomorphism problem (informally) as a problem of finding a certain type of subgraph in $\widehat{H}$, which in turn will point us in the direction of our (IPF) instance in a natural way. The rephrased problem statement is the following. Given $G$, $H$, $c_H$, $c_G$, and the resulting auxiliary graph $\widehat{H}$, find a set of $2|E(H)|$ edges in $\widehat{H}$ such that the following properties hold.

- (Selection) For every $\{v_i, v_j\} \in E(G)$, we pick a unique edge in $\widehat{H}$ with one endpoint in (Row $i$, Column $i$) and the other endpoint in (Row $i$, Column $j$) and we pick a unique edge with one endpoint in (Row $j$, Column $j$) and the other endpoint in (Row $j$, Column $i$).
- (Consistency 1) All the edges we pick from Row $i$ of $\widehat{H}$ share a common endpoint at the position (Row $i$, Column $i$).
- (Consistency 2) For any edge $e = \{a, b\} \in E(H)$ such that $c_H(a) = i, c_H(b) = j$, if the copy of $e$ in Row $i$ is selected in our solution then our solution contains the copy of $e$ in Row $j$ as well.

It is straightforward to see that a set of edges of $\widehat{H}$ which satisfy the stated properties imply a solution to our Partitioned Subgraph Isomorphism instance in an obvious way. In order to obtain our (IPF) instance, we create a variable for every edge in $\widehat{H}$ (or 2 for every edge in $E(H)$) and encode the properties stated above in the form of constraints. We now formally define the (IPF) instance output by our reduction.

The set of variables $x$ of the (IPF) instance is

$$\{x(\{a, b\}, c_H(a), c_H(b)) \colon \{a, b\} \in E(H)\}\,.$$

Notice that for any edge $\{a, b\} \in E(H)$, there exist an associated pair of variables, namely $x(\{a, b\}, c_H(a), c_H(b))$ and $x(\{a, b\}, c_H(b), c_H(a))$. Thus the dimension of $x$ is upper bounded by $2|E(H)| = 2m$. Recall that $\{v_1, \ldots, v_\ell\} = V(G)$ and $c_G(v_i) = i$ for all $i \in [\ell]$, where $\ell = |V(G)|$. For each $v_i \in V(G)$ we define $2d_G(v_i) - 1$ many constraints as explained below. Let $r = d_G(v_i)$ and $N_G(v_i) = \{v_{j_1}, \ldots, v_{j_r}\}$. The

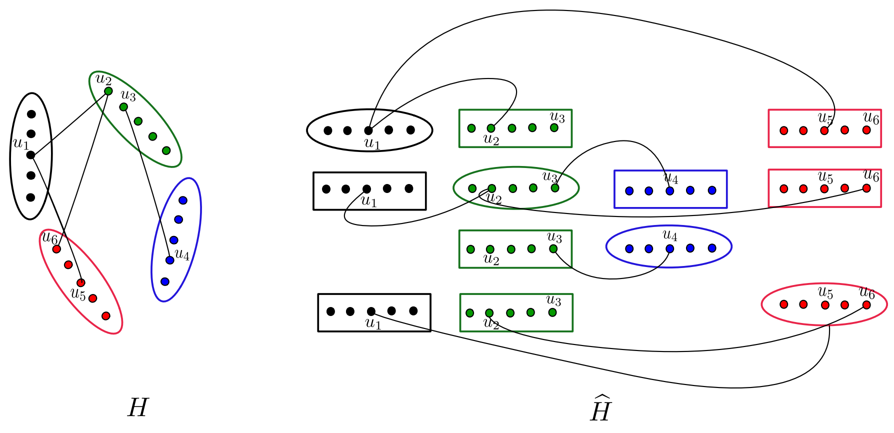


**Fig. 4** An illustration of the auxiliary graph $\widehat{H}$ capturing the representation of the vertices and *some* edges of $H$

constraints for $v_i \in V(G)$ are the following. For all $q \in [r]$,

$$\sum_{e \in E_H(i, j_q)} x(e, i, j_q) = 1 \tag{6}$$

The constraints of the form above enforce the (Selection) property described in our informal summary.

For all $q \in [r-1]$,

$$\sum_{\substack{\{a,b\} \in E_H(i, j_q) \\ a \in c_H^{-1}(i)}} a \cdot x(\{a, b\}, i, j_q) + \sum_{\substack{\{a,b'\} \in E_H(i, j_{q+1}) \\ a \in c_H^{-1}(i)}} (n-a) \cdot x(\{a, b'\}, i, j_{q+1}) = n \tag{7}$$

The constraints of the form above together enforce the (Consistency 1) property described in our informal summary.

For each $\{v_i, v_j\} \in E(G)$ with $i < j$, we define the following constraint in the (IPF) instance.

$$\sum_{\substack{\{a,b\} \in E_H(i, j) \\ a \in c_H^{-1}(i)}} \alpha(\{a, b\}) \cdot x(\{a, b\}, i, j) + \sum_{\substack{\{a,b\} \in E_H(i, j) \\ b \in c_H^{-1}(j)}} (m - \alpha(\{a, b\})) \cdot x(\{a, b\}, j, i) = m \tag{8}$$

The constraints of the form above together enforce the (Consistency 2) property described in our informal summary.

This completes the construction of the (IPF) instance $Ax = b, x \geq 0$. Notice that the construction of instance $Ax = b, x \geq 0$ can be done in polynomial time. Clearly, the number of rows in $A$ is $|E(G)| + \sum_{v \in V(G)} 2d_G(v) - 1 \leq 5k$ and number of columns in $A$ is $2m$. Now we prove the correctness of the reduction.

**Lemma 4** *$(G, H, c_G, c_H)$ is a* Yes *instance of* Partitioned Subgraph Isomorphism *if and only if $Ax = b, x \geq 0$ is feasible. Moreover, if $Ax = b, x \geq 0$ is feasible, then for any solution $x^*$, each entry of $x^*$ belongs to $\{0, 1\}$.*

***Proof*** Suppose $(G, H, c_G, c_H)$ is a Yes instance of Partitioned Subgraph Isomorphism. Let $\phi\colon V(G) \mapsto V(H)$ be a solution to $(G, H, c_G, c_H)$. Now we define a solution $x^* \in \{0, 1\}^{2m}$ to the instance $Ax = b, x \geq 0$ of (IPF). We know that for each edge $\{v_i, v_j\} \in E(G)$, $\{\phi(v_i), \phi(v_j)\} \in E(H)$. For each edge $\{v_i, v_j\} \in E(G)$, we set $x^*(\{\phi(v_i), \phi(v_j)\}, i, j) = x^*(\{\phi(v_i), \phi(v_j)\}, j, i) = 1$. For every other variable, we set its value to 0. Now we prove that $Ax^* = b$.

Towards that first consider (6). Fix a vertex $v_i \in V(G)$ and $v_{j_q} \in N_G(v_i)$. Since $\{v_i, v_{j_q}\} \in E(G), x^*(\{\phi(v_i), \phi(v_{j_q})\}, i, j_q) = 1$. Moreover, since $G$ is a simple graph, for any edge $e \in E_H(i, j_q) \setminus \{\{\phi(v_i), \phi(v_{j_q})\}\}$, $x^*(e, i, j_q) = 0$. This implies that (6) is satisfied by $x^*$. Next we consider (7). Fix a vertex $v_i \in V(G)$. Let $N_G(v_i) = \{v_{j_1}, \ldots, v_{j_r}\}$. Also, fix $q \in [r-1]$. We know that $\{v_i, v_{j_q}\}, \{v_i, v_{j_{q+1}}\} \in E(G)$. By the definition of $x^*$, we have that $x^*(e, i, j_q) = 1$ if and only if $e = \{\phi(v_i), \phi(v_{j_q})\}$ and $x^*(e', i, j_{q+1}) = 1$ if and only if $e' = \{\phi(v_i), \phi(v_{j_{q+1}})\}$. Thus we have that

$$\sum_{\substack{\{a,b\} \in E_H(i,j_q) \\ a \in c_H^{-1}(i)}} a \cdot x(\{a, b\}, i, j_q) + \sum_{\substack{\{a,b'\} \in E_H(i,j_{q+1}) \\ a \in c_H^{-1}(i)}} (n - a) \cdot x(\{a, b'\}, i, j_{q+1})$$
$$= \phi(v_i) + (n - \phi(v_i)) = n$$

That is, $x^*$ satisfies (7). Now we consider (8). Fix an edge $\{v_i, v_j\} \in E(G)$ where $i < j$. Again by the definition of $x^*$, we have that $x^*(e, i, j) = 1$ if and only if $e = \{\phi(v_i), \phi(v_j)\}$ and $x^*(e, j, i) = 1$ if and only if $e = \{\phi(v_i), \phi(v_j)\}$. This implies that (8) is satisfied by $x^*$. Therefore $Ax = b, x \geq 0$ is feasible.

Now we prove the converse direction of the lemma. Suppose that $Ax = b, x \geq 0$ is feasible and let $x' \in \mathbb{Z}_{\geq 0}^{2m}$ be a solution.

***Claim*** Let $i, j \in [\ell]$ such that $i \neq j$ and $\{v_i, v_j\} \in E(G)$. Then there exists exactly one edge $e \in E_H(i, j)$ such that $x'(e, i, j) = x'(e, j, i) = 1$. Moreover, for any $e' \in E_H(i, j) \setminus \{e\}, x'(e', i, j) = x'(e', j, i) = 0$.

