


Parameterized Algorithms for Matching Integer Programs with Additional Rows and Columns

Alexandra Lassota ✉ 

Eindhoven University of Technology, Netherlands

Koen Ligthart ✉ 

Eindhoven University of Technology, Netherlands

Abstract

We study integer linear programs (ILP) of the form $\min\{c^\top x \mid Ax = b, l \leq x \leq u, x \in \mathbb{Z}^n\}$ and analyze their parameterized complexity with respect to their distance to the generalized matching problem—following the well-established approach of capturing the hardness of a problem by the distance to triviality. The generalized matching problem is an ILP where each column of the constraint matrix has 1-norm of at most 2. It captures several well-known polynomial time solvable problems such as matching and flow problems. We parameterize by the size of variable and constraint backdoors, which measure the least number of columns or rows that must be deleted to obtain a generalized matching ILP. This extends generalized matching problems by allowing a parameterized number of additional arbitrary variables or constraints, yielding a novel parameter.

We present the following results: (i) a fixed-parameter tractable (FPT) algorithm for ILPs parameterized by the size p of a minimum variable backdoor to generalized matching; (ii) a randomized slice-wise polynomial (XP) time algorithm for ILPs parameterized by the size h of a minimum constraint backdoor to generalized matching as long as c and A are encoded in unary; (iii) we complement (ii) by proving that solving an ILP is $W[1]$ -hard when parameterized by h even when c, A, l, u have coefficients of constant size. To obtain (i), we prove a variant of lattice-convexity of the degree sequences of weighted b -matchings, which we study in the light of SBO jump M -convex functions. This allows us to model the matching part as a polyhedral constraint on the integer backdoor variables. The resulting ILP is solved in FPT time using an integer programming algorithm. For (ii), the randomized XP time algorithm is obtained by pseudo-polynomially reducing the problem to the exact matching problem. To prevent an exponential blowup in terms of the encoding length of b , we bound the Graver complexity of the constraint matrix and employ a Graver augmentation local search framework. The hardness result (iii) is obtained through a parameterized reduction from ILP with h constraints and coefficients encoded in unary.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Theory of computation → Complexity classes

Keywords and phrases integer programming, fixed-parameter tractability, polyhedral optimization, matchings

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

We study integer linear programs (ILPs) and analyze their parameterized complexity with respect to their distance to the generalized matching problem. In general, an ILP is of the form

$$\min\{c^\top x \mid Ax = b, l \leq x \leq u, x \in \mathbb{Z}^n\}, \quad (\text{G})$$

where $l \in (\mathbb{Z} \cup \{-\infty\})^n, u \in (\mathbb{Z} \cup \{\infty\})^n, c \in \mathbb{Z}^n$ and $A \in \mathbb{Z}^{m \times n}$. The underlying integer linear programming problem is to either decide that the system is infeasible, there exists an optimal feasible solution, or it is unbounded and provide a solution with an unbounded direction of improvement.



© Alexandra Lassota and Koen Ligthart;

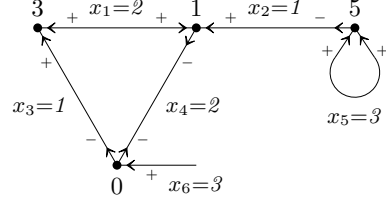
licensed under Creative Commons License CC-BY 4.0

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & -1 & -1 & 0 & 1 \end{pmatrix} x = \begin{pmatrix} 3 \\ 1 \\ 5 \\ 0 \end{pmatrix}$$

(a) Example constraints $Mx = b$ with $\|M\|_1 = 2$ of a generalized matching instance.



(b) Graphic interpretation of a solution to the system of Figure 1a.

■ **Figure 1** An example instance of the generalized matching problem and a corresponding solution, disregarding potential variable bounds.

Integer linear programming is a powerful language that has been applied to many key problems such as graph problems [20, 21], scheduling and bin packing [25, 31], multichoice optimization [19] and computational social choice [4, 34], among others. Unfortunately, solving ILPs is, in general, NP-hard.

However, most ILP formulations of problems are naturally well-structured, see e.g. [20, 25, 26, 31, 34] and the references therein. Motivated by this insight and their daunting general hardness, classes of ILPs with highly-structured constraint matrices and parameterizations have been intensively and successfully studied to obtain polynomial and FPT time algorithms [20, 25, 26, 31, 34]. Arguably most famous is Lenstra’s 1983 algorithm who presented the first FPT time algorithm for constraint matrices with few columns [33]. The body of literature for the over three-decades-long research and the many structures, parameters, and applications are too vast to cover here, we thus refer to [24] for an overview.

Central to this paper is the polynomial time solvable class of the generalized matching problem, which is the ILP restricted to coefficient matrices satisfying $\|A\|_1 \leq 2$, i.e., all columns have 1-norm at most 2.

► **Theorem 1** (Theorem 36.1 in [46]). *When $\|A\|_1 \leq 2$, ILP (G) can be solved in strongly polynomial time.*

The generalized matching problem captures a variety of well-known problems in P such as minimum cost flow, minimum cost (b -)matching and certain graph factor problems [46]. This connection becomes apparent by interpreting the variables as values assigned to edges of a graph whose vertices correspond to the constraints of the ILP¹. In this way, a constraint matrix A with $\|A\|_1 \leq 2$ can be interpreted as an incidence matrix of a generalized bidirected graph where all endpoints of edges are given signs and where degenerate self-loops and half-edges may be present [14]. An example of a graphic interpretation of an instance of the generalized matching problem is shown in Figure 1. For instance, a problem such as the minimum cost perfect matching problem on a graph $G = (V, E)$ can be cast as a generalized matching ILP. In this scenario, $A \in \{0, 1\}^{V \times E}$ is the incidence matrix of G and $l = \mathbf{0}, u = \mathbf{1}, b = \mathbf{1}$. That is, $A_{ij} = 1$ if and only if vertex i is incident to edge j . Here, $\mathbf{0}$ and $\mathbf{1}$ denote the all-zero and all-ones vector respectively.

This paper studies the degree to which we can extend the generalized matching problem while maintaining tractability. For this purpose, we study constraint matrices A which are similar to the constraint matrix of a generalized matching problem. Such matrix A

¹ Note that, due to the conventions in ILP, the vertices of the graph are the numbers $[m] := \{1, \dots, m\}$ and the edges are identified with the numbers $[n]$.

consists of a small corner block $C \in \mathbb{Z}^{h \times p}$, wide block $W \in \mathbb{Z}^{h \times n}$, tall block $T \in \mathbb{Z}^{m \times p}$ and matching-like block $M \in \mathbb{Z}^{m \times n}$ with $\|M\|_1 \leq 2$ and is of the form

$$A = \begin{pmatrix} C & W \\ T & M \end{pmatrix}.$$

We consider the cases where either of $h = 0$ or $p = 0$, yielding the ILPs

$$\min\{a^\top y + c^\top x \mid Ty + Mx = b, e \leq y \leq g, l \leq x \leq u, (y, x) \in \mathbb{Z}^{p+n}\}, \quad (\text{T})$$

which describes a generalized matching ILP admitting additional variables, and

$$\min\{c^\top x \mid Wx = d, Mx = b, l \leq x \leq u, x \in \mathbb{Z}^n\}, \quad (\text{W})$$

which describes a generalized matching ILP admitting additional constraints. The ILPs (T) and (W) represent generalizations of well-known flow and matching problems. See [2] for an application of bipartite matching with additional variables to a scheduling problem.

In the context of the fixed-parameter tractability of ILPs, graphs and corresponding structural parameters associated with the coefficient matrix of an ILP are usually studied. However, in our case, the coefficient matrix of the perfect matching problem may have arbitrary associated primal or dual graphs, unlike the case for two-stage stochastic and n -fold ILP, which have associated coefficient matrix graphs with limited tree-depth, see [17] for an overview. In addition, the incidence matrix of an undirected graph may have arbitrarily large subdeterminants, which makes it unsuited for methods such as the one discussed in [22].

The goal of this paper is to study the complexity of the above integer linear programs (T) and (W), i.e., ILPs where the constraint matrices are *nearly* generalized matching constraint matrices. For this purpose, we parameterize by the number of variables p and constraints h that must be deleted from ILP (G) to obtain a generalized matching problem. Such parameter choice follows the classical approach of studying parameters measuring the *distance to triviality*, also called (*deletion*) *backdoors to triviality*—a concept that was first proposed by Niedermeier [42]. Roughly speaking, this approach introduces a parameter that measures the distance of the given instance from an instance that is solvable in polynomial time. This approach was already used for many different problems such as clique, set cover, power dominating set, longest common subsequence, packing problems, and the satisfiability problem [42, 3, 25, 23].

For some problems, such as satisfiability, having obtained a variable backdoor immediately leads to an FPT algorithm parameterized by the size of the backdoor. For ILP, however, this is not as straightforward as variable domains may be arbitrarily large, which invalidates a brute-force approach to obtain FPT results in terms of variable backdoor size. In fact, solving ILPs in FPT time parameterized by the number of variables is already highly nontrivial [33, 45].

Despite such difficulties, a significant number of results have been obtained in the context of backdoors in ILPs. A large line of research has established efficient algorithms for solving ILPs given a variable or constraint backdoor to a remaining ILP which consists of many isolated ILPs with small coefficients [11, 13, 17]. Two-stage stochastic and n -fold ILP, which are two important special cases of such block-structured ILPs, may be viewed as ILPs of the form shown in respectively (T) and (W) where M is a block-diagonal matrix with small nonzero blocks and small coefficients. To relate these results with backdoors and backdoor identification, Dvořák et al. [13] define fracture numbers, which describe the number of variable and/or constraint removals needed to decouple the ILP into small blocks. They give

an overview of the parameterized complexity of ILPs parameterized by various backdoor sizes and coefficient sizes of the constraint matrix. Their work and the recent result of Cslovjecsek et al. [11] shows that the complexity of ILPs parameterized by backdoor size is subtle, as ILPs may already become NP-hard when there are only 3 complicating global variables connecting constant dimension, otherwise independent ILPs with unary coefficient size [13]. On the other hand, when additionally parameterizing by the coefficient size of the small blocks, the problem becomes FPT [11]. With this paper, we aim to reveal the parameterized complexity with respect to backdoor sizes to another class of well-known efficiently solvable ILPs, that of the generalized matching problem.