***Proof*** By (6), we have that there exists exactly one edge $e_1 \in E_H(i, j)$ such that $x'(e_1, i, j) = 1$ and for all other edges $h \in E_H(i, j) \setminus \{e_1\}, x'(h, i, j) = 0$. Again by (6), we have that there exists exactly one edge $e_2 \in E_H(i, j)$ such that $x'(e_2, j, i) = 1$ and for all other edges $h \in E_H(i, j) \setminus \{e_2\}$, $x'(h, j, i) = 0$. By (8), we have that $e_1 = e_2$. This completes the proof of the claim. □

Now we define an injection $\phi\colon V(G) \mapsto V(H)$ and prove that indeed $\phi$ is a subgraph isomorphism from $G$ to $H$. For any $i, j \in [\ell]$ with $i \neq j$ and $\{v_i, v_j\} \in E(G)$

consider the edge $e = \{a, b\} \in E_H(i, j)$ such that $x'(\{a, b\}, i, j) = x'(\{a, b\}, j, i) = 1$ (by Claim 3.2, there exists exactly one such edge in $E_H(i, j)$). Let $a \in c_H^{-1}(i)$ and $b \in c_H^{-1}(j)$. Now we set $\phi(v_i) = a$ and $\phi(v_j) = b$. We claim that $\phi$ is well defined. Fix a vertex $v_i \in V(G)$. Let $r = d_G(v_i)$ and $N_G(v_i) = \{v_{j_1}, \dots, v_{j_r}\}$. By Claim 3.2, we know that for any $q \in [r]$, there exists exactly one edge $\{a_q, b_q\} \in E_H(i, j)$ such that $x'(\{a_q, b_q\}, i, j_q) = x'(\{a_q, b_q\}, j_q, i) = 1$. Here, $a_q \in c_H^{-1}(i)$ and $b_q \in c_H^{-1}(j_q)$. To prove that $\phi$ is well defined, it is enough to prove that $a_1 = a_2 = \dots = a_r = \phi(v_i)$. By (7), we have that for any $q \in [r-1]$, $a_q = a_{q+1}$. Also since $x'(\{a_q, b_q\}, i, j_q) = x'(\{a_q, b_q\}, j_q, i) = 1$ for all $q \in [r]$, we have that $a_1 = a_2 = \dots = a_r = \phi(v_i)$. From the construction of $\phi$, we have that for any $i, j \in [\ell], i \neq j, \phi(v_i) \in c_H^{-1}(i)$ and $\phi(v_j) \in c_H^{-1}(j)$. Moreover, $c_H^{-1}(i) \cap c_H^{-1}(j) = \emptyset$. This implies that $\phi$ is an injective map.

Now we prove that $\phi$ is an isomorphism from $G$ to $H$. Since $\phi(v_i) \in c_H^{-1}(i)$ for all $i \in [\ell]$, to prove that $\phi$ is an isomorphism, it is enough to prove that for any edge $\{v_i, v_j\} \in V(G), \{\phi(v_i), \phi(v_j)\} \in E(H)$. Fix an edge $\{v_i, v_j\} \in V(G)$ with $i < j$. By Claim 3.2, there exists exactly one edge $\{a, b\} \in E_H(i, j)$ such that $x'(\{a, b\}, i, j) = x'(\{a, b\}, j, i) = 1$, where $a \in c_H^{-1}(i)$ and $b \in c_H^{-1}(j)$. From the definition of $\phi$, we have that $\phi(v_i) = a$ and $\phi(v_j) = b$. That is, $\{\phi(v_i), \phi(v_j)\} = \{a, b\} \in E(H)$.

By Claim 3.2, we conclude that if $Ax = b, x \geq 0$ is feasible, then for any solution $x^*$, each entry of $x^*$ belongs to $\{0, 1\}$. This completes the proof of the lemma. □

***Proof of Theorem 2*** Let $(G, H, c_G, c_H)$ be an instance of Partitioned Subgraph Isomorphism. Let $Ax = b, x \geq 0$ be the instance of (IPF) constructed from $(G, H, c_G, c_H)$ as mentioned above. We know that the construction of $Ax = b, x \geq 0$ takes time polynomial in $n$, where $n = |V(H)|$. Also, we know that the number of rows and columns in $A$ is $\leq 5|E(G)|$ and $2|E(H)|$, respectively. Moreover, the maximum entry in $b$ is $\max\{|V(H)|, |E(H)|\}$.

Suppose there is an algorithm $\mathcal{A}$ for (IPF), running in time $f(m')(n' \cdot d')^{o\left(\frac{m'}{\log m'}\right)}$ on instances where the constraint matrix is non-negative and is of dimension $m' \times n'$, and the maximum entry in the target vector is $d'$. Then, by running $\mathcal{A}$ on $Ax = b, x \geq 0$ and applying Lemma 4, we solve Partitioned Subgraph Isomorphism in time $f(G)n^{o\left(\frac{k}{\log k}\right)}$. Thus by Lemma 3, ETH fails. This completes the proof of the theorem. □

## 4 Path-width parameterization: SETH bounds

In this section we prove Theorems 4 and 5.

### 4.1 Overview of our reductions

We prove Theorems 4 and 5 by giving reductions from CNF- SAT. At this point, one might be tempted to start the reduction from $k$- CNF SAT as seen in [2]. However, the fact that in our case we also need to control the path-width of the reduced instance poses serious technical difficulties if one were to take this route. Therefore, we take a different route and reduce from CNF- SAT which allows us to construct appropriate gadgets for propagation of consistency in our instance while simultaneously controlling the path-width. Moreover, the parameters in the reduced instances are required to obey certain strict conditions. For example, the reduction we give to prove Theorem 4 must output an instance of (IPF), where the path-width of the column matroid $M(A)$ of the constraint matrix $A$ is a constant. Similarly, in the reduction used to prove Theorem 5, we need to construct an instance of (IPF) where the largest entry in the target vector is upper bounded by a constant. These stringent requirements on the parameters make the SETH-based reductions quite challenging. However, reductions under SETH can take super polynomial time—they can even take $2^{(1-\epsilon)n}$ time for some $\epsilon > 0$, where $n$ is the number of variables in the instance of CNF- SAT. This freedom to avail exponential time in SETH-based reductions is used crucially in the proofs of Theorems 4 and 5.

Now we give an overview of the reduction used to prove Theorem 4. Let $\psi$ be an instance of CNF- SAT with $n$ variables and $m$ clauses. Given $\psi$ and a fixed constant $c \geq 2$, we construct an instance $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$ of (IPF) satisfying certain properties. Since for every $c \geq 2$, we have a different $A_{(\psi,c)}$ and $b_{(\psi,c)}$, this can be viewed as a family of instances of (IPF). In particular our main technical lemma is the following.

**Lemma 5** *Let $\psi$ be an instance of* CNF- SAT *with $n$ variables and $m$ clauses. Let $c \geq 2$ be a fixed integer. Then, in time $\mathcal{O}(m^2 2^{\frac{n}{c}})$, we can construct an instance $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$, of (IPF) with the following properties.*

*(a.) $\psi$ is satisfiable if and only if $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$ is feasible.*
*(b.) The matrix $A_{(\psi,c)}$ is non-negative and has dimension $\mathcal{O}(m) \times \mathcal{O}(m 2^{\frac{n}{c}})$.*
*(c.) The path-width of the column matroid of $A_{(\psi,c)}$ is at most $c + 4$.*
*(d.) The largest entry in $b_{(\psi,c)}$ is at most $2^{\lceil \frac{n}{c} \rceil} - 1$.*

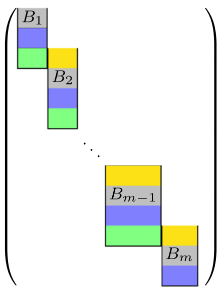
Once we have Lemma 5, the proof of Theorem 4 follows from the following observation: if we have an algorithm $\mathcal{A}$ solving (IPF) in time $f(k)(\|b\|_\infty + 1)^{(1-\epsilon)k}(mn)^a$ for some $\epsilon, a > 0$, then we can use this algorithm to refute SETH, where $k$ is the path-width of the column matroid of the input matrix. In particular, given an instance $\psi$ of CNF- SAT, we choose an appropriate $c$ depending only on $\epsilon$ and $a$, construct an instance $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$, of (IPF), and run $\mathcal{A}$ on it. Our careful choice of $c$ will imply a faster algorithm for CNF- SAT, refuting SETH. More formally, we choose $c$ to be an integer such that $(1-\epsilon) + \frac{4(1-\epsilon)}{c} + \frac{a}{c} < 1$. Then the total running time to test whether $\psi$ is satisfiable, is the time require to construct $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$ plus the time required by $\mathcal{A}$ to solve the constructed instance of (IPF). That is, the time required to test whether $\psi$ is satisfiable is

$$\begin{aligned}\mathcal{O}(m^2 2^{\frac{n}{c}}) + f(c+4) 2^{\frac{n}{c}(1-\epsilon)(c+4)} 2^{\frac{a \cdot n}{c}} m^{\mathcal{O}(1)} &= 2^{\left((1-\epsilon) + \frac{4(1-\epsilon)}{c} + \frac{a}{c}\right)n} m^{\mathcal{O}(1)} \\ &= 2^{\epsilon' n} m^{\mathcal{O}(1)},\end{aligned}$$

where $\epsilon' < 1$ is a constant depending on the choice of $c$. It is important to note that the utility of the reduction described in Lemma 5 is extremely sensitive to the value of

$$\begin{pmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \; 1 \end{pmatrix}$$

**(a)** A matrix $B$ for which path-width of its column matroid is 1



**(b)** A pictorial representation of the matrix $A_{(\psi,c)}$. Here, the different shadings of $B_i$ correspond to different parts of the matrix $B_i$ for any $i \in [m]$.

**Fig. 5** Comparison of $A_{(\psi,c)}$ with a low path-width matrix

the numerical parameters involved. In particular, even when the path-width blows up slightly, say up to $\delta c$, or when the largest entry in $b_{(\psi,c)}$ blows up slightly, say up to $2^{\delta \frac{n}{c}}$, for some $\delta > 1$, then the calculation above will *not* give us the desired refutation of SETH. Thus, the challenging part of the reduction described in Lemma 5 is making it work under these strict restrictions on the relevant parameters.

As stated in Lemma 5, in our reduction, we need to obtain a constraint matrix with small path-width. An important first step towards this is understanding what a matrix of small path-width looks like. We first give an intuitive description of the structure of such matrices. Let $A$ be a $m \times n$ matrix of small path-width and let $M(A)$ be the column matroid of $A$. For any $i \in \{1, \ldots, n-1\}$, recall that $A|\{1, \ldots i\}$ is the set of columns (or vectors) in $A$ whose index is at most $i$ (that is, the first $i$ columns) and $A|\{i+1, \ldots n\}$ is the set of columns with index strictly greater than $i$. The path-width of $M(A)$ is at most

$$\max_i \dim\langle \mathrm{span}(A|\{1, \ldots, i\}) \cap \mathrm{span}(A|\{i+1, \ldots, n\})\rangle + 1.$$

Hence, in order to obtain a bound on the pathwidth, it is sufficient to bound $\dim\langle \mathrm{span}(A|\{1, \ldots, i\}) \cap \mathrm{span}(A|\{i+1, \ldots, n\})\rangle$ for every $i \in [n]$. Consider for example, the matrix $B$ given in Fig. 5a. The path-width of $M(B)$ is clearly at most 1. In *our* reduced instance, the constructed constraint matrix $A_{(\psi,c)}$ will be an appropriate extension of $B$. That is $A_{(\psi,c)}$ will have the "same form" as $B$ but with each 1 replaced by a submatrix of order $\mathcal{O}(c) \times n'$ for some $n'$. See Fig. 5b for a pictorial representation of $A_{(\psi,c)}$.