We show that ILPs parameterized by p , the number of variables that, upon deletion, give a general matching problem, is FPT by designing a corresponding algorithm. Further, we give a randomized, XP time algorithm with respect to h , and prove the corresponding W[1]-hardness result for this case.

We note that studying the complexity of ILP with respect to p is also motivated by the hardness of the general factor problem with a gap size of 2 [37]. Lovász’s reduction from edge coloring on cubic graphs shows that, even when T consists solely of columns with only one nonzero coefficient with value 3, ILP remains NP-hard. This, together with the tractability of generalized matching and our FPT result, shows that if one were to judge the complexity of an ILP by independently and individually labeling each column as “hard” or “easy”, the easy columns are precisely those with 1-norm at most 2.

Finally, given an arbitrary constraint matrix A of ILP (G), a permutation of the columns or rows to obtain the form ILP (T) or (W) with minimum p or h respectively can be obtained efficiently². That is, one can identify a minimum cardinality variable or constraint deletion backdoor to the generalized matching ILP. To obtain the form ILP (T), one may greedily move all columns with 1-norm greater than 2 to the left. The problem of minimizing h in ILP (W) is equivalent to the 3-uniform hitting set problem, which is NP-hard but FPT in h , cf. [23]. For this reason, we will assume that the given ILPs are of the form ILP (T) or (W) throughout the rest of the paper.

1.1 Contributions

In this paper, we show that solving integer linear programs is fixed-parameter tractable parameterized by the number of columns of A with 1-norm greater than 2.

► **Theorem 2.** *ILP (T) is FPT parameterized by p .*

This reveals an entirely new class of ILPs that is FPT and adds to the story of parameterized ILPs and distance to triviality paradigm. To obtain this algorithm, we use a remainder guessing strategy introduced by Cslovjecsek et al. [11]. Central in this approach is modeling non-backdoor variables as polyhedral constraints on the backdoor variables. In [11], these polyhedral constraints may be exponentially complex, whereas we exploit the structure of matching problems and employ an efficient description of a global polyhedral constraint. To accomplish this, we study the convexity of degree sequences for which a corresponding graph factor exists. These systems have been studied previously, see [1], and are a well-known example of discrete systems known as jump systems [7, 40, 41, 35]. For our application, we prove a new result involving the lattice-convexity of a particular class

² Through the use of elementary row operations, some systems may be modified to a row-equivalent system admitting smaller backdoors. See [8] for such study on block-structured ILP. We leave this problem open in the context of the parameterizations discussed in this paper.

of jump M-convex functions on the shifted lattice $2\mathbb{Z}^m + r$, see Section 2 for a technical overview of this algorithm.

As for the constraint backdoors, we show that matching-like ILPs are solvable in polynomial time with a randomized algorithm for a fixed number of additional complicating constraints if the constraints and the objective are encoded in unary. As ILP (W) can encode the NP-hard subset sum problem for $h = 1$, we will need to assume that the coefficients of the constraint matrix are small. That is, we assume that they are bounded by Δ in absolute value. The given randomized XP time algorithm unifies the known tractability in terms of membership in RP of multiple classes of constrained problems where the edges of a graph correspond to variables.

► **Theorem 3.** *ILP (W) can be solved with a randomized XP time algorithm in terms of h . More specifically, it can be solved in time $\tilde{O}(\|c\|_\infty \log^2 \|c\|_\infty \cdot \Delta^h \log^5 \Delta) \cdot \mathcal{O}(n)^{10+h} \cdot \mathcal{O}(\Delta h m)^{8h+h^2} + \mathcal{O}(nm)$.*

In this paper, the running time is measured in the number of arithmetic operations on numbers with encoding length polynomial in the encoding length of the instance. The \tilde{O} hides polylogarithmic factors of the encoding length of the instance.

Camerini, Galbiati and Maffioli [9] already observed in 1992 that one can solve constrained flow, circulation and matching problems in randomized, pseudo-polynomial time. Theorem 3 differs in that it reveals the tractability of a generalized class of problems by removing the exponential dependency on the encoding length of b that would appear in a purely pseudo-polynomial algorithm.

The randomized XP time algorithm is obtained through the use of a Graver augmentation framework, which has been effective in obtaining FPT algorithms for two-stage stochastic and n -fold ILP with small coefficients [17]. For those ILPs, the Graver complexity is bounded by a function of the parameters, resulting in FPT algorithms, whereas the XP time dependence on h of Theorem 3 is the result of ILP (W) having a large Graver complexity and proximity. We provide upper and lower bounds for these respective quantities, see Sections 2.3 and 2.4 for the technical overview. Finally, in Section 2.4, we note that ILP (T) has a similarly large Graver complexity even for binary matrices T , which motivates the use of the different technique.

We provide a W[1]-hardness result in terms of the number of additional constraints h to match the randomized XP time algorithm in a complexity theoretic sense. The hardness persists even for binary constrained perfect matching on a restricted class of graphs. Theorem 4 additionally shows that parameterizing with Δ in addition to the number of complicating constraints h is unlikely to result in an FPT algorithm.

► **Theorem 4.** *Solving ILP (W) encoded in unary is W[1]-hard parameterized by h . This hardness persists when restricting to M being the incidence matrix of a graph which is the disjoint union of even length cycles and $W \in \{0, 1\}^{h \times n}$, $c = \mathbf{0}$, $l = \mathbf{0}$, $u = \mathbf{1}$, $b = \mathbf{1}$.*

We like to conclude the introduction with noting that recent methodology used by Eisenbrand and Rothvoss [18] can be used to derive a similar results to Theorem 2 in a different way, which is presented in Appendix A. This work was carried out independently to ours.

2 Technical Overview

This section gives a technical overview of our results. It aims at presenting the main ideas and new concepts. Therefore, details are omitted.

We first preprocess an ILP instance in Section 2.1, after which both algorithms are covered in Sections 2.2 and 2.3. The W[1]-hardness of Theorem 4 is additionally discussed in the latter section. Section 2.4 provides lower bounds on the Graver complexity and proximity.

2.1 Reduction to perfect b -matching

Before presenting the key components of the algorithms that solve the generalized matching problem with additional variables or constraints, we first discuss a reduction that modifies ILPs (T) and (W) in a favorable way. By doing so, it suffices to give algorithms for such restricted instances, see Sections 2.2 and 2.3.

In particular, we show that ILPs (T) and (W) can, in polynomial time, be transformed to equivalent ILPs where the generalized matching submatrix is transformed to the constraint matrix of the perfect b -matching problem. Such matrix M is a binary matrix with unique columns that each have precisely two nonzero entries, and is the incidence matrix of a simple graph $G(M)$, the *constraint graph*, with vertices $[m]$. Since M is the incidence matrix of $G(M)$, we may identify a variable $i \in [n]$ with the edge of $G(M)$ connecting the vertices j_1, j_2 such that $M_{i,j_1} = M_{i,j_2} = 1$. Now, a b -matching x of $G(M)$ is a solution to the generalized matching ILP with constraint matrix M , nonnegative integral variables $x \in \mathbb{Z}_{\geq 0}^n$, and right-hand-side b . That is, x assigns a value to every edge such that for every vertex i the sum of x_j over the incident edges j is exactly b_i .

The used reduction is an extension of the reduction in [46] from generalized matching to b -matching. However, we show in Lemma 5 that we can also keep track of the additional general variables and constraints.

► **Lemma 5.** *An ILP of the form*

$$\min \left\{ a^\top y + c^\top x \mid \begin{pmatrix} C & W \\ T & M \end{pmatrix} (y, x) = \begin{pmatrix} d \\ b \end{pmatrix}, e \leq y \leq g, l \leq x \leq u, (y, x) \in \mathbb{Z}^{p+n} \right\},$$

and finite variable bounds can be reduced in input+output-linear time to an instance of the same form that additionally satisfies

- $M' \in \{0, 1\}^{m' \times n'}$ is the incidence matrix of a simple graph,
- $c' \geq 0, W' \in \mathbb{Z}_{\geq 0}^{h' \times n'}$,
- $e', l' = 0, \|g'\|_1 = \mathcal{O}(\|g - e\|_1), u' = \infty$,
- $n', m' = \mathcal{O}(n + m)$,
- $h' = h, p' = p$,
- $\|d'\|_1 = \mathcal{O}(\|d\|_1 + \|A\|_1(\|e\|_1 + \|l\|_1))$,
- $\|b'\|_1 = \mathcal{O}(\|u - l\|_1)$ if $p = 0$ and $\|b'\|_1 = \mathcal{O}(\|b\|_1 + \|g - e\|_1 + \|u - l\|_1 + \|A\|_1(\|e\|_1 + \|l\|_1))$ otherwise,
- a', c', C', T', W' contain the same collections of entries as a, c, C, T, W up to sign changes, the insertion of zeros and the insertion of at most one binary row to T .

Here, parameters with primes are used to denote the parameters of the new instance and $u = \infty$ denotes the absence of upper bounds on x . The proof transforms the constraint matrix and variable bounds to the intended form by creating additional constraints and variables. It consists of steps that individually treat columns with negative entries, columns with 1-norm strictly less than 2 and finite variable upper bounds. Most of these obstacles are circumvented by conceptually subdividing edges or adding redundant constraints. Due to space restrictions, we postpone the proof to Section 3.