The construction used in Lemma 5 takes as input an instance $\psi$ of CNF- SAT with $n$ variables and a fixed integer $c \geq 2$, and outputs an instance $A_{(\psi,c)}x = b_{(\psi,c)}$, $x \geq 0$, of (IPF), that satisfies all four properties of the lemma. Let $X$ denote the set of variables in the input CNF-formula $\psi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$. For the purposes of the present discussion we assume that $c$ divides $n$. We partition the variable set $X$ into $c$ blocks $X_0, \ldots, X_{c-1}$, each of size $\frac{n}{c}$. Let $\mathcal{X}_i$, $i \in \{0, \ldots, c-1\}$, denote the set of assignments of variables corresponding to $X_i$. Set $\ell = \frac{n}{c}$ and $L = 2^{\ell}$. Clearly, the size of $\mathcal{X}_i$ is upper bounded by $2^{\frac{n}{c}} = 2^\ell = L$. We denote the assignments in $\mathcal{X}_i$ by $\phi_0(X_i), \phi_1(X_i), \ldots, \phi_{L-1}(X_i)$. To construct the matrix $A_{(\psi,c)}$, we view each of these assignments as a different assignment for each clause. In other words we have separate sets of variables in the constraints corresponding to different pairs $(C_r, X_i)$, where $C_r$ is a clause and $X_i$ is a block in the partition of $X$. That is for each clause $C_r$ and block $X_i$, we have variables $\{y_{C_r,i,a}\ a \in [0, 2L-1]\ \}$. In other words for each $C_r$ and assignment $\phi_a(X_i)$, $a \in [0, L-1]$, we have two variables $y_{C_r,i,2a}$ and $y_{C_r,i,2a+1}$. For any clause $C_r$, $i \in [0, c-1]$ and $a \in [0, 2L-1]$, assigning value 1 to $y_{C,i,a}$ corresponds to choosing an assignment $\phi_{\lfloor \frac{a}{2} \rfloor}(X_i)$ for $X_i$. In our reduction we will create the following set of constraints.

$$\sum_{\substack{i\in[c],a\in[0,2L-1] \text{ such that} \\ a \text{ is even and} \\ \phi_{\lfloor \frac{a}{2}\rfloor}(X_i) \text{ satisfies } C}} y_{C,i,a} = 1 \qquad \text{for all } C \in \mathcal{C} \tag{9}$$

$$\sum_{a\in[0,2L-1]} y_{C,i,a} = 1 \qquad \text{for all } C \in \mathcal{C} \text{ and } i \in [0, c-1] \tag{10}$$

Equation (9) takes care of satisfiability of clauses, while Eq. (10) allows us to pick only one assignment from $\{\phi_0(X_i), \phi_1(X_i), \ldots, \phi_{L-1}(X_i)\}$ per clause $C$ and block $X_i$. Note that this implies that we will choose an assignment in $\mathcal{X}_i$ for each clause $C_r$. That way we might choose $m$ assignments from $\mathcal{X}_i$ corresponding to $m$ different clauses. However, for the backward direction of the proof, it is important that we choose the *same* assignment from $\mathcal{X}_i$ for each clause. This will ensure that we have selected an assignment to the variables in $X_i$. Towards this we will have a third set of constraints as follows. For all $r \in [m-1]$ and $i \in [0, c-1]$

$$\sum_{a\in[0,2L-1]} \left(\lfloor \frac{a}{2} \rfloor \cdot y_{C_r,i,a}\right) + \left((L-1-\lfloor \frac{a}{2} \rfloor) y_{C_{r+1},i,a}\right) = L-1 \tag{11}$$

Eq. (11) enforce consistencies of assignments of blocks across clauses in a *sequential* manner. That is, for any block $X_i$, we make sure that the two variables set to 1 corresponding to $(C_r, X_i)$ and $(C_{r+1}, X_i)$ are consistent for any $r \in \{1, \ldots, m-1\}$, as opposed to checking the consistency for every pair $(C_r, X_i)$ and $(C_{r'}, X_i)$ for $r \neq r'$. Thus in some sense these consistencies *propagate*. Furthermore, the idea of making consistency in a sequential manner also allows us to bound the path-width of column matroid of $A_{(\psi,c)}$ by $c+4$.

The proof technique for Theorem 5 is similar to that for Theorem 4. This is achieved by modifying the matrix $A_{(\psi,c)}$ constructed in the reduction described for Lemma 5. The largest entry in $A_{(\psi,c)}$ is $2^{\frac{n}{c}} - 1$ (see Eq. (11)). So each of these values can be represented by a binary string of length at most $\ell = \frac{n}{c}$. We remove each row, say row indexed by $\gamma$, with entries greater than 1 and replace it with $\frac{n}{c}$ rows, $\gamma_1, \ldots, \gamma_\ell$. Where, for any $j$, if the value $A_{(\psi,c)}[\gamma, j] = W$ then $A_{(\psi,c)}[\gamma_k, j] = \eta_k$, where $\eta_k$ is the $k^{th}$ bit in the $\ell$-sized binary representation of $W$. This modification reduces the largest entry in $A_{(\psi,c)}$ to 1 and increases the path-width from constant to approximately $n$.

Finally, we set all the entries in $b_{(\psi,c)}$ to be 1. This concludes the overview of our reductions and we now proceed to a detailed exposition.

### 4.2 Proof of Theorem 4

In this section we provide a Proof of Theorem 4, which states that unless SETH fails, (IPF) with non-negative matrix $A$ cannot be solved in time $f(k)(\|b\|_\infty + 1)^{(1-\epsilon)k}(mn)^{\mathcal{O}(1)}$ for any function $f$ and $\epsilon > 0$, where $d = \max\{b[1], \ldots, b[m]\}$ and $k$ is the path-width of the column matroid of $A$.

Towards the proof of Theorem 4, we first present the proof of our main technical lemma (Lemma 5), which we restate here for the sake of completeness.

**Lemma** 5 *Let $\psi$ be an instance of* CNF- SAT *with $n$ variables and $m$ clauses. Let $c \geq 2$ be a fixed integer. Then, in time $\mathcal{O}(m^2 2^{\frac{n}{c}})$, we can construct an instance $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$, of* (IPF) *with the following properties.*

(a.) *$\psi$ is satisfiable if and only if $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$ is feasible.*
(b.) *The matrix $A_{(\psi,c)}$ is non-negative and has dimension $\mathcal{O}(m) \times \mathcal{O}(m2^{\frac{n}{c}})$.*
(c.) *The* path-width*of the column matroid of $A_{(\psi,c)}$ is at most $c + 4$.*
(d.) *The largest entry in $b_{(\psi,c)}$ is at most $2^{\lceil \frac{n}{c} \rceil} - 1$.*

Let $\psi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ be an instance of CNF- SAT with variable set $X = \{x_1, x_2, \ldots, x_n\}$ and let $c \geq 2$ be a fixed constant given in the statement of Lemma 5. We construct the instance $A_{(\psi,c)}x = b_{(\psi,c)}, x \geq 0$ of (IPF) as follows.

**Construction.** Let $\mathcal{C} = \{C_1, \ldots, C_m\}$. Without loss of generality, we assume that $n$ is divisible by $c$, otherwise we add at most $c$ dummy variables to $X$ such that $|X|$ is divisible by $c$. We divide $X$ into $c$ blocks $X_0, X_1, \ldots, X_{c-1}$. That is $X_i = \{x_{\frac{i \cdot n}{c}+1}, x_{\frac{i \cdot n}{c}+2}, \ldots, x_{\frac{(i+1)\cdot n}{c}}\}$ for each $i \in [0, c-1]$. Let $\ell = \frac{n}{c}$ and $L = 2^\ell$. For each block $X_i$, there are exactly $2^\ell$ assignments. We denote these assignments by $\phi_0(X_i), \phi_1(X_i), \ldots, \phi_{L-1}(X_i)$.

Now, we create $m \cdot c \cdot 2^{\ell+1}$ variables; they are named $y_{C,i,a}$, where $C \in \mathcal{C}$, $i \in [0, c-1]$ and $a \in [0, 2L-1] = [0, 2^{\ell+1} - 1]$. In other words, for a clause $C$, a block $X_i$ and an assignment $\phi_a(X_i)$, we create two variables; they are $y_{C,i,2a}$ and $y_{C,i,2a+1}$. Then, we create the (IPF) constraints given by Eqs. (9), (10), and (11).

This completes the construction of (IPF) instance. Let $A_{(\psi,c)}y = b_{(\psi,c)}$ be the (IPF) instance defined using Eq. (9), (10), and (11). The purpose of Eq. (9) is to ensure satisfiability of all the clauses. Because of Eq. (10), for each clause $C$ and for each block $X_i$, we select only one assignment. Notice, that, so far it is allowed to choose many assignments from a block $X_i$, for different clauses. To ensure the consistency of assignments in each block across clauses, we added a system of constraints (Eq. (11)). Equation (11) ensures the consistency of assignments in the adjacent clauses (in the order $C_1, \ldots, C_m$). Thus, the consistency of assignments propagates in a sequential manner. Notice that number constraints defined by Eqs. (9), (10), and (11) are $m$, $m \cdot c$ and $(m-1) \cdot c$, respectively. The number of variables is $m \cdot c \cdot 2^{\ell+1}$. Also notice that all the coefficients in Eqs. (9), (10) and (11) are non-negative. This implies that $A_{(\psi,c)}$ is non-negative and has dimension $\mathcal{O}(m) \times \mathcal{O}(m2^{\frac{n}{c}})$. Thus, the property $(b.)$ of Lemma 5 is satisfied. The largest entry in $b_{(\psi,c)}$ is $L - 1 = 2^{\lceil \frac{n}{c} \rceil} - 1$ (see Eq. (11))

and hence the property $(d.)$ of Lemma 5 is satisfied. Now we prove property $(a.)$ of Lemma 5.

"From here on, we use $A$ instead of $A_{(\psi,c)}$ and $b$ instead of $b_{(\psi,c)}$ for clarity."