By virtue of Lemma 5, we can now restrict ourselves to finding algorithms for ILPs (T) and (W) which represent perfect b -matching problems with additional variables or constraints.

2.2 Few arbitrary variables

We now focus on solving ILP (T). We first motivate the main idea underlying the algorithm and present the algorithm itself, before we give an outline of remaining parts. The key idea is to treat the backdoor variables y and the matching variables x separately. Let us first consider the problem of determining the feasibility of ILP (T). In essence, we replace the constraint $Ty + Mx = b$ and matching variables x with a polyhedral constraint of the form $b - Ty \in P$ for some polyhedron P that models that there must exist a $x \in \mathbb{Z}_{\geq 0}^n$ such that $Mx = b - Ty$, in other words, there must exist a perfect $(b - Ty)$ -matching in $G(M)$. To construct such polyhedral constraint on y , the first idea may be to use the convex hull of all points $z \in \mathbb{Z}^m$ such that there exists a perfect z -matching—unfortunately, a naive execution of this strategy fails as when $G(M) = K_3$, the complete graph on 3 vertices, we have that $z = (0, 0, 0)$ and $z = (2, 2, 2)$ and the convex combination $z = (1, 1, 1)$ is in the hull, but the latter does not admit a perfect z -matching. Hence, we attempt to model a discrete set that is not lattice-convex on \mathbb{Z}^3 with a convex constraint, which is impossible.

To contrast this, the work by Cslovjcek et al. [11] shows that, even for general matrices M , one can work around this issue by restricting y to a fixed remainder $r \in \mathbb{Z}^m$ modulo some large integer B . That is, one can construct a polyhedron P_r such that for all $y \in B\mathbb{Z}^m + r$, we have that $\{Mx = b - Ty, x \geq 0\}$ is feasible if and only if $b - Ty \in P_r$. Note that through an affine transformation, this may equivalently be posed as a polyhedral constraint directly on y . Their choice of B may grow with the dimensions of M , which calls for a more problem specific approach to solve ILP (T) in FPT time. When M is the incidence matrix of a simple graph, we show that we may restrict the modulus B to be the constant 2, which paves the way for an FPT algorithm. In addition, we show that one may optimize over a linear objective function by encoding the objective contribution of the matching part of the ILP in an additional dimension of P_r . The required convexity property to make this work translates to a lattice-convexity property, see Lemma 7, of the function $f_{c,M}$ from Definition 6.

► **Definition 6.** Let $f_{c,M} : \mathbb{Z}^m \rightarrow \mathbb{Z} \cup \{\infty\}$ be defined by

$$f_{c,M}(z) = \min\{c^\top x \mid Mx = z, x \in \mathbb{Z}_{\geq 0}^n\}.$$

That is, $f_{c,M}(z)$ is the cost of a minimum c -cost perfect z -matching of $G(M)$ or ∞ if $G(M)$ has no perfect z -matching.

► **Lemma 7.** Let $z^{(1)}, z^{(2)}, \dots, z^{(\ell)} \equiv r \pmod{2}$ be given and $\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(\ell)}$ be real convex multipliers so that $z := \sum_{k \in [\ell]} \lambda^{(k)} z^{(k)} \equiv r \pmod{2}$. Then $f_{c,M}(z) \leq \sum_{k \in [\ell]} \lambda^{(k)} f_{c,M}(z^{(k)})$.

We use modular congruence on vectors to denote component-wise modular congruence. Next, we show in which way Lemma 7 allows us to reformulate ILP (T). First, we perform a binary search on the objective to obtain the problem of finding a feasible point of ILP (T) subject to an additional linear constraint $a^\top y + c^\top x \leq \omega^*$. We guess the remainder vector t of y in an optimal solution modulo 2. It then suffices to search for an integer point y restricted to $y \equiv t \pmod{2}$ in the box $e \leq y \leq g$ subject to the constraint that there is a perfect $(b - Ty)$ -matching with cost at most $\omega^* - a^\top y$. This is equivalent to finding a point in the set

$$\{y \in \mathbb{Z}^P : y \equiv t \pmod{2}, e \leq y \leq g, f(b - Ty) \leq \omega^* - a^\top y\}.$$

In particular, as the parity of $b - Ty \equiv r \pmod{2}$ is constant for $y \equiv t \pmod{2}$, Lemma 7 shows that there exists a convex set P_r such that the constraint $f(b - Ty) \leq \omega^* - a^\top y$ may

be replaced with $(\omega^* - a^\top y, b - Ty) \in P_r$. When we obtain a good representation of P_r , we use the algorithm by Reis and Rothvoss [45] to solve the integer program in FPT time. After guessing all 2^p parity vectors t , an optimal solution to ILP (T) is found.

We now discuss the missing ingredients needed to make the previously discussed algorithm work. First, we explain how Lemma 7 is derived. To do so, we consider jump systems, where degree sequences of graphs and functions such as $f_{c,M}$ have been studied exhaustively [7, 40, 41, 35]. Murota [41] observed that $f_{c,M}$ describes a jump M-convex function, which is a valuated generalization of jump systems. In fact, recent work on the general factor problem [35] reveals that a similar jump M-convex function defined for weighted graph factorizations has additional exploitable properties. Informally, small steps, as in Definition 8, connecting points in the effective domain of such function may be rearranged in any order. For this reason, Kobayashi defines strongly base orderable (SBO) jump systems and a valuated extension. Our function $f_{c,M}$ defined over weighted uncapacitated perfect b -matchings can be seen to also satisfy the properties of Definition 9. As Lemma 7 holds for general SBO jump M-convex functions, we discuss our proof in terms of this abstract property.

► **Definition 8.** A 2-step decomposition of a vector $d \in \mathbb{Z}^m$ is a multiset of steps $p^{(1)}, \dots, p^{(\ell)} \in \mathbb{Z}^m$, satisfying $\|p^{(k)}\|_1 = 2$ and $p^{(k)} \subseteq d$ for all $k \in [\ell]$, such that $d = \sum_{k \in [\ell]} p^{(k)}$.

Here, $x \subseteq y$ is the conformal partial order defined by $x \subseteq y$ if and only if $|x_i| \leq |y_i|$ and $x_i y_i \geq 0$ for all i .

► **Definition 9.** A function $f: \mathbb{Z}^m \rightarrow \mathbb{Q} \cup \{\infty\}$ is SBO jump M-convex if for every two points $z^{(1)}, z^{(2)}$ in the domain $\{z \in \mathbb{Z}^m : f(z) < \infty\}$ there exist a 2-step decomposition $p^{(1)}, \dots, p^{(\ell)}$ of $z^{(2)} - z^{(1)}$ and real numbers $g^{(1)}, \dots, g^{(\ell)}$ such that

- $f(z^{(2)}) = f(z^{(1)}) + \sum_{i \in [\ell]} g^{(i)}$,
- for all $I \subseteq [\ell]$ it holds that $f(z^{(1)} + \sum_{k \in I} p^{(k)}) \leq f(z^{(1)}) + \sum_{k \in I} g^{(k)}$.

The alternating path argument employed by Kobayashi [35] to show that 1-capacitated perfect b -matchings satisfy Definition 9 can be adapted to show the SBO jump M-convexity of $f_{c,M}$. For this, we use a known gadget construction that relates b -matchings to perfect matchings [46]. See Section 4 for the full proof.

Lemma 7 is derived by first showing that a 2-step decomposition of a difference $z^{(2)} - z^{(1)}$ can be used to construct a small even step $\sum_{k \in I} p^{(k)} \in \{-2, 0, 2\}^m$ by combining some of the steps. Such an even, small step can be used to modify a pair of points $z^{(2)}, z^{(1)}$ closer to each other while remaining on the lattice $2\mathbb{Z}^m + r$ and without increasing the sum of the function values on these points. Such pairwise modifications may then be exhaustively performed to move the points appearing in a convex combination towards the target z as in Lemma 7 and obtain the required result.

The final ingredient needed to implement the FPT algorithm for ILP (T) is a good characterization of a suitable polyhedron P_r that models the matching part of the ILP. We obtain this by letting P_r be the convex hull of the vectors $(\omega, z) \in \mathbb{R} \times \mathbb{Z}^m$ for which $f_{c,M}(z) \leq \omega$ and z is in some bounding box. That is, P_r is the epigraph of a natural convex extension of $f_{c,M}$ restricted to a bounding box. Since proximity arguments allow us to assume that e and g are finite, it is possible to bound z , see [10]. We obtain a polynomial time separation oracle for P_r through the ellipsoid method [27] and by noting that P_r can efficiently be optimized over³. This follows from the fact that weights associated with a vertex i may

³ We note that it may be possible to derive a combinatorial separation algorithm, but this is not needed to establish the fixed-parameter tractability of ILP (T). For a closely related separation problem, see [49].

be transferred to the edge weights of the incident edges, which shows that optimizing over $(\omega, z) \in P_r$ corresponds to solving a minimum cost parity constrained b -matching problem. The parity constraints can straightforwardly be translated into constraints compatible with the polynomially solvable generalized matching problem [46, 15].

2.3 Few arbitrary constraints

The randomized XP time algorithm for solving ILP (W) is obtained through a series of reductions. As a key intermediate step, we will employ a pseudo-polynomial reduction in [46] that transforms the minimum cost perfect b -matching to the minimum cost perfect matching problem on a graph with a total of $\|b\|_1$ vertices. In order to prevent an exponential blow-up in terms of the encoding size of b , we split the solution process up into parts in which b may be bounded through the use of the Graver augmentation framework [17]. This is an exact local search framework which iteratively improves a primal solution to an ILP by taking smaller steps solving a similar, but slightly easier ILP. In these easier ILPs, the variable domains may be bounded by a function of the Graver complexity of the constraint matrix of the ILP. By Lemma 5, bounds on the variable domains translate into bounds on the right-hand side b . Before providing the required bounds, we first introduce the Graver augmentation framework and relevant definitions.

► **Definition 10.** *The Graver basis $\mathcal{G}(A) \subseteq \mathbb{Z}^n \setminus \{\mathbf{0}\}$ of an integer matrix $A \in \mathbb{Z}^{m \times n}$ is the set of conformally minimal nonzero integral kernel elements of A . That is, $g \in \mathcal{G}(A)$ if and only if $g \neq \mathbf{0}$, $Ag = \mathbf{0}$ and there is no $g' \notin \{\mathbf{0}, g\}$ such that $Ag' = \mathbf{0}$ and $g' \sqsubseteq g$.*

The Graver complexity of A refers to norm bounds on $\mathcal{G}(A)$, and we are interested in $g_\infty(A) = \max\{\|g\|_\infty \mid g \in \mathcal{G}(A)\}$ and $g_1(A) = \max\{\|g\|_1 \mid g \in \mathcal{G}(A)\}$, referring to the ∞ -norm and 1-norm respectively. An oracle that can compute a single small improving step, used in the local search framework, is called a Graver-best oracle.

► **Definition 11.** *A Graver-best oracle is an oracle that, given variable bounds l and u finds a Graver-best step. That is, it finds an integral solution x to $\{Ax = \mathbf{0}, l \leq x \leq u\}$ so that*

$$c^\top x \leq \min \{c^\top g \mid l \leq g \leq u, g \in \mathcal{G}(A)\}.$$

Such an oracle can be implemented by solving an ILP over variable domains of size at most $\mathcal{O}(g_\infty(A))$, which we show to be polynomially bounded for fixed h for ILP (W). In this case, we may employ Lemma 5 to find that implementing a Graver-best oracle corresponds to finding a minimum cost constrained b -matching where $\|b\|_1 \leq n\|b\|_\infty = \mathcal{O}(ng_\infty(A))$ is polynomially bounded. Theorem 12 concludes with how this Graver-best oracle can be used to optimize an ILP.

► **Theorem 12** (Lemma 12 in [17], Lemma 5 in [16]). *Given a feasible solution x to ILP (G) with finite l and u , an optimal solution to the ILP can be found with $\mathcal{O}(n \log(\|l - u\|_\infty) \log(c^\top x - c^\top x^*))$ queries of a Graver-best oracle. Here $c^\top x^*$ is the value of an optimal solution.*

By constructing an auxiliary ILP with a trivial feasible solution to the form ILP (W), we can use Theorem 12 to find a feasible solution as well as optimize such solution. For this reason, we will assume that we are given a feasible solution to ILP (W).

The polynomial bound on $g_\infty(A)$ for fixed h follows from literature: Berndt, Mnich and Stamm [6] show that the Graver basis elements of a matrix M with $\|M\|_1 \leq 2$ are small.

► **Theorem 13** (Theorem 13 in [6]). *Let $M \in \mathbb{Z}^{m \times n}$ be an integer matrix with $\|M\|_1 \leq 2$. Then $g_\infty(M) \leq 2$ and $g_1(M) \leq 2m + 1$.*

Note that we may assume that $m = \mathcal{O}(n)$ after preprocessing. Lemma 3 in [16], which proves a Graver complexity bound for n -fold ILPs, reveals that Theorem 13 suffices to yield a polynomial bound on the Graver complexity of the composite matrix appearing in ILP (W).

► **Corollary 14.**

$$g_\infty \left(\begin{pmatrix} W \\ M \end{pmatrix} \right) \leq 2(2\Delta h(2m + 1) + 1)^h = \mathcal{O}(\Delta h m)^h.$$

The number of Graver-best steps that need to be computed in Theorem 12 can be polynomially bounded by computing a solution to the LP relaxation of ILP (W) and using Theorem 3.14 in [29], which shows that the proximity of an ILP is bounded by n times the obtained Graver bound from Corollary 14.

► **Corollary 15.** *Let y^* be an optimal solution to the LP relaxation of ILP (W). If the ILP is feasible, then there exists an optimal ILP solution x^* with $\|y^* - x^*\|_\infty = n \cdot \mathcal{O}(\Delta h m)^h$.*

Thus, we may restrict the search of an optimal solution to ILP (W) to a polynomially bounded domain, which results in the needed bound on $c^\top x - c^\top x^*$ to apply Theorem 12. Now, after applying the pseudo-polynomial reduction to a perfect matching problem [46], which is compatible with additional constraints as noted in [9], we find that solving ILP (W) can be polynomially reduced to finding a minimum cost constrained perfect matching in a graph with polynomially many vertices. As all variables in this resulting problem are binary, we can condense all h constraints into a single constraint by encoding the constraints in base- B for sufficiently large B . This generalizes the reduction steps used in [5, 44] and only increases the coefficient size of the constraint matrix polynomially for fixed h . The resulting constrained perfect matching problem can then be solved with a randomized pseudo-polynomial algorithm such as the one by Mulmuley, Vazirani and Vazirani [39]. Combining all steps and binary searching on the values of the variables in an optimal solution yields the randomized XP time algorithm from Theorem 3.

To the best of our knowledge, the complexity of constrained matching where the objective is encoded in binary is still open. It is worth noting that the reduction chain shows that ILP (W) is polynomially equivalent to the exact matching problem, which aims to find a perfect matching in a graph with exactly a given target weight. Finding a deterministic pseudo-polynomial algorithm for this problem has been a central and intensively studied open problem for over four decades.

Finally, to complement the XP time complexity of the algorithm from Theorem 3, we reveal the W[1]-hardness of solving ILP (W) parameterized by h in Theorem 4. This result is obtained through a parameterized reduction from the ILP problem, which is strongly W[1]-hard when parameterized by the number of constraints [13].

► **Theorem 16** (Theorem 22 in [13]). *Determining the feasibility of the system*

$$\{Wx = d : x \in \mathbb{Z}_{\geq 0}^n\},$$

where $W \in \mathbb{Z}_{\geq 0}^{h \times n}$ is encoded in unary and $d \in \{0, 1\}^h$, is W[1]-hard parameterized by h . This hardness persists when restricting x to $\mathbf{0} \leq x \leq u$ where u is encoded in unary.

The essence of the reduction is to split every variable x_i into u_i many binary variables. The resulting binary variables can then be duplicated and linked through graph cycles to reduce the unary sized coefficients in W and obtain a binary constraint matrix.

2.4 Lower bounds

We finish the overview by presenting Graver complexity and proximity lower bounds for ILPs (T) and (W). These are shown through explicit constraint matrix constructions, which can be found in Sections 4 and 5.

First, we observe that the Graver augmentation framework is unlikely to assist in deriving simpler FPT algorithms to solve ILP (T), as Proposition 17 shows.

► **Proposition 17.** *For any fixed positive integer p , there exists an infinite family of binary constraint matrices $(T \ M) \in \{0, 1\}^{m \times (p+n)}$ of ILP (T) with Graver basis elements that have an ∞ -norm of order $\Omega(n)^p$.*

Proposition 18 gives a construction showing that Corollary 14 is asymptotically tight when h is constant.

► **Proposition 18.** *For any fixed positive integer h , there exists an infinite family of constraint matrices $\begin{pmatrix} W \\ M \end{pmatrix} \in \{0, 1, \Delta\}^{(h+m) \times n}$ of ILP (W) with Graver basis elements that have an ∞ -norm of order $\Omega(\Delta m)^h$.*

Additionally, a similar construction as in the proof of Proposition 18 shows that Corollary 15 is tight for constant h and $n = \Theta(m)$. We note that a conjecture posed by Berndt, Mnich and Stamm involving upper bounds of the proximity of ILPs and their relation to the Graver-complexity [6] hints at that the upper bound from Corollary 14 may be able to be improved to $m \cdot \mathcal{O}(\Delta m)^h$, at least for the case where $u = \infty$.

► **Proposition 19.** *For any fixed positive integer h , there exists an infinite family of constraint matrices $\begin{pmatrix} W \\ M \end{pmatrix} \in \{0, 1, \Delta\}^{(h+m) \times n}$ of ILP (W) and vertex LP relaxation solutions to the corresponding ILP (W) such that the nearest integral solution is at ∞ -norm distance $\Omega(m) \cdot \Omega(\Delta m)^h$.*

We note that constant multiplicative factors of $1/p$ and $1/h$ in the lower bounds of Propositions 17–19 are hidden in the Ω .

3 Reduction to perfect b -matching

We proceed to present the details of Section 2 and prove the required intermediate results. First, we consider Lemma 5 which simplifies the design of the algorithms for ILPs (T) and (W) by showing that it suffices to give algorithms exploiting backdoors to b -matching problems. To do so, we generalize the reduction steps in [46] that transform the generalized matching problem to the perfect b -matching problem. Most steps of Lemma 5 can be interpreted as gadget constructions in bidirected graphs, see [46], but we present them in terms of ILPs to retain a direct connection with the additional variables and linear constraints.