**Lemma 6** *Formula $\psi$ is satisfiable if and only if there exists $y^* \in \mathbb{Z}^{n'}_{\geq 0}$ such that $Ay^* = b$, where $n' = m \cdot c \cdot 2^{\ell+1}$ is the number of columns in $A$.*

***Proof*** Let $Y = \{y_{C,i,a} \mid C \in \mathcal{C}, i \in [0, c-1], a \in [0, 2L-1]\}$. Suppose $\psi$ is satisfiable. We need to show that there is an assignment of non-negative integer values to the variables in $Y$ such that Eqs. (9), (10) and (11) are satisfied. Let $\phi$ be a satisfying assignment of $\psi$. Then, there exist $a_0, a_1, \ldots, a_{c-1} \in [0, L-1]$ such that $\phi$ is the union of $\phi_{a_0}(X_0), \phi_{a_1}(X_1), \ldots, \phi_{a_{c-1}}(X_{c-1})$. Any clause $C \in \mathcal{C}$ is satisfied by at least one of the assignments $\phi_{a_0}(X_0), \phi_{a_1}(X_1), \ldots, \phi_{a_{c-1}}(X_{c-1})$. For each $C$, we fix an arbitrary $i \in [0, c-1]$ such that the assignment $\phi_{a_i}(X_i)$ satisfies clause $C$. Let $\alpha$ be a function which fixes these assignments for each clause. That is, $\alpha : \mathcal{C} \to [0, c-1]$ such that the assignment $\phi_{a_{\alpha(C)}}(X_{\alpha(C)})$ satisfies the clause $C$ for every $C \in \mathcal{C}$. Now we assign values to $Y$ and prove that these assignment satisfy Eqs. (9), (10) and (11).

$$y_{C,i,a} = \begin{cases} 1, & \text{if } \alpha(C) = i \text{ and } a \text{ is even and } \lfloor \frac{a}{2} \rfloor = a_i \\ 1, & \text{if } \alpha(C) \neq i \text{ and } a \text{ is odd and } \lfloor \frac{a}{2} \rfloor = a_i \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Notice that, by Eq. (12), for any fixed $C \in \mathcal{C}$, exactly $c$ variables from $\{y_{C,i,a} \mid i \in [0, c-1], a \in [2^{\ell+1}]\}$ are set to 1. They are $y_{C,\alpha(C),2a_{\alpha(C)}}$ and the variables in the set $Y_C = \{y_{C,i,2a_i+1} \mid i \neq \alpha(C)\}$. This implies that in Eq. (9), only $y_{C,\alpha(C),2a_{\alpha(C)}}$ is set to 1, and hence Eq. (9) is satisfied. Now consider Eq. (10) for any fixed $C \in \mathcal{C}$ and $i \in [0, c-1]$. By Eq. (12), exactly one variable from $\{y_{C,i,a} \mid a \in [0, 2L-1]\}$ is set to 1, and hence Eq. (10) is satisfied. Now consider Eq. (11) for fixed $r \in [m-1]$ and $i \in [0, c-1]$. By Eq. (12), exactly one variable from each set $\{y_{C_r,i,a} \mid a \in [0, 2L-1]\}$ and $\{y_{C_{r+1},i,a} \mid a \in [0, 2L-1]\}$ are set to 1; they are one variable each from $\{y_{C_r,i,2a_i}, y_{C_r,i,2a_i+1}\}$ and $\{y_{C_{r+1},i,2a_i}, y_{C_{r+1},i,2a_i+1}\}$. So we get the following when we substitute values for $Y$ in Eq. (11).

$$\sum_{a \in [0,2L-1]} \left( \lfloor \frac{a}{2} \rfloor \cdot y_{C_r,i,a} \right) + \left( (L - 1 - \lfloor \frac{a}{2} \rfloor) \cdot y_{C_{r+1},i,a} \right) = a_i + L - 1 - a_i = L - 1$$

Hence, Eq. (11) is satisfied by the assignments given in Eq. (12).

Now we need to prove the converse direction. Suppose there are non-negative integer assignments to $Y$ such that Eqs. (9), (10) and (11) are satisfied. Now we need to show that $\psi$ is satisfiable. Because of Eq. (10) all the variables in $Y$ are set to 0 or 1. We will extract a satisfying assignment from the values assigned to variables in $Y$. Towards that, first we prove the following claim.

***Claim*** Let $y_{C_1,i,a} = 1$ for some $i \in [0, c-1]$ and $a \in [0, 2L-1]$. Then, for any $C' \in \mathcal{C}$, exactly one among $\{y_{C',i,2\lfloor \frac{a}{2} \rfloor}, y_{C',i,2\lfloor \frac{a}{2} \rfloor+1}\}$ is set to 1.

***Proof*** Towards the proof, we first show that if $y_{C_r,i,a} = 1$ for some $r \in [m-1]$, then exactly one among $\{y_{C_{r+1},i,2\lfloor \frac{a}{2} \rfloor}, y_{C_{r+1},i,2\lfloor \frac{a}{2} \rfloor+1}\}$ is set to 1. By Eq. (10) and the fact that $y_{C_r,i,a} = 1$, we get that

$$\sum_{a' \in [0,2L-1]} \left( \lfloor \frac{a'}{2} \rfloor \cdot y_{C_r,i,a'} \right) = \lfloor \frac{a}{2} \rfloor. \tag{13}$$

Eqs. (11) and (13) imply that

$$\sum_{a' \in [0,2L-1]} \left( (L-1-\lfloor \frac{a'}{2} \rfloor) \cdot y_{C_{r+1},i,a'} \right) = L-1-\lfloor \frac{a}{2} \rfloor. \tag{14}$$

By Eqs. (10) and (14), we get that exactly one among $\{y_{C_{r+1},i,2\lfloor \frac{a}{2} \rfloor}, y_{C_{r+1},i,2\lfloor \frac{a}{2} \rfloor+1}\}$ is set to 1. Thus, by applying the above arguments for $i = 1, 2, \ldots, m-1$, we get that for any $C' \in \mathcal{C} \setminus \{C_1\}$, exactly one among $\{y_{C',i,2\lfloor \frac{a}{2} \rfloor}, y_{C',i,2\lfloor \frac{a}{2} \rfloor+1}\}$ is set to 1.

Suppose $C' = C_1$. Then, by Eq. (10) and the assumption that $y_{C_1,i,a} = 1$, exactly one among $\{y_{C_1,i,2\lfloor \frac{a}{2} \rfloor}, y_{C_1,i,2\lfloor \frac{a}{2} \rfloor+1}\}$ is set to 1. □

Now we define a satisfying assignment for $\psi$. Towards that we give assignments to all blocks $X_0, \ldots, X_{c-1}$, such that the union of these assignments satisfies $\psi$. Fix any block $X_i$. By Eq. (10), exactly one among $\{y_{C_1,i,a} \mid a \in [0, 2L-1]\}$ is set to 1. Let $a_i \in [0, 2L-1]$ such that $y_{C_1,i,a_i} = 1$. Then we choose the assignment $\phi_{\lfloor \frac{a_i}{2} \rfloor}(X_i)$ for $X_i$. Let $\phi$ be the assignment of $X$ which is the union of $\psi_{\lfloor \frac{a_1}{2} \rfloor}(X_1)$, $\psi_{\lfloor \frac{a_2}{2} \rfloor}(X_2), \ldots, \psi_{\lfloor \frac{a_{c-1}}{2} \rfloor}(X_{c-1})$. By Eq. (9) and Claim 4.2, $\phi$ satisfies all the clauses in $\mathcal{C}$ and hence $\psi$ is satisfiable. □

Now we need to prove property $(c.)$ of Lemma 5. That is the path-width of $A$ is at most $c+4$. Towards that we need to understand the structure of matrix $A$. We decompose the matrix $A$ into $m$ disjoint submatrices $B_1, \ldots B_m$ which cover all the non-zero entries in the matrix $A$. First we define some notation and fix the column indices of $A$ corresponding to the variables in the constraints. Let $Y$ denote the set $\{y_{C,i,a} \mid C \in \mathcal{C}, i \in [0, c-1], a \in [0, 2L-1]\}$ of variables in the constraints defined by Eqs. (9), (10) and (11). These variables can be partitioned into $\biguplus_{C \in \mathcal{C}} Y_C$, where $Y_C = \{y_{C,i,a} \mid i \in [0, c-1], a \in [0, 2L-1]\}$. Further for each $C \in \mathcal{C}$, $Y_C$ can be partitioned into $\bigcup_{i \in [0,c-1]} Y_{C,i}$, where $Y_{C,i} = \{y_{C,i,a} \mid a \in [0, 2L-1]\}$. The set of columns indexed by $[r \cdot c2^{\ell+1}] \setminus [(r-1) \cdot c \cdot 2^{\ell+1}]$, for any $r \in [m]$, corresponds to the set of variables in $Y_{C_r}$. Among the set of columns corresponding to $Y_C$, the first $2^{\ell+1}$ columns corresponds to the variables in $Y_{C,1}$, second $2^{\ell+1}$ columns corresponds to the variables in $Y_{C,2}$, and so on. Among the set of columns corresponds to $Y_{C,i}$ for any $C \in \mathcal{C}$ and $i \in [0, c-1]$, the first two columns corresponds to the variable $y_{C,i,0}$ and $y_{C,i,1}$, and second two columns corresponds to the variables $y_{C,i,2}$ and $y_{C,i,3}$, and so on.

Now we move to the description of $B_j$, $j \in [m]$. The matrix $B_j$ will cover the coefficients of $Y_{C_j}$ in Eqs. (9), (10) and (11). In other words $B_j$ covers the non-zero entries in the columns corresponding to $Y_{C_j}$, i.e, in the columns of $A$ indexed by

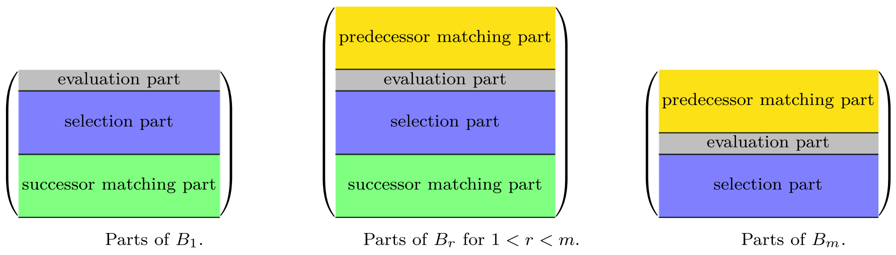


**Fig. 6** Parts of $B_r$

$[j \cdot c \cdot 2^{\ell+1}] \setminus [(j-1) \cdot c \cdot 2^{\ell+1}]$. Now we explain these submatrices. Each matrix $B_j$ has $c \cdot 2^{\ell+1}$ columns; each of them corresponds to a variable in $Y_{C_j}$. Each row in $A$ corresponds to a constraint in the system of Eqs. defined by Eqs. (9), (10) and (11). So we use notations $\widehat{f}(C_1), \ldots \widehat{f}(C_m)$ to represents the constraints defined by Eq. (9). Similarly we use notations $\{s(C,i) \mid C \in \mathcal{C}, i \in [0, c-1]\}$ and $\{t(C,i) \mid C \in \mathcal{C}, i \in [0, c-1]\}$ to represents the constraints defined by Eqs. (10) and (11), respectively.
*Matrix* $B_1$. The matrix $B_1$ is of dimension $(2c+1) \times (c \cdot 2^{\ell+1})$. In the first row of $B_1$, we have coefficients of $Y_{C_1}$ from $\widehat{f}(C_1)$. For $j \in [c]$, the rows indexed by $j+1$ and $c+j+1$ are defined as follows. In the $(j+1)^{st}$ row of $B_1$, we have coefficients of $Y_{C_1}$ from $s(C_1, j)$ while in the $(c+j+1)^{st}$ row of $B_1$, we have coefficients of $Y_{C_1}$ from $t(C_1, j)$. That is the entries of $B_1$ are as follows, where $i \in [0, c-1]$ and $a \in [0, L-1]$.

$$B_1[1, i \cdot 2^{\ell+1} + 2a + 1] = \begin{cases} 1 \text{ if } \phi_a(X_i) \text{ satisfies } C_1, \\ 0 \text{ otherwise.} \end{cases} \tag{15}$$

$$B_1[1, i \cdot 2^{\ell+1} + 2a + 2] = 0, \text{ and} \tag{16}$$

$$B_1[2+i, i \cdot 2^{\ell+1} + 2a + 1] = B_1[2+i, i \cdot 2^{\ell+1} + 2a + 2] = 1, \tag{17}$$

$$B_1[c+2+i, i \cdot 2^{\ell+1} + 2a + 1] = B_1[c+2+i, i \cdot 2^{\ell+1} + 2a + 2] = a, \tag{18}$$

Here, Eqs. (15) and (16), follow from Eq. (9). Eqs. (17) and (18) follow from Eqs. (10) and (11), respectively. All other entries in $B_1$ are zeros. That is, for all $i, i' \in [0, c-1]$ and $g \in [2^{\ell+1}]$ such that $i \neq i'$,

$$B_1[2+i, i' \cdot 2^{\ell+1} + g] = B_1[c+2+i, i' \cdot 2^{\ell+1} + g] = 0, \tag{19}$$

This completes the definition of $B_1$. By its role in the reduction, the matrix $B_1$ is partitioned into three parts. The first row is called the *evaluation part* of $B_1$. The part composed of rows indexed by $2, 3, \ldots, c+1$ is called the *selection part* and the part composed of the last $c$ rows is called the *successor matching part* (See Fig. 6c).

*Matrices* $B_r$ *for* $1 < r < m$. The matrix $B_r$ is of dimension $(3c+1) \times (c \cdot 2^{\ell+1})$. The first $c$ rows are defined by Eq. (11). For $j \in [c]$, in $i^{th}$ row, we have coefficients of $Y_{C_r}$ from $t(C_{r-1}, i)$. In the $(c+1)^{st}$ row of $B_r$, we have coefficients of $Y_{C_r}$ from $\widehat{f}(C_r)$. For $i \in [c]$, the rows indexed by $c+1+i$ and $2c+1+i$ are defined as follows. In the $(c+1+i)^{th}$ row of $B_r$, we have coefficients of $Y_{C_r}$ from $s(C_r, i)$ while in the $(2c+1+i)^{th}$ row of $B_r$, we have coefficients of $Y_{C_r}$ from $t(C_r, i)$. This completes the definition of $B_r$. By its role in the reduction, the matrix $B_r$ is partitioned in to four parts. The part composed of the first $c$ rows is called the *predecessor matching part*. The part composed of the row indexed by $c+1$ is called the *evaluation part* of $B_1$. The part composed of rows indexed by $c+2, c+3, \ldots, 2c+1$ is called the *selection part* and the part composed of the last $c$ rows is called the *successor matching part* (For illustration see Fig. 6b). That is the entries of $B_1$ are as follows, where $i \in [0, c-1]$ and $a \in [0, L-1]$.

The predecessor matching part is defined by

$$B_r[i+1, i \cdot 2^{\ell+1} + 2a + 1] = B_r[i+1, i \cdot 2^{\ell+1} + 2a + 2] = L - 1 - a. \quad (20)$$

The evaluation part is defined by

$$B_r[c+1, i \cdot 2^{\ell+1} + 2a + 2] = 0, \quad (21)$$

and

$$B_r[c+1, i \cdot 2^{\ell+1} + 2a + 1] = \begin{cases} 1, & \text{if } \phi_a(X_i) \text{ satisfies } C_r, \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

The selection part for $B_r$ is defined as

$$B_r[c+2+i, i \cdot 2^{\ell+1} + 2a + 1] = B_r[c+2+i, i \cdot 2^{\ell+1} + 2a + 2] = 1, \quad (23)$$

The successor matching part for $B_r$ is defined as

$$B_r[2c+2+i, i \cdot 2^{\ell+1} + 2a + 1] = B_r[2c+2+i, i \cdot 2^{\ell+1} + 2a + 2] = j, \quad (24)$$

All other entries in $B_r$, which are not listed above, are zero. That is, for all $i, i' \in [0, c-1]$ and $g \in [2^{\ell+1}]$ such that $i \neq i'$,

$$B_r[i+1, i' \cdot 2^{\ell+1} + g] = 0, \quad (25)$$
$$B_r[c+2+i, i' \cdot 2^{\ell+1} + g] = 0, \text{ and} \quad (26)$$
$$B_r[2c+2+i, i' \cdot 2^{\ell+1} + g] = 0. \quad (27)$$

For an example, see Fig. 7.

*Matrices $B_m$*. The matrix $B_m$ is of dimension $(2c+1) \times (c \cdot 2^{\ell+1})$. For $j \in [c]$, in $i^{the}$ row, we have coefficients of $Y_{\widehat{C}_m}$ from $t(C_{m-1}, i)$. In the $(c+1)^{st}$ row of $B_r$, we have coefficients of $Y_{C_m}$ from $\widehat{f}(C_m)$. In the $(c+1+i)^{th}$ row of $B_m$, we have coefficients of $Y_r$ from $s(C_m, i)$. That is, $B_m$ is obtained by deleting the successor

$$\begin{bmatrix}
0&0&1&1&2&2&3&3&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&2&2&3&3\\
1&0&0&0&1&0&1&0&0&0&1&0&0&0&0&1\\
1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
3&3&2&2&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&3&3&2&2&1&1&0&0
\end{bmatrix}
\qquad
\begin{bmatrix}
0&0&1&1&2&2&3&3&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&2&2&3&3\\
1&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1
\end{bmatrix}$$

**Fig. 7** Let $n = 4, c = 2, \ell = 2$ and $C_r = x_1 \vee \overline{x_2} \vee x_4$. The assignments are $\phi_0(X_0) = \{x_1 = x_2 = 0\}, \phi_1(X_0) = \{x_1 = 0, x_2 = 1\}, \phi_2(X_0) = \{x_1 = 1, x_2 = 0\}, \phi_3(X_0) = \{x_1 = x_2 = 1\}, \phi_0(X_1) = \{x_3 = x_4 = 0\}, \phi_1(X_1) = \{x_3 = 0, x_4 = 1\}, \phi_2(X_1) = \{x_3 = 1, x_4 = 0\}, \phi_3(X_1) = \{x_3 = x_4 = 1\}$. The entries defined according to $\phi_1(X_0)$ and $\phi_3(X_1)$ are colored red and blue respectively. If $1 < r < m$, then the matrix on the left represents $B_r$ and if $r = 1$, then $B_r$ can be obtained by deleting the yellow colored portion from the top matrix. The matrix on the right represents $B_m$

matching part from the construction of $B_r$ above. The entries of $B_m$ are as follows, where $i \in [0, c-1]$ and $a \in [0, L-1]$.

$$\begin{aligned}
&B_m[i+1, i \cdot 2^{\ell+1} + 2a + 1] = B_m[i+1, i \cdot 2^{\ell+1} + 2a + 2] = L - 1 - a, \\
&B_m[c+1, i \cdot 2^{\ell+1} + 2a + 2] = 0, \text{ and} \\
&B_m[c+1, i \cdot 2^{\ell+1} + 2a + 2] = \begin{cases} 1, & \text{if } \phi_a(X_i) \text{ satisfies } C_m, \\ 0, & \text{otherwise.} \end{cases} \\
&B_m[c+2+i, i \cdot 2^{\ell+1} + 2a + 1] = B_m[c+2+i, i \cdot 2^{\ell+1} + 2a + 2] = 1,
\end{aligned} \tag{28}$$

All other entries in $B_m$ are zeros. That is, for all $i, i' \in [0, c-1]$ and $g \in [2^{\ell+1}]$ such that $i \neq i'$,

$$B_m[1+i, i' \cdot 2^{\ell+1} + g] = 0 \tag{29}$$

$$B_m[c+2+i, i' \cdot 2^{\ell+1} + g] = 0, \tag{30}$$

$$B_m[2c+2+i, i' \cdot 2^{\ell+1} + g] = 0. \tag{31}$$

*Matrix A*.Now we explain how the matrix $A$ is formed from $B_1, \ldots, B_m$. The matrices $B_1, \ldots, B_m$ are disjoint submatrices of $A$ and they cover all non zero entries of $A$. Informally, the submatrices $B_1, \ldots, B_m$ form a chain such that the rows corresponding to the successor matching part of $B_r$ will be the same as the rows in the predecessor matching part of $B_{r+1}$ (because of Eq. (11). A pictorial representation of $A$ can be found in Fig. 5b. Formally, let $I_1 = [2c+1]$ and $I_m = [(m-1)(2c+1) + (c+1)] \setminus [(m-1)(2c+1) - c]$. For every $1 < r < m$, let $I_r = [r(2c+1)] \setminus [(r-1)(2c+1) - c]$, and for $r \in [m]$, let $J_r = [r \cdot c \cdot 2^{\ell+1}] \setminus [(r-1) \cdot c \cdot 2^{\ell+1}]$. Now for each $r \in [m]$, the matrix $A[I_r, J_r] := B_r$. All other entries of $A$ not belonging to any of the submatrices $A[I_r, J_r]$ are zero.

Towards upper bounding the path-width of $A$, we start with some notation. We partition the set of columns of $A$ into $m$ parts $J_1, \ldots, J_m$ (we have already defined these sets) with one part per clause. For each $r \in [m]$, $J_r$ is the set of columns associated with $Y_{C_r}$. We further divide $J_r$ into $c$ equal parts, one per variable set $Y_{C_r,i}$.