► **Lemma 5.** *An ILP of the form*

$$\min \left\{ a^\top y + c^\top x \mid \begin{pmatrix} C & W \\ T & M \end{pmatrix} (y, x) = \begin{pmatrix} d \\ b \end{pmatrix}, e \leq y \leq g, l \leq x \leq u, (y, x) \in \mathbb{Z}^{p+n} \right\},$$

and finite variable bounds can be reduced in input+output-linear time to an instance of the same form that additionally satisfies

- $M' \in \{0, 1\}^{m' \times n'}$ is the incidence matrix of a simple graph,
- $c' \geq 0, W' \in \mathbb{Z}_{\geq 0}^{h' \times n'}$,

XX:12 Parameterized Algorithms for Matching IPs with Additional Rows and Columns

- $e', l' = \mathbf{0}, \|g'\|_1 = \mathcal{O}(\|g - e\|_1), u' = \infty,$
- $n', m' = \mathcal{O}(n + m),$
- $h' = h, p' = p,$
- $\|d'\|_1 = \mathcal{O}(\|d\|_1 + \|A\|_1(\|e\|_1 + \|l\|_1)),$
- $\|b'\|_1 = \mathcal{O}(\|u - l\|_1)$ if $p = 0$ and $\|b'\|_1 = \mathcal{O}(\|b\|_1 + \|g - e\|_1 + \|u - l\|_1 + \|A\|_1(\|e\|_1 + \|l\|_1))$ otherwise,
- a', c', C', T', W' contain the same collections of entries as a, c, C, T, W up to sign changes, the insertion of zeros and the insertion of at most one binary row to T .

Proof. We split the reduction into multiple steps where we restrict the generalized incidence matrix M and the variable bounds on the variables x corresponding to the matching. To accomplish this, we add new columns and rows to the matrix M and add corresponding zero columns and rows to T and W respectively, unless mentioned otherwise. In each step, transformations are performed on the previous instances and the growth of the instance parameters are expressed in terms of the result of the previous step.

▷ **Claim.** All objective coefficients and columns of A corresponding to the variables x can be made nonnegative while additionally ensuring that $M' \in \{0, 1\}^{m' \times n'}$ at the cost of increasing $n', m' = \mathcal{O}(n + m)$.

Proof. We split this into three steps.

Inversion of nonpositive columns of M . First, multiply any column j for which $M_{\cdot,j}$ has a negative coefficient and no positive coefficients by -1 as shown in Figure 2.

$$\begin{array}{c} -c_j \\ -A_{\cdot,j} \\ [l_j, u_j] \end{array} \rightarrow \begin{array}{c} -c_j \\ -A_{\cdot,j} \\ [-u_j, -l_j] \end{array}$$

■ **Figure 2** A column corresponding to an edge with total negative sign can be inverted to reduce the number of negative coefficients in the matching matrix.

That is, we replace the j -th column $A_{\cdot,j}$ of A with $-A_{\cdot,j}$, the domain $[l_j, u_j]$ of x_j with $[-u_j, -l_j]$ and its objective coefficient c_j with $-c_j$.

Ensuring the nonnegativity of c and W . Second, we ensure that $c \geq \mathbf{0}, W \in \mathbb{Z}_{\geq 0}^{h \times n}$ and, as a side product, additionally eliminate entries with a coefficient of 2. To do so, we split the columns into their positive and negative parts as visualized in Figure 3.

$$\begin{array}{c}
\begin{array}{|c|} \hline c_j \\ \hline W_{:,j} \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline M_{i_1,j} \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline M_{i_2,j} \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline [l_j, u_j] \\ \hline \end{array}
\end{array} = b_{i_1} \rightarrow \begin{array}{c}
\begin{array}{|c|} \hline (c_j)_+ \\ \hline (W_{:,j})_+ \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline M_{i_1,j} \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 0 \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 1 \\ \hline [l_j, u_j] \\ \hline \end{array}
\end{array} = b_{i_1} \rightarrow \begin{array}{c}
\begin{array}{|c|} \hline (-c_j)_- \\ \hline (-W_{:,j})_- \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 0 \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline -M_{i_2,j} \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 1 \\ \hline [-u_j, -l_j] \\ \hline \end{array}
\end{array} = b_{i_2}$$

(a) Splitting a column without a 2 coefficient

$$\begin{array}{c}
\begin{array}{|c|} \hline c_j \\ \hline W_{:,j} \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 2 \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline [l_j, u_j] \\ \hline \end{array}
\end{array} = b_i \rightarrow \begin{array}{c}
\begin{array}{|c|} \hline (c_j)_+ \\ \hline (W_{:,j})_+ \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 1 \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline -1 \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 1 \\ \hline [l_j, u_j] \\ \hline \end{array}
\end{array} = b_i \rightarrow \begin{array}{c}
\begin{array}{|c|} \hline (-c_j)_- \\ \hline (-W_{:,j})_- \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline -1 \\ \hline \mathbf{0} \\ \hline \end{array} \\
\begin{array}{|c|} \hline 1 \\ \hline [-u_j, -l_j] \\ \hline \end{array}
\end{array} = 0$$

(b) Splitting a column with a 2 coefficient

Figure 3 A column may be split into its positive and negative part to ensure that all the objective coefficients and coefficients corresponding to the additional global constraints are nonnegative.

For this, let $a_+ = \max\{0, a\}$ and $a_- = \min\{0, a\}$ denote the positive and negative part of some scalar a and employ the same notation for coordinate-wise vector operations. For every column $j \in [n]$ such that $\|M_{:,j}\|_\infty \leq 1$ for which its two nonzero entries are contained in $\{M_{i_1,j}, M_{i_2,j}\}$ and $M_{i_1,j} \geq 0$, we implement the following modifications:

- Split x_j into the variables $x_j^{(+)}, x_j^{(-)}$ where $x_j^{(+)} \in [l_j, u_j], x_j^{(-)} \in [-u_j, -l_j]$.
- Introduce the constraint $x_j^{(+)} + x_j^{(-)} = 0$ so that $x_j^{(-)} = -x_j^{(+)}$.
- Split the original objective term $c_j x_j$ into $(c_j)_+ x_j^{(+)} + (-c_j)_- x_j^{(-)}$.
- Similarly replace $W_{:,j} x_j$ with $(W_{:,j})_+ x_j^{(+)} + (-W_{:,j})_- x_j^{(-)}$.
- Replace the term $M_{i_1,j} x_j$ in the i_1 -th constraint with $M_{i_1,j} x_j^{(+)}$.
- Replace the term $M_{i_2,j} x_j$ in the i_2 -th constraint with $-M_{i_2,j} x_j^{(-)}$.

For a column with a single coefficient of $2 = M_{i,j}$, we may apply the same procedure by interpreting $i_1 = i_2 = i, M_{i_1,j} = M_{i_2,j} = 1$ and replacing the term $2x_j$ with $1x_j^{(+)} - 1x_j^{(-)}$.

Eliminating mixed sign columns in M . Third, for a remaining column $M_{:,j}$ with nonzero coefficients -1 and 1 we again split x_j into $x_j^{(+)} \in [l_j, u_j]$ and $x_j^{(-)} \in [-u_j, -l_j]$ as shown in Figure 4.

$$\begin{array}{ccc|c}
 -\frac{c_j}{W_{\cdot,j}} & -\frac{c_j}{W_{\cdot,j}} & -\frac{0}{0} & \\
 \mathbf{0} & \mathbf{0} & \mathbf{0} & \\
 -1 & 0 & 1 & \\
 \mathbf{0} & \mathbf{0} & \mathbf{0} & \\
 1 & 1 & 0 & \\
 \mathbf{0} & \mathbf{0} & \mathbf{0} & \\
 & 1 & 1 & \\
 \hline
 [l_j, u_j] & [l_j, u_j] & [-u_j, -l_j] & = 0
 \end{array}$$

■ **Figure 4** An edge with mixed signs can be split into two positively signed edges which take opposing values.

This encompasses the following:

- Replace $1x_j$ with $1x_j^{(+)}$ and $-1x_j$ with $1x_j^{(-)}$.
- Add the constraint $x_j^{(+)} + x_j^{(-)} = 0$.
- Copy the coefficients of W and c corresponding to x_j over to $x_j^{(+)}$.
- Set the corresponding coefficients of $x_j^{(-)}$ to zero.

After this, all columns of M have either no nonzero coefficients, a single 1 coefficient or two 1 coefficients. The size of the ILP in terms of $n + m$ is increased by a constant factor. \triangleleft

▷ **Claim.** All lower bounds e, l can be set to $e', l' = \mathbf{0}$ resulting in $\|g'\|_1 = \|g - e\|_1, \|u'\|_1 = \|u - l\|_1, \|d'\|_1 = \mathcal{O}(\|d\|_1 + \|A\|_1(\|e\|_1 + \|l\|_1)), \|b'\|_1 = \mathcal{O}(\|b\|_1 + \|A\|_1(\|e\|_1 + \|l\|_1))$ and, if $p = 0$, one may restrict $\|b'\| = \mathcal{O}(\|u - l\|_1)$.

Proof. We translate the system by substituting $x' = x - l$. This yields new bounds $(e', l') = \mathbf{0}, (g', u') = (g, u) - (e, l)$ and right-hand side vector $(d', b') = (d, b) - A(e, l)$. The objective becomes a translation of the original. The bounds on $\|d'\|_1$ and $\|b'\|_1$ follow. For the special case where $p = 0$, we may observe that $\|b'\|_1 \leq \|M\|_1 \|u'\|_1 \leq 2\|u - l\|_1 = \mathcal{O}(\|u - l\|_1)$ holds or the system must be trivially infeasible. \triangleleft

▷ **Claim.** All columns of M can be extended to full sum $\|M_{\cdot,j}\|_1 = 2$ by increasing $\|u'\|_1 = \mathcal{O}(\|g\|_1 + \|u\|_1)$ and $\|b'\|_1 = \mathcal{O}(\|b_1\| + \|g\|_1 + \|u\|_1)$, and adding a binary vector to T .