These parts are

$$\begin{aligned}P_{r,i} &= \{(r-1)c\cdot 2^{\ell+1}+i\cdot 2^{\ell+1}+1,\dots,(r-1)c\cdot 2^{\ell+1}\\ &\quad +(i+1)\cdot 2^{\ell+1}\},\ i\in[0,c-1].\end{aligned}$$

In other words, $P_{r,i}$ is the set of columns corresponding to $Y_{C_r,i}$ and $|P_{r,i}|=2^{\ell+1}$. We also put $n'=m\cdot c\cdot 2^{\ell+1}$ to be the number of columns in $A$.

**Lemma 7** *The path-width of the column matroid of $A$ is at most $c+4$*

***Proof*** Recall that $n'=m\cdot c\cdot 2^{\ell+1}$ is the number of columns in $A$ and $m'$ be the number of rows in $A$. To prove that the path-width of $A$ is at most $c+4$, it is sufficient to show that for all $j\in[n'-1]$,

$$\dim\langle \operatorname{span}(A|\{1,\dots,j\})\cap \operatorname{span}(A|\{j+1,\dots,n'\})\rangle \le c+3. \tag{32}$$

The idea for proving Eq. (32) is based on the following observation. For $V'=A|\{1,\dots,j\}$ and $V''=A|\{j+1,\dots,n'\}$, let

$$I=\{q\in[m']\mid \text{there exists } v'\in V' \text{ and } v''\in V'' \text{ such that } v'[q]\neq v''[q]\neq 0\}.$$

Then the dimension of $\operatorname{span}(V')\cap\operatorname{span}(V'')$ is at most $|I|$. Thus to prove (32), for each $j\in[n'-1]$, we construct the corresponding set $I$ and show that its cardinality is at most $c+3$.

We proceed with the details. Let $v_1,v_2,\dots,v_{n'}$ be the column vectors of $A$. Let $j\in[n'-1]$. Let $V_1=\{v_1,\dots,v_j\}$ and $V_2=\{v_{j+1},\dots,v_{n'}\}$. We need to show that $\dim\langle\operatorname{span}(V_1)\cap\operatorname{span}(V_2)\rangle\le c+3$. Let

$$I'=\{q\in[m']\mid \text{there exists } v\in V_1 \text{ and } v'\in V_2 \text{ such that } v[q]\neq 0\neq v'[q]\}.$$

We know that $[n']$ is partitioned into parts $P_{r',i'}$, $r'\in[m]$, $i'\in[0,c-1]$.

**Fix** $r\in[m]$ **and** $i\in[0,c-1]$ **such that** $j\in P_{r,i}$.

Let $j=(r-1)c\cdot 2^{\ell+1}+i\cdot 2^{\ell+1}+g$, where $g\in[2^{\ell+1}]$. Let $q_1=\max\{0,(r-1)(2c+1)-c\}$, $q_2=r(2c+1)$, $j_1=(r-1)\cdot c\cdot 2^{\ell+1}$, and $j_2=r\cdot c\cdot 2^{\ell+1}$ Then $[q_2]\setminus[q_1]=I_r$ and $[j_2]\setminus[j_1]=J_r$ (recall the definition of sets $I_r$ and $J_r$).

By the decomposition of matrix $A$, for every $q>q_2$ and for every vector $v\in V_1$, we have $v[q]=0$. Also, for every $q\le q_1$ and for any $v\in V_2$, we have that $v[q]=0$. This implies that $I'\subseteq[q_2]\setminus[q_1]=I_r$. Now we partition $I_r$ into 4 parts: $R_1$, $R$, $S$, and $R_2$, These parts are defined as follows.

$$\begin{aligned}R_1 &= \begin{cases}\emptyset, & \text{if } r=1,\\ \{(r-2)(2c+1)+i'\mid i'\in[0,c-1]\}, & \text{otherwise,}\end{cases}\\ R &= \{(r-1)(2c+1)+1\},\\ S &= \{(r-1)(2c+1)+2+i'\mid i'\in[0,c-1]]\}\end{aligned}$$

$$R_2 = \begin{cases} \emptyset, & \text{if } r = m, \\ \{(r-1)(2c+1)+c+2+i' \mid i' \in [0, c-1]\}, & \text{otherwise} \end{cases} \tag{33}$$

***Claim*** For each $r' \in [m]$, $q \notin I_{r'}$ and $j'' \in J_{r'}$, $v_{j''}[q] = 0$.

***Proof*** The non-zero entries in $A$ are covered by the disjoint sub-matrices $A[I_{r'}, J_{r'}] = B_{r'}$, $r' \in [m]$. Hence the claim follows. □

***Claim*** $|I' \cap R_1| \leq c - (i-1)$.

***Proof*** When $r = 1$, $R_1 = \emptyset$ and the claim trivially follows. Let $r > 1$, and let $q \in R_1$ be such that $q < (r-2)(2c+1)+i$. Then $q = (r-2)(2c+1)+1+i'$ for some $0 \leq i' < i$. Notice that $q \notin I_{r'}$ for every $r' > r$. By Claim 4.2, for every $v \in \bigcup_{r'>r} J_{r'}$, $v[q] = 0$. Now consider the vector $v_{j''} \in V_2 \setminus (\bigcup_{r'>r} J'_r)$. Notice that $j'' > j$ and $j'' \in J_r$. Let $j'' = j + a = (r-1)c \cdot 2^{\ell+1} + i \cdot 2^{\ell+1} + g + a$ for some $a \in [rc2^{\ell+1} - j]$. From the decomposition of $A$, $v_{j''}[q] = B_r[i'+1, i \cdot 2^{\ell+1} + g + a] = 0$, by (25). Thus for every $q \in R$, $q < (r-2)(2c+1)+i$ and $v \in V_2$, $v[q] = 0$.

This implies that

$$|I' \cap R_1| \leq |\{q \geq (r-2)(2c+1)+i\} \cap R_1| \leq c - (i-1).$$

□

***Claim*** $|I' \cap R_2| \leq i$.

***Proof*** When $r = m$, $R_2 = \emptyset$ and the claim trivially holds. So, now let $r < m$ and consider any $q \in R_2 \cap \{q' > (r-1)(2c+1)+c+2+i\}$. Let $i' > i$ such that $q = (r-1)(2c+1)+c+2+i'$. Notice that $q \notin I_{r'}$ for any $r' < r$. Hence, by Claim 4.2, for any $v \in \bigcup_{r'<r} J_{r'}$, $v[q] = 0$. Now consider any vector $v_{j''} \in V_1 \setminus (\bigcup_{r'<r} J'_r)$. Notice that $j'' \leq j$ and $j'' \in J_r$. Let $j'' = (r-1)c \cdot 2^{\ell+1} + i'' \cdot 2^{\ell+1} + a$ for some $a \in [2^{\ell+1}]$ and $i'' \leq i < i'$. From the decomposition of $A$, $v_{j''}[q] = B_r[2c+2+i', i'' \cdot 2^{\ell+1} + a] = 0$, by (27). Hence we have shown that for any $q \in R$, $q > (r-2)(2c+1)+c+2+i$ and $v \in V_1$, $v[q] = 0$. This implies that

$$|I' \cap R_2| \leq |\{q \leq (r-1)(2c+1)+c+2+i\} \cap R_1| \leq i.$$

□

***Claim*** $|I' \cap S| \leq 1$.

***Proof*** Consider any $q \in S$. Let $i' \in [0, c-1]$ such that $q = (r-1)(2c+1)+2+i'$. Notice that $q \notin I_{r'}$ for any $r' < r$, and hence, by Claim 4.2, for any $v \in \bigcup_{r'<r} J_{r'}$, $v[q] = 0$. Also notice that $q \notin I_{r'}$ for any $r' > r$, and hence, by Claim 4.2, for any $v \in \bigcup_{r'>r+1} J_{r'}$, $v[q] = 0$. So the only potential $j''$ for which $v_{j''}[q] \neq 0$, are from $J_r$.

We claim that if $q \in I' \cap S$, then $q = (r-1)(2c+1)+2+i$. Suppose $q \in I' \cap S$ and $q < (r-1)(2c+1)+2+i$. Let $q = (r-1)(2c+1)+2+i'$, where $0 \leq i' < i$. Then by the decomposition of $A$, for any $j'' > j$, $v_{j''}[q] = B_r[c+2+i', j'' - (r-1)c2^{\ell+1}] =$

$B_r[c+2+i', i_1 2^{\ell+1}+a]$, where $c-1 \geq i_1 \geq i$ and $a \in [2^{\ell+1}]$. Thus by (26), $v_{j''}[q] = B_r[c+2+i', i_1 2^{\ell+1}+a] = 0$. This contradicts the assumption that $q \in I' \cap S$.

Suppose $q \in I' \cap S$ and $q > (r-1)(2c+1)+c+2+i$. Let $q = (r-1)(2c+1)+c+2+i'$, where $i < i' < c$. Then by the decomposition of $A$, for any $j'' \leq j$, $v_{j''}[q] = B_r[c+2+i', j''-(r-1)c2^{\ell+1}] = B_r[c+2+i', i_1 2^{\ell+1}+a]$, where $0 \leq i_1 \leq i$, $a \in [2^{\ell+1}]$. Thus by (26), $v_{j''}[q] = B_r[c+2+i', i_1 2^{\ell+1}+a] = 0$. This contradicts the assumption that $i \in I' \cap S$. This implies that $|I' \cap S| \leq 1$. This completes the proof of the claim. □

Therefore, we have

$$\begin{aligned}
|I'| &= |I' \cap I_r| && \text{(Because } I' \subseteq I_r\text{)}\\
&= |I' \cap R_1| + |I' \cap R| + |I' \cap S| + |I' \cap R_2| && \text{(By (33))}\\
&\leq c-(i-1)+1+1+i && \text{(By Claims 4.2, 4.2 and 4.2)}\\
&= c+3
\end{aligned}$$

This completes the proof of the lemma. □

***Proof of Theorem 4.*** We prove the theorem by assuming a fast algorithm for (IPF) and use it to give a fast algorithm for CNF- SAT, refuting SETH. Let $\psi$ be an instance of CNF- SAT with $n_1$ variables and $m_1$ clauses. We choose a sufficiently large constant $c$ such that $(1-\epsilon)+\frac{4(1-\epsilon)}{c}+\frac{a}{c} < 1$ holds. We use the reduction mentioned in Lemma 5 and construct an instance $A_{(\psi,c)}x = b_{(\psi,c)}$, $x \geq 0$, of (IPF) which has a solution if and only if $\psi$ is satisfiable. The reduction takes time $\mathcal{O}(m_1^2 2^{\frac{n_1}{c}})$. Let $\ell = \lceil \frac{n_1}{c} \rceil$. The constraint matrix $A_{(\psi,c)}$ has dimension $((m_1-1)(2c+1)+1+c) \times (m_1 \cdot c \cdot 2^{\ell+1})$ and the largest entry in vector $b_{(\psi,c)}$ does not exceed $2^\ell - 1$. The path-width of $M(A_{(\psi,c)})$ is at most $c+4$.

Assuming that any instance of (IPF) with a non-negative constraint matrix of path-width $k$ is solvable in time $f(k)(\|b\|_\infty+1)^{(1-\epsilon)k}(mn)^a$, where $d$ is the maximum value in an entry of $b$ and $\epsilon$, $a > 0$ are constants, we have that $A_{(\psi,c)}x = b_{(\psi,c)}$, $x \geq 0$, is solvable in time

$$\begin{aligned}
f(c+4) \cdot 2^{\ell \cdot (1-\epsilon)(c+4)} \cdot 2^{\ell \cdot a} \cdot m_1^{\mathcal{O}(1)} &= 2^{\frac{n_1}{c}(1-\epsilon)(c+4)} \cdot 2^{\frac{n_1 \cdot a}{c}} \cdot m_1^{\mathcal{O}(1)}\\
&= 2^{n_1\left((1-\epsilon)+\frac{4(1-\epsilon)}{c}+\frac{a}{c}\right)} \cdot m_1^{\mathcal{O}(1)}.
\end{aligned}$$

Here the constant $f(c+4)$ is subsumed by the term $m_1^{\mathcal{O}(1)}$. Hence the total running time for testing whether $\psi$ is satisfiable or not, is,

$$\mathcal{O}(m_1^2 2^{\frac{n_1}{c}}) + 2^{n_1\left((1-\epsilon)+\frac{4(1-\epsilon)}{c}+\frac{a}{c}\right)} m_1^{\mathcal{O}(1)} = 2^{n_1\left((1-\epsilon)+\frac{4(1-\epsilon)}{c}+\frac{a}{c}\right)} m_1^{\mathcal{O}(1)} = 2^{\epsilon' \cdot n_1} m_1^{\mathcal{O}(1)},$$

where $\epsilon' = (1-\epsilon)+\frac{4(1-\epsilon)}{c}+\frac{a}{c} < 1$. This completes the proof of Theorem 4. □

## 4.3 Proof of Theorem 5

In this section we prove Theorem 5: (IPF) with non-negative matrix $A$ cannot be solved in time $f(\|b\|_\infty)(\|b\|_\infty + 1)^{(1-\epsilon)k}(mn)^{\mathcal{O}(1)}$ for any function $f$ and $\epsilon > 0$, unless SETH fails, where $k$ is the path-width of the column matroid of $A$. Here, we do not give a complete proof, but we give an adaptation of the proof of Theorem 4.

In Sect. 3.2, we gave a reduction from CNF- SAT to (IPF). However in this reduction the values in the constraint matrix $A_{(\psi,c)}$ and target vector $b_{(\psi,c)}$ can be as large as $2^{\lceil \frac{n}{c} \rceil} - 1$, where $n$ is the number of variables in the CNF-formula $\psi$ and $c$ is a constant. Let $m$ be the number of clauses in $\psi$. In this section we briefly explain how to get rid of these large values, at the cost of making large, but still bounded path-width. From a CNF-formula $\psi$, we construct a matrix $A = A_{(\psi,c)}$ as described in Sect. 3.2. The only rows in $A$ which contain values strictly greater than 1 (values other than 0 or 1) are the ones corresponding to the constraints defined by Eq. (11).

In other words, the values greater than 1 are in the rows in yellow/green colored portion in Fig. 5b. Recall that $\ell = \lceil \frac{n}{c} \rceil$ and the largest value in $A$ is $2^\ell - 1$. Any number less than or equal to $2^\ell - 1$ can be represented by a binary string of length $\ell = \frac{n}{c}$. Now we rewrite the Eq. (11), by $\ell$ new Equation. For each $j \in [\ell]$ and $N \in \mathbb{N}$, let $b_j(N)$ represent the $j^{th}$ bit in the $\ell$-bit binary representation of $N$. Then for all $r \in [m-1]$, $i \in [0, c-1]$ and $j \in [\ell]$, we have a system of constraints

$$\sum_{a \in [0,2L-1]} \left(b_j\left(\lfloor \frac{a}{2} \rfloor\right) \cdot y_{C_r,i,a}\right) + \left(b_j(L - 1 - \lfloor \frac{a}{2} \rfloor) \cdot y_{C_{r+1},i,a}\right) = 1 \qquad (34)$$

In other words, let $P = \{(r-1)(2c+1)+c+1+i \mid r \in [m-2], i \in [0, c-1]\}$. The rows of $A$ containing values larger than one are indexed by $P$. Now we construct a new matrix $A'$ from $A$ by replacing each row of $A$ whose index is in the set $P$ with $\ell$ rows and for any value $A[i, j]$, $i \in P$ we write its $\ell$-bit binary representation in the column corresponding to $j$ and the newly added $\ell$ rows of $A'$. That is, for any $\gamma \in P$, we replace the row $\gamma$ with $\ell$ rows, $\gamma_1, \ldots, \gamma_\ell$. Where, for any $j$, if the value $A[\gamma, j] = W$ then $A'[\gamma_k, j] = \eta_k$, where $\eta_k$ is the $k^{th}$ bit in the $\ell$-sized binary representation of $W$.

Let $m'$ be the number of rows in $A'$. Now the target vector $b'$ is defined as $b'[i] = 1$ for all $i \in [m']$. This completes the construction of the reduced (IPF) instance $A'x = b'$. The correctness proof of this reduction is using arguments similar to those used for the correctness of Lemma 6.

**Lemma 8** *The path-width of the column matroid of $A'$ is at most $(c+1)\frac{n}{c} + 3$.*

***Proof*** We sketch the proof, which is similar to the proof of Lemma 7. We define $I'_r$ and $J'_r$ for any $r \in [m]$ like $I_r$ and $J_r$ in Sect. 3.2. In fact, the rows in $I'_r$ are the rows obtained from $I_r$ in the process explained above to construct $A'$ from $A$. We need to show that $\dim\langle \text{span}(A'|\{1, \ldots, j\}) \cap \text{span}(A'|\{j+1, \ldots, n'\})\rangle \leq (c+1)\frac{n}{c} + 2$ for all $j \in [n'-1]$, where $n'$ is the number of columns in $A'$. The proof proceeds by bounding the number of indices $I$ such that for any $q \in I$ there exist vectors $v \in A'|\{1, \ldots, j\}$ and $u \in A'|\{j+1, \ldots, n'\}$ with $v[q] \neq 0 \neq u[q]$. By arguments similar to the ones used in the proof of Lemma 7, we can show that for any $j \in [n'-1]$, the corresponding set $I'$ of indices is a subset of $I'_r$ for some $r \in [m]$. Recall the partition of $I_r$ into $R_1$, $R$, $S$ and $R_2$ in Lemma 7. We partition $I'_r$ into parts $Q_1$, $W$, $U$ and $Q_2$. Notice that $R_1, R_2 \subseteq P$, where $P$ is the set of rows which covers all values strictly greater than 1. The set $Q_1$ and $Q_2$ are obtained from $R_1$ and $R_2$, respectively, by the process mentioned above to construct $A'$ from $A$. That is, each row in $R_i$, $i \in \{1, 2\}$ is replaced by $\ell$ rows in $Q_i$. Rows in $W$ corresponds to rows in $R$ and $U$ corresponds to the rows in $W$. This allows us to bound the following terms for some $i \in [0, c-1]$:

$$\begin{aligned}|I' \cap Q_1| &\leq (c-(i-1))\ell = (c-(i-1))\ell,\\ |I' \cap Q_2| &\leq i \cdot \ell,\\ |I' \cap U| &\leq 1, \text{ and}\\ |I' \cap W| &\leq 1.\end{aligned}$$

By using the fact that $I' \subseteq I'_r$ and the above system of inequalities, we can show that

$$\dim\langle \operatorname{span}(A'|\{1,\ldots,j\}) \cap \operatorname{span}(A'|\{j+1,\ldots,n'\})\rangle \leq (c+1)\lceil \frac{n}{c} \rceil + 2.$$

This completes the proof sketch of the lemma. □

Now the proof of the theorem follows from Lemma 8 and the correctness of the reduction (it is similar to the arguments in the proof of Theorem 4).

## 5 Proof of Theorem 6

In this section, we sketch how the proof of Cunningham and Geelen [1] of Theorem 3, can be adapted to prove Theorem 6. Recall that a path decomposition of width $k$ can be obtained in $f(k) \cdot n^{\mathcal{O}(1)}$ time for some function $f$ by making use of the algorithm by Jeong et al. [13]. However, we do not know if such a path decomposition can be constructed in time $\mathcal{O}((\|b\|_\infty + 1)^{k+1})n^{\mathcal{O}(1)}$, so the assumption that a path decomposition is given is essential.

Roughly speaking, the only difference in the proof is that when parameterized by the branch-width, the most time-consuming operation is the "merge" operation, when we have to construct a new set of partial solutions with at most $(\|b\|_\infty + 1)^k$ vectors from two already computed sets of sizes $(\|b\|_\infty + 1)^k$ each. Thus to construct a new set of vectors, one has to go through all possible pairs of vectors from both sets, which takes time roughly $(\|b\|_\infty + 1)^{2k}$. For path-width parameterization, the new partial solution set is constructed from two sets, but this time one set contains at most $(\|b\|_\infty + 1)^k$ vectors while the second contains at most $\|b\|_\infty + 1$ vectors. This allows us to construct the new set in time roughly $(\|b\|_\infty + 1)^{k+1}$.

Recall that for $X \subseteq [n]$, we define $S(A, X) = \operatorname{span}(A|X) \cap \operatorname{span}(A|E \setminus X)$, where $E = [n]$. The key lemma in the proof of Theorem 3 is the following.