Proof. First, we treat the columns with $\|M_{\cdot,j}\|_1 = 1$. For this purpose, add a new variable $s^{(1)}$ with zero objective coefficient that will occur in a new redundant constraint. Let $B = (T \ M)$ be the lower part of the constraint matrix A . Now, for every column $B_{\cdot,j}$ of which the sum of the entries is odd, we introduce a coefficient of one in the new constraint. More formally, we define the row vector $r^{(1)}$ by $r^{(1)} = \mathbf{1}^\top B \bmod 2$ and add the constraint $2s^{(1)} + r^{(1)}(y, x) = t^{(1)}$ where $t^{(1)} = (\mathbf{1}^\top b \bmod 2) + 2\lceil r^{(1)}(g, u)/2 \rceil$ to the ILP. Here, $\bmod 2$ denotes the component-wise remainder of the vector modulo 2. By assigning $s^{(1)}$ a sufficiently large domain and ensuring that all integral solutions x to the original problem instance satisfy that $t^{(1)} - r^{(1)}(y, x) \equiv \mathbf{1}^\top b - r^{(1)}(y, x) \equiv 0 \pmod{2}$, this constraint becomes redundant and preserves the equivalence of the problems. For this, observe that $\mathbf{1}^\top B(y, x) = \mathbf{1}^\top b$ when (y, x) is a feasible solution, which shows that $\mathbf{1}^\top b - r^{(1)}(y, x) = \mathbf{1}^\top B(y, x) - r^{(1)}(y, x) \equiv 0 \pmod{2}$. By construction, it is admissible to restrict $s^{(1)} \in [0, \lceil r^{(1)\top}(g, u)/2 \rceil]$, which increases $\|u\|_1$ by at most $\|g\|_1 + \|u\|_1$.

A similar technique can be used to treat the zero columns of M . Define the row vector $r^{(2)}$ by $r_j^{(2)} = 2$ if j corresponds to a column of M and $M_{\cdot,j} = \mathbf{0}$, and $r_j^{(2)} = 0$ otherwise. We may add the redundant constraint $2s^{(2)} + r^{(2)}(y, x) = 2\|u\|_1$ and restrict $s^{(2)} \in [0, \|u\|_1]$. The coefficients 2 in M introduced in this step can be eliminated without increasing the asymptotic size of the ILP as shown in the earlier step which splits the column. \triangleleft

\triangleright **Claim.** One can set $u' = \infty$ and restrict M to be the incidence matrix of a simple graph while only increasing $\|b'\|_1 = \mathcal{O}(\|b\|_1 + \|u\|_1)$.

Proof. The edge capacities are removed by essentially subdividing every edge into three edges and two intermediate vertices as shown in Figure 5.

$$\begin{array}{c|c|c|c|c}
 \begin{array}{c} c_j \\ \hline W_{\cdot,j} \\ \hline \mathbf{0} \\ 1 \\ \mathbf{0} \\ 1 \\ \mathbf{0} \\ \hline [0, u_j] \end{array} & & \begin{array}{c} c_j \\ \hline W_{\cdot,j} \\ \hline \mathbf{0} \\ 1 \\ \mathbf{0} \\ 0 \\ \mathbf{0} \\ 1 \\ 0 \\ \hline [0, \infty) \end{array} & \begin{array}{c} 0 \\ \hline \mathbf{0} \\ 0 \\ 0 \\ \mathbf{0} \\ 1 \\ 1 \\ 1 \\ \hline [0, \infty) \end{array} & \begin{array}{c} 0 \\ \hline \mathbf{0} \\ 0 \\ 1 \\ \mathbf{0} \\ 0 \\ 1 \\ 1 \\ \hline [0, \infty) \end{array} \\
 & & \rightarrow & & \\
 & & & & \begin{array}{c} = b_{i_1} \\ \\ \\ = b_{i_2} \\ \\ \\ = u_j \\ = u_j \end{array}
 \end{array}$$

■ **Figure 5** Variable upper bounds can be eliminated by modeling the slack of the constraint with an intermediate variable.

To accomplish this, we implement the following modifications:

- Split every variable x_j with a coefficient 1 in rows i_1 and i_2 into $x_j^{(1)}, x_j^{(s)}, x_j^{(2)}$, each with lower bound 0 and no upper bound.
- Variable $x_j^{(1)}$ copies the column $W_{\cdot,j}$ and objective coefficient c_j from x_j .
- The other variables $x_j^{(s)}, x_j^{(2)}$ have zeroes in their respective parts of W and c .
- Add the constraints $x_j^{(1)} + x_j^{(s)} = u_j$ and $x_j^{(2)} + x_j^{(s)} = u_j$ so that $x_j^{(s)}$ models the slack of the capacity constraint $x_j \leq u_j$ and $x_j^{(1)} = x_j^{(2)}$.
- Replace the term x_j with $x_j^{(1)}$ in row i_1 and with $x_j^{(2)}$ in row i_2 .

\triangleleft

The bounds on the coefficients can be derived by following the reduction steps taken and suitably substituting the size increments. \blacktriangleleft

Having shown that the instances of ILP (G) may be restricted to be a perfect b -matching problem with additional complicating variables or constraints, we may rely on this in the algorithm constructions in Sections 4 and 5.

4 Few arbitrary variables

This section is devoted to proving Theorem 2 as outlined in Section 2.2, which shows that ILP is FPT when parameterized by the number of columns with 1-norm larger than 2.

► **Theorem 2.** *ILP (T) is FPT parameterized by p .*

Before discussing the algorithm itself, we observe that it is unlikely that a Graver-best augmentation procedure along the lines of Theorem 12 will be effective at solving ILP (T), even in the case that the coefficients in the block T are bounded by a constant.

► **Proposition 17.** *For any fixed positive integer p , there exists an infinite family of binary constraint matrices $(T \ M) \in \{0,1\}^{m \times (p+n)}$ of ILP (T) with Graver basis elements that have an ∞ -norm of order $\Omega(n)^p$.*

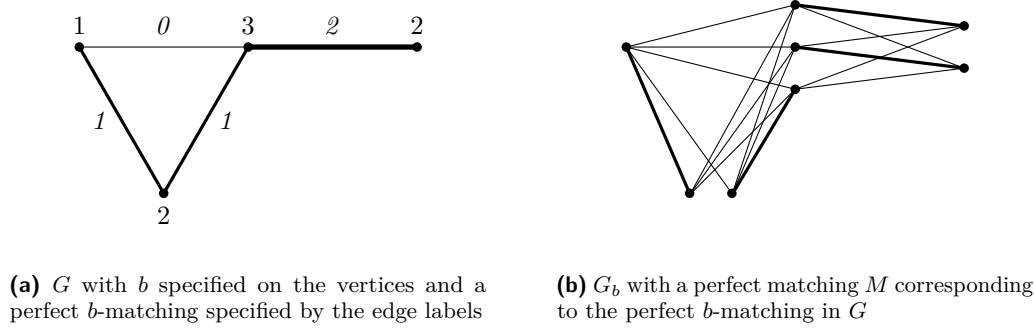
Proof. Consider the matrix

$$(T \ M) = \left(\begin{array}{cccccc|cccccc|cccc} 1 & 0 & 0 & \cdots & 0 & 0 & I & O & O & \cdots & O & O & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & O & I & O & \cdots & O & O & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & O & O & I & \cdots & O & O & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & O & O & O & \cdots & I & O & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & O & O & O & \cdots & O & I & 0 \\ \hline 0 & 1 & 0 & \cdots & 0 & 0 & \mathbf{1}^\top & \mathbf{0}^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top & \mathbf{0}^\top & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & \mathbf{0}^\top & \mathbf{1}^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top & \mathbf{0}^\top & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \mathbf{0}^\top & \mathbf{0}^\top & \mathbf{1}^\top & \cdots & \mathbf{0}^\top & \mathbf{0}^\top & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & \mathbf{0}^\top & \mathbf{0}^\top & \mathbf{0}^\top & \cdots & \mathbf{1}^\top & \mathbf{0}^\top & 0 \\ \hline 0 & 0 & 0 & \cdots & 0 & 0 & \mathbf{0}^\top & \mathbf{0}^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top & \mathbf{1}^\top & 1 \end{array} \right)$$

where the vectors and submatrices that comprise the matrix have dimension k for a given $k \in \mathbb{Z}_{\geq 1}$. Here O denotes the all-zero matrix and I the identity matrix. By construction, any nonzero kernel element must have that the absolute value of the last entry is k^p times the absolute value of the first entry. Such nonzero kernel element also exists. Therefore, there must be Graver basis elements with ∞ -norm at least k^p . We conclude with $n = pk + 1$, i.e., $k = (n - 1)/p$. ◀

Recall that our strategy to solve ILP (T) is to guess the remainder of the p complicating variables modulo 2 and then solve an ILP on only these p variables by modeling the remaining matching part of the problem using a polyhedral constraint. For this, we require the convexity result from Lemma 7. Recall Definitions 6 and 9. First, we modify the alternating path argument of Kobayashi [35] to show that the function $f_{c,M}$ which models the cost of a minimum cost b -matching is SBO jump M-convex. Then, we show that an arbitrary jump M-convex function f satisfies the required lattice convexity on $2\mathbb{Z}^m + r$. The essence of Kobayashi's argument is to consider two optimal matchings and decompose their difference into alternating paths. The endpoints of these paths then yield a suitable 2-step decomposition $p^{(k)}$ and the changes they induce in the cost of the matchings yield the required objective steps $g^{(k)}$. As Kobayashi's argument works with $\mathbf{1}$ -capacitated b -matchings, we need a slightly different approach to obtain suitable alternating paths. For this purpose, we present a construction used to pseudo-polynomially reduce the perfect b -matching problem to a perfect matching problem in Proposition 20 [46]. The construction expands the vertices and edges of the original graph into multiple copies so that assigning a value of x_e to e in a b -matching in G corresponds to choosing x_e copies of e in the constructed graph. This is visualized in Figure 6.