**Lemma 9** ([1]) *Let $A \in \{0, 1, \ldots, \|b\|_\infty\}^{m \times n}$ and $X \subseteq [n]$ such that $\lambda_{M(A)}(X) = k$. Then the number of vectors in $S(A, X) \cap \{0, \ldots, \|b\|_\infty\}^m$ is at most $(\|b\|_\infty + 1)^{k-1}$.*

To prove Theorem 6, without loss of generality, we assume that the columns of $A$ are ordered in such a way that for every $j \in [n-1]$,

$$\dim\langle \operatorname{span}(A|\{1,\dots,i\}) \cap \operatorname{span}(A|\{i+1,\dots,n\})\rangle \leq k-1.$$

Let $A' = [A, b]$. That is $A'$ is obtained by appending the column-vector $b$ to the end of $A$. Then for each $i \in [n]$,

$$\dim\langle \operatorname{span}(A'|\{1,\dots,i\}) \cap \operatorname{span}(A'|\{i+1,\dots,n+1\})\rangle \leq k. \tag{35}$$

Now we use dynamic programming to check whether the following conditions are satisfied. For $X \subseteq [n+1]$, let $\mathcal{B}(X)$ be the set of all vectors $b' \in \mathbb{Z}^m_{\geq 0}$ such that

(1) $0 \leq b' \leq b$,
(2) there exists $z \in \mathbb{Z}^{|X|}_{\geq 0}$ such that $(A'|X)z = b'$, and
(3) $b' \in S(A', X)$.

Then (IPF) has a solution if and only if $b \in \mathcal{B}([n])$. Initially the algorithm computes for all $i \in [n]$, $\mathcal{B}(\{i\})$ and by Lemma 9, we have that $|\mathcal{B}(\{i\})| \leq \|b\|_\infty + 1$. In fact $\mathcal{B}(\{i\}) \subseteq \{a \cdot v \mid v \text{ is the } i^{th} \text{ column vector of } A' \text{ and } a \in [\|b\|_\infty + 1]\}$. Then for each $j \in \{2,\dots,n\}$ the algorithm computes $\mathcal{B}([j])$ in increasing order of $j$ and outputs YES if and only if $b \in \mathcal{B}([n])$. That is, $\mathcal{B}([j])$ is computed from the already computed sets $\mathcal{B}([j-1])$ and $\mathcal{B}(\{j\})$. Notice that $b' \in \mathcal{B}([j])$ if and only if

(a) there exist $b_1 \in \mathcal{B}(\{1,\dots,j-1\})$ and $b_2 \in \mathcal{B}(\{j\})$ such that $b' = b_1 + b_2$,
(b) $b' \leq b$ and
(c) $b' \in S(A', [j])$.

So the algorithm enumerates vectors $b'$ satisfying condition $(a)$, and each such vector $b'$ is included in $\mathcal{B}([j])$, if $b'$ satisfy conditions $(b)$ and $(c)$. Since by (35) and Lemma 9, $|\mathcal{B}([j-1])| \leq (\|b\|_\infty + 1)^k$ and $|\mathcal{B}(\{j\})| \leq \|b\|_\infty + 1$, the number of vectors satisfying condition $(a)$ is $(\|b\|_\infty + 1)^k$, and hence the exponential factor of the required running time follows. This provides the bound on the claimed exponential dependence in the running time of the algorithm. The bound on the polynomial component of the running time follows from exactly the same arguments as in [1].

## 6 Conclusion

We would like to mention that our proofs of Theorems 4 and 5 imply lowerbounds in terms of the dual path-width of the constraint matrix $A$. The dual graph $G$ of a matrix $A$ is defined as follows. For each row $i$ of $A$ there is a vertex $v_i$ in $G$. There is an edge between $v_i$ and $v_j$ if and only if the corresponding rows overlap (i.e., there is an index $r$ such that $A[i,r] \neq 0$ and $A[j,r] \neq 0$). The dual pathwidth of $A$ is the path-width of the graph $G$. We observe that the proofs of Theorems 4 and 5 imply the following results.

– Unless SETH fails, (IPF) with even a non-negative $m \times n$ constraint matrix $A$ cannot be solved in time $f(k)(\|b\|_\infty + 1)^{(1-\epsilon)k}(mn)^{\mathcal{O}(1)}$ or $f(\|b\|_\infty)(\|b\|_\infty +$

$1)^{(1-\epsilon)k}(mn)^{\mathcal{O}(1)}$ for any computable function $f$ and $\epsilon > 0$, where $k$ is the dual path-width of $A$.

Towards the proof of the above result we observe that the dual path-width of the matrix $A$ constructed in the proof of Lemma 5 is at most $c+2$. That is, we construct a path-decomposition of the dual graph of $A$ as follows. Notice that the rows $1, \ldots, 2c+1$ of $A$ cover all the non-zero entries of the submatrix $B_1$. Let $\sigma_1$ be the sequence $S_{1,1}, \ldots, S_{1,c}$ of subsets of $\{1, \ldots, 2c+1\}$, where $S_{1,1}, \ldots, S_{1,c}$ are defined as follows. For each $j \in [c]$,

$$S_{1,j} = \{1\} \cup \{1+j\} \cup \{\{c+2, \ldots, c+1+j\}$$

For each $1 < r < m$, let $p_r$ be the number such that the rows $p_r+1, p_r+2, \ldots, p_r+3c+1$ of $A$ cover all the non-zero entries of the sub-matrix $B_r$. Let $\sigma_r$ be the sequence $S_{r,1}, \ldots, S_{r,c}$ of subsets of $\{p_r+1, \ldots, p_r+3c+1\}$, where $S_{r,1}, \ldots, S_{r,c}$ are defined as follows. For each $j \in [c]$,

$$\begin{aligned} S_{r,j} = \{&p_r + j, \ldots, p_r + c\} \cup \{p_r + c + 1\} \cup \{p_r + c + 1 + j\} \\ &\cup \{\{p_r + 2c + 2, \ldots, p_r + 2c + 1 + j\} \end{aligned}$$

Let $p_m$ be the number such that the rows $p_m+1, p_m+2, \ldots, p_m+2c+1$ of $A$ cover all the non-zero entries of the sub-matrix $B_m$. Let $\sigma_m$ be the sequence $S_{m,1}, \ldots, S_{m,c}$ of subsets of $\{p_m+1, \ldots, p_m+2c+1\}$, where $S_{m,1}, \ldots, S_{m,c}$ are defined as follows. For each $j \in [c]$,

$$S_{m,j} = \{p_m + j, \ldots, p_m + c\} \cup \{p_m + c + 1\} \cup \{p_m + c + 1 + j\}$$

Then, $\sigma_1, \sigma_2, \ldots, \sigma_m$ is a path-decompostion of the dual graph of $A$ where each subset is of size at most $c+3$. Therefore, the dual path-width of $A$ is at most $c+2$. Similarly one can prove that the dual path-width of the matrix $A'$ constructed in Sect. 4.3 is at most $(c+1)\frac{n}{c}+1$. Then by following arguments similar to that of proofs of Theorems 4 and 5, one can prove the required result.

## References

1. Cunningham, W. H., and Geelen, J.: *On integer programming and the branch-width of the constraint matrix*, in Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization (IPCO), vol. 4513 of Lecture Notes in Comput. Sci., Springer, 2007, pp. 158–166
2. Cygan, M., Dell, H., Lokshtanov, D., Marx, D., Nederlof, J., Okamoto, Y., Paturi, R., Saurabh, S., and Wahlström, M.: *On problems as hard as CNF-SAT*, in Proceedings of the 27th IEEE Conference on Computational Complexity (CCC), IEEE, 2012, pp. 74–84
3. Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., and Saurabh, S.: *Parameterized Algorithms*, Springer, 2015
4. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J. M. M., and Wojtaszczyk, J. O.: *Solving connectivity problems parameterized by treewidth in single exponential time*, in Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2011, pp. 150–159

5. Dorn, F.: *Dynamic programming and fast matrix multiplication*, in Proceedings of the 14th Annual European Symposium on Algorithms (ESA), vol. 4168 of Lecture Notes in Comput. Sci., Springer, Berlin, 2006, pp. 280–291
6. Eisenbrand, F., and Weismantel, R.: *Proximity results and faster algorithms for integer programming using the steinitz lemma*, in Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2018, pp. 808–816
7. Gajentaan, A., Overmars, M.H.: On a class of $o(n^2)$ problems in computational geometry. Comput. Geom. **5**, 165–185 (1995)
8. Ganian, R., Ordyniak, S., and Ramanujan, M. S.: *Going beyond primal treewidth for (M)ILP*, in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, S. P. Singh and S. Markovitch, eds., AAAI Press, 2017, pp. 815–821
9. Horn, G.B., Kschischang, F.R.: On the intractability of permuting a block code to minimize trellis complexity. IEEE Trans. Inf. Theory **42**, 2042–2048 (1996)
10. Impagliazzo, R., Paturi, R.: On the complexity of $k$-SAT. J. Computer Syst. Sci. **62**, 367–375 (2001)
11. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity. J. Computer Syst. Sci. **63**, 512–530 (2001)
12. Jansen, K., and Rohwedder, L.: *On integer programming and convolution*, in 10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA, vol. 124 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019, pp. 43:1–43:17
13. Jeong, J., Kim, E. J., and Oum, S.: *Constructive algorithm for path-width of matroids*, in Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2016, pp. 1695–1704
14. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Op. Res. **12**, 415–440 (1987)
15. Knop, D., Pilipczuk, M., and Wrochna, M.: *Tight complexity lower bounds for integer linear programming with few constraints*, in 36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany, vol. 126 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019, pp. 44:1–44:15
16. Lenstra, H.W., Jr.: Integer programming with a fixed number of variables. Math. Op. Res. **8**, 538–548 (1983)
17. Lokshtanov, D., Marx, D., and Saurabh, S.: *Known algorithms on graphs on bounded treewidth are probably optimal*, in Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2011, pp. 777–789
18. Margulies, S., Ma, J., Hicks, I.V.: The Cunningham-Geelen method in practice: Branch-decompositions and integer programming. INFORMS J. Comput. **25**, 599–610 (2013)
19. Marx, D.: Can you beat treewidth?, Theory of. Computing **6**, 85–112 (2010)
20. Papadimitriou, C.H.: On the complexity of integer programming. J. ACM **28**, 765–768 (1981)
21. Robertson, N., Seymour, P.D.: Graph minors. X. obstructions to tree-decomposition. J. Combinatorial Theory Ser. B **52**, 153–190 (1991)
22. van Rooij, J. M. M., Bodlaender, H. L., and Rossmanith, P.: *Dynamic programming on tree decompositions using generalised fast subset convolution*, in Proceedings of the 17th Annual European Symposium on Algorithms (ESA), vol. 5757 of Lecture Notes in Comput. Sci., Springer, 2009, pp. 566–577