► **Proposition 20** (Section 31.1 in [46]). *Let $b, \bar{b} \in \mathbb{Z}_{\geq 0}^V$ satisfy $b \leq \bar{b}$ and $G = (V, E)$ be a simple graph. Let $G_{\bar{b}}$ be the graph defined by creating \bar{b}_v copies of each vertex $v \in V$ and*



■ **Figure 6** A pseudo-polynomial reduction construction between matching and b -matching [46].

creating an edge between every pair of copies of v_1 and v_2 when v_1 and v_2 are adjacent in G . Then G has a perfect b -matching if and only if $G_{\bar{b}}$ has a matching saturating exactly b_v copies of v for each $v \in V$. In particular, a given x is a perfect b -matching of G if and only if $G_{\bar{b}}$ has a matching M such that M contains exactly $x_{\{v_1, v_2\}}$ edges between the copies of v_1 and v_2 and saturates exactly b_v copies of v .

By using this construction for an appropriate \bar{b} , we can use the alternating paths appearing in Kobayashi's [35] argument in $G_{\bar{b}}$ to provide a suitable 2-step decomposition for points in the domain of $f_{c,M}$.

► **Proposition 21.** *The function $f_{c,M}(z) = \min\{c^\top x \mid Mx = z, x \in \mathbb{Z}_{\geq 0}^n\}$ is SBO jump M -convex.*

Proof. Let $z^{(1)}, z^{(2)} \in \mathbb{Z}^m$ be arbitrary integer points in the domain of $f_{c,M}$. For $k = 1, 2$, let $x^{(k)}$ be a perfect $z^{(k)}$ -matching with minimum cost $f_{c,M}(z^{(k)})$. Consider arbitrary associated matchings $M^{(1)}$ and $M^{(2)}$ in $G_{\bar{z}}$ where $\bar{z} = \max\{z^{(1)}, z^{(2)}\}$ is the coordinate-wise maximum of $z^{(1)}$ and $z^{(2)}$. The spanning subgraph with the symmetric difference between $M^{(1)}$ and $M^{(2)}$ as edge set can be uniquely decomposed into disjoint paths and cycles of which the edges alternate between $M^{(1)}$ and $M^{(2)}$. We associate an alternating walk W with a vector $d \in \mathbb{Z}^n$, defined over the edges $[n]$ in the original graph, so that it corresponds to the change in the b -matching x when adding edges to and removing edges from the matching in $G_{\bar{z}}$ along this walk. For this purpose, we define d by setting d_i to be the number of edges that are copies of i in $W \cap M^{(2)}$ minus the number of copies in $W \cap M^{(1)}$.

In this way, the difference $x^{(2)} - x^{(1)}$ decomposes as $x^{(2)} - x^{(1)} = \sum_k d^{(k)}$, where k ranges over all decomposed paths and cycles. Similarly, when defining $p^{(k)} = Md^{(k)}$ we find a decomposition $z^{(2)} - z^{(1)} = \sum_k Md^{(k)}$. As an alternating cycle induces a difference vector d with $Md = 0$, they may be left out of the decomposition of $z^{(2)} - z^{(1)}$. For this reason, assume that the indices $1, \dots, \ell$ index the difference vectors associated with the alternating paths and consider the decomposition $z^{(2)} - z^{(1)} = \sum_{k \in [\ell]} p^{(k)}$. From the fact that a vector $d^{(k)}$ is induced by an alternating path, it follows that $p^{(k)}$ has a 1-norm of zero or two. We may observe that $d^{(k)}$ cannot be the zero vector as this would imply that $b_i^{(1)} < \bar{b}_i$ and $b_i^{(2)} < \bar{b}_i$ for the vertex i in G that corresponds to the beginning and end vertices of the path. This is impossible due to the tight choice of \bar{z} . An analogous argument shows that each step $d^{(k)}$ is sign-compatible with $z^{(2)} - z^{(1)}$, which allows us to conclude that $p^{(1)}, \dots, p^{(\ell)}$ is indeed a two-step decomposition of $z^{(2)} - z^{(1)}$.

This also suggests to define $g^{(k)}$ as $c^\top d^{(k)}$. Note that $c^\top d$ must be 0 for a difference vector d induced by an alternating cycle, because if it were not, one of $M^{(1)}$ or $M^{(2)}$ is not

of minimum cost. This shows that $f_{c,M}(z^{(2)}) - f_{c,M}(z^{(1)}) = \sum_{k \in [\ell]} g^{(k)}$. Note that, for any $I \subseteq [\ell]$, the vector $x^{(1)} + \sum_{k \in I} d^{(k)}$ is a proper perfect $(z^{(1)} + \sum_{k \in I} p^{(k)})$ -matching with cost $f_{c,M}(z^{(1)}) + \sum_{k \in I} g^{(k)}$, showing that $f_{c,M}(z^{(1)} + \sum_{k \in I} p^{(k)})$ is bounded from above as required. \blacktriangleleft

We now proceed to show the lattice-convexity of SBO jump M-convex functions. The proof is split into two steps. Lemma 22 shows how two points in a convex combination can be pairwise modified without increasing the overall sum of function values. This operation is then exhaustively employed in the proof of the final lattice-convexity result.

► **Lemma 22.** *Let f be an SBO jump M-convex function. Let $z^{(1)}, z^{(2)} \equiv r \pmod{2}$ and let $i^* \in [m]$ be so that $z_{i^*}^{(1)} - z_{i^*}^{(2)} \geq 2$. Then, there exist $z^{(1)'}$ and $z^{(2)'}$ such that*

- $z^{(1)'}, z^{(2)'} \equiv r \pmod{2}$,
- $z^{(1)'} + z^{(2)'} = z^{(1)} + z^{(2)}$,
- $f(z^{(1)'}) + f(z^{(2)'}) \leq f(z^{(1)}) + f(z^{(2)})$,
- $z_{i^*}^{(1)'} = z_{i^*}^{(1)} - 2, z_{i^*}^{(2)'} = z_{i^*}^{(2)} + 2$,
- $z_i^{(1)} = z_i^{(2)} \implies z_i^{(1)'} = z_i^{(1)} = z_i^{(2)'} = z_i^{(2)}$ for all $i \in [m]$.

Proof. Consider a 2-step decomposition $p^{(1)}, \dots, p^{(\ell)}$ of $z^{(2)} - z^{(1)}$ and $g \in \mathbb{R}^\ell$ from Definition 9. We construct a set $I \subseteq [\ell]$ that corresponds with a suitable even step $\sum_{k \in I} p^{(k)} \in \{-2, 0, 2\}^n$ from $z^{(1)}$ to $z^{(2)}$, which will define $z^{(1)'}$. First, pick an arbitrary step $k \in [\ell]$ to include in I so that $p_{i^*}^{(k)} < 0$. Then, as long as there is a coordinate i such that $\sum_{k \in I} p_i^{(k)} \in \{-1, 1\}$, we pick an additional step $k' \in [\ell] \setminus I$ with $p_i^{(k')} = \sum_{k \in I} p_i^{(k)}$ to include into I . Such step always exists because the difference $z^{(2)} - z^{(1)}$ is even. Additionally, as $[\ell]$ is finite, this process ends with a set of steps I . As suggested, define $z^{(1)'} = z^{(1)} + \sum_{k \in I} p^{(k)}$ and $z^{(2)'} = z^{(2)} - \sum_{k \in I} p^{(k)}$. Observe that $f(z^{(1)'}) + f(z^{(2)'}) \leq f(z^{(1)}) + \sum_{k \in I} g^{(k)} + f(z^{(1)}) + \sum_{k \in [\ell] \setminus I} g^{(k)} = f(z^{(1)}) + f(z^{(2)})$. The other properties follow straightforwardly by construction. \blacktriangleleft

We can now exhaustively employ Lemma 22 pairwise on vectors of a given convex combination to derive the lattice-convexity of f . Lemma 23 generalizes Lemma 7.

► **Lemma 23.** *Let f be an SBO jump M-convex function. Let $z^{(1)}, z^{(2)}, \dots, z^{(\ell)} \equiv r \pmod{2}$ be given and $\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(\ell)}$ be real convex multipliers so that $z := \sum_{k \in [\ell]} \lambda^{(k)} z^{(k)} \equiv r \pmod{2}$. Then $f(z) \leq \sum_{k \in [\ell]} \lambda^{(k)} f(z^{(k)})$.*

Proof. Since we may assume that λ is rational, we may additionally assume that $\lambda_k = 1/\ell$ after duplicating vectors of the convex combination a suitable number of times.

For every $i^* \in [m]$ we exhaustively employ the following procedure: if there exist $z^{(k_1)}, z^{(k_2)}$ with $z_{i^*}^{(k_1)} - z_{i^*}^{(k_2)} \leq -2$ and $z_{i^*}^{(k_2)} - z_{i^*}^{(k_1)} \geq 2$, apply Lemma 22 to replace $z^{(k_1)}$ and $z^{(k_2)}$ with the resulting $z^{(k_1)'}$ and $z^{(k_2)'}$ from this lemma. This preserves the total sum $\sum_{k \in [\ell]} z^{(k)} = \ell z$ and guarantees that $\sum_{k \in [\ell]} f(z^{(k)})$ does not increase. Observe that if such $z^{(k_1)}$ and $z^{(k_2)}$ do not exist, w.l.o.g. all $z_{i^*}^{(k)} \geq z_{i^*}$. As z_{i^*} is the average of $z_{i^*}^{(k)}$, it must hold that $z_{i^*} = z_{i^*}^{(k)}$. Since the application of Lemma 22 does not modify any $z_i^{(k)}$ at i for which $z_i^{(k)} = z_i$ for all $k \in [\ell]$, exhaustive application will eventually result in $z^{(k)'} = z$.

It is left to show that Lemma 22 can be exhaustively applied. To see this, we employ $\sum_{k \in [\ell]} |z_{i^*} - z_{i^*}^{(k)}|$ as progress measure. With respect to the progress made after applying

Lemma 22, it holds that

$$\begin{aligned} & \sum_{k \in [\ell]} |z_{i^*} - z_{i^*}^{(k)'}| - \sum_{k \in [\ell]} |z_{i^*} - z_{i^*}^{(k)}| \\ &= |z_{i^*} - z_{i^*}^{(k_1)'}| + |z_{i^*} - z_{i^*}^{(k_2)'}| - |z_{i^*} - z_{i^*}^{(k_1)}| + |z_{i^*} - z_{i^*}^{(k_2)}| \\ &< 0, \end{aligned}$$

which follows from Lemma 22 that guarantees that $z_{i^*}^{(k_1)}$ and $z_{i^*}^{(k_2)}$ both move strictly closer to z_{i^*} . After applying Lemma 22 exhaustively for all $i^* \in [m]$, we end up with a situation where $z^{(k)'} = z$ for all k and that $\ell f(z) = \sum_{k \in [\ell]} f(z^{(k)'}) \leq \sum_{k \in [\ell]} f(z^{(k)})$. ◀

We note that the lattice-convexity of the domain $\{z \in \mathbb{Z}^m : f_{c,M}(z) < \infty\}$ on $2\mathbb{Z}^m + r$ immediately follows from a generalization of Tutte's characterization of graphs with perfect matchings, see Corollary 31.1a in [46]. Alternatively, it is a special case of Theorem 1.2 in [1]. Having established the convexity of $f_{c,M}$ on the scaled and shifted lattice $2\mathbb{Z}^m + r$, we find that, for a fixed r , the set of z -s with corresponding perfect z -matchings and its optimal objective can be modeled as the points in a convex set in $1 + m$ dimensions.

We define a polyhedron in Definition 24, which models this set within some prescribed bound U and can be interpreted as the epigraph of a convex extension of $f_{c,M}$.

► **Definition 24.** Let $r \in \{0, 1\}^m$ be a remainder vector modulo 2 and $U \in \mathbb{Z}$ an upper bound with polynomial encoding length. We define the polyhedron $P_{r,U}$ to be the convex hull of the points $S_{r,U}$ defined by

$$S_{r,U} := \{(\omega, z) \in \mathbb{R} \times \mathbb{Z}^m : z \equiv r \pmod{2}, \|z\|_\infty \leq U, f_{c,M}(z) \leq \omega\}.$$

That is, $P_{r,U}$ is the Minkowski sum of the polytope which is the convex hull of the points

$$\tilde{S}_{r,U} = \{(f_{r,U}(z), z) \mid z \equiv r \pmod{2}, \|z\|_\infty \leq U, f_{c,M}(z) < \infty, z \in \mathbb{Z}^m\}$$

and the recession cone $\{(\lambda, \mathbf{0}) \mid \lambda \geq 0\}$.

That is, $S_{r,U}$ contains points (ω, z) for which there is a perfect z -matching with cost at most ω , restricted to remainder vector r modulo 2 and a bounding box. Note that we do not have an explicit compact outer description of $P_{r,U}$ and are not aware of whether a polyhedron that models $f_{c,M}$ on $2\mathbb{Z}^m + r$ with polynomially many facets that are efficiently computable exists. However, such a description is not needed, as known integer programming algorithms, e.g. [45, 33], can solve integer programs when the feasible region is a bounded polyhedron equipped with a strong separation oracle. Similar as to what was observed in [49] for a slightly different polytope, the known methods for optimizing generalized matchings, imply that such a strong separation oracle running in polynomial time exists as a consequence of the ellipsoid method.

► **Lemma 25.** The strong separation problem for $P_{r,U}$ can be solved in polynomial time. That is, given a fractional $\tilde{q} \in \mathbb{Q}^{1+m}$ one can efficiently decide whether $\tilde{q} \in P_{r,U}$ and if not, yield a separating hyperplane $\alpha \in \mathbb{Q}^{1+m}$ such that $\alpha^\top \tilde{q} > \alpha^\top q$ for all $q \in P_{r,U}$.

Proof. By construction, $P_{r,U}$ is a polyhedron with its vertices having z bounded by U and consequently additionally having suitably bounded ω . As its recession cone is similarly well-behaved, we can apply a deep result following from the ellipsoid method, Theorem 6.4.9 in [27], which shows that the strong separation problem can efficiently be solved if one can optimize an arbitrary objective \tilde{c} over $P_{r,U}$ in polynomial time.

XX:20 Parameterized Algorithms for Matching IPs with Additional Rows and Columns

For completeness, we explicitly show how one can optimize over $P_{r,U}$. It suffices to solve the following problem

$$\min \{ \tilde{c}^\top (\omega, z) \mid (\omega, z) \in P_{r,U} \}.$$

If the first component of \tilde{c} is negative, the problem is unbounded in the $(1, \mathbf{0})$ direction if and only if it is feasible. Therefore, this case can be reduced to the same problem with $\tilde{c} = \mathbf{0}$ and we may further assume that the first component of \tilde{c} is nonnegative. In this case, it suffices to optimize \tilde{c} over the set $\tilde{S}_{r,U}$, which contains all vertices of $P_{r,U}$. Expanding the definitions of $\tilde{S}_{r,U}$ and $f(z)$ yields

$$\min \{ \tilde{c}^\top (c^\top x, z) \mid z \equiv r \pmod{2}, \|z\|_\infty \leq U, Mx = z, x \in \mathbb{Z}_{\geq 0}^n, z \in \mathbb{Z}^m \}.$$

Substituting $z = Mx$ yields

$$\min \{ \tilde{c}^\top (c^\top x, Mx) \mid Mx \equiv r \pmod{2}, Mx \leq U\mathbf{1}, x \in \mathbb{Z}_{\geq 0}^n \}.$$

This is an instance of a polynomial time solvable matching generalization ([15], Theorem 36.5 in [46]). For completeness, we further reduce it with an additional step. Observe that after defining $\tilde{r} := (U\mathbf{1} + r) \bmod 2$, we can reformulate the constraints on Mx by equating Mx to $U\mathbf{1} - (\tilde{r} + 2Is)$ for some auxiliary integer variables $s \in \mathbb{Z}_{\geq 0}^m$. This yields the generalized matching problem

$$\min \left\{ (\tilde{c}^\top (c^\top, M))^\top x \mid Mx + 2Is = U\mathbf{1} - \tilde{r}, (x, s) \in \mathbb{Z}_{\geq 0}^{n+m} \right\}.$$

By Theorem 1, this problem can be solved in polynomial time. ◀

We have now gathered the necessary ingredients to prove Theorem 2.

Proof of Theorem 2. Consider an instance of ILP (T). We first perform some preprocessing steps. Using the proximity result from Cook et al. [10], we may assume that the variable bounds e, l, g, u are finite and have polynomial encoding length. Now apply Lemma 5 to transform the ILP. The resulting problem is of the form

$$\min \{ a^\top y + c^\top x \mid Ty + Mx = b, \mathbf{0} \leq y \leq g, y \in \mathbb{Z}^h, x \in \mathbb{Z}_{\geq 0}^n \},$$

where M is the incidence matrix of a simple graph and the bounds g are finite with polynomial encoding length. Since this is a polynomial transformation of the bounded problem, which has an optimal objective with polynomial encoding length, an optimal solution to this reduced ILP can be found in a polynomial number of binary search steps solving a feasibility problem of the form

$$\{ a^\top y + c^\top x \leq \omega^*, Ty + Mx = b, \mathbf{0} \leq y \leq g, y \in \mathbb{Z}^h, x \in \mathbb{Z}_{\geq 0}^n \}. \quad (1)$$

It now suffices to show that the feasibility of ILP (1) can be determined in FPT time. To do so, we guess the value of $y \bmod 2$ by substituting $y = 2v + t$ for a set of variables $v \in \mathbb{Z}^h$ and guessing all 2^p remainder vectors $t \in \{0, 1\}^p$ as done in [11]. The ILP (1) is feasible if and only if at least one of these subproblems is feasible. After fixing t , our subproblem is equivalent to

$$\{ a^\top (2v + t) + c^\top x \leq \omega^*, T(2v + t) + Mx = b, \mathbf{0} \leq 2v + t \leq g, v \in \mathbb{Z}^h, x \in \mathbb{Z}_{\geq 0}^n \} \quad (2)$$

We show that this may equivalently be posed as finding an integer vector v in a particular convex body. For this purpose, define the affine function F by

$$F(v) := \begin{pmatrix} \hat{\omega}(v) \\ \hat{z}(v) \end{pmatrix} := \begin{pmatrix} -2a^\top v - a^\top t + \omega^* \\ -2Tv - Tt + b \end{pmatrix},$$

define the requirement R that there exists a suitably low costed perfect $\hat{z}(v)$ -matching by

$$R := \{v \mid c^\top x \leq \hat{\omega}(v), Mx = \hat{z}(v), v \in \mathbb{Z}^p, x \in \mathbb{Z}_{\geq 0}^n\},$$

and define the box B expressing the variable bounds on v by

$$B := \{v \in \mathbb{R}^p : \mathbf{0} \leq 2v + t \leq g\}.$$

In this way, we may look for $v \in R \cap B$ to determine the feasibility of ILP (2). In fact, if such v exists, it can straightforwardly be completed to a solution to ILP (1) by setting $y = 2v + t$ and computing a minimum cost perfect $\hat{z}(v)$ -matching x .

Since we fixed our remainder modulo 2, the function F satisfies that $\hat{z}(v) \equiv r \pmod{2}$ for all $v \in \mathbb{Z}^p$ where $r := (-Tt + d) \pmod{2}$, which will enable us to find such v if it exists. Set U to a value with polynomial encoding length that is sufficiently large so that $\|\hat{z}(v)\|_\infty \leq U$ for all $v \in B$.

▷ **Claim.** $R \cap B = F^{-1}(P_{r,U}) \cap B \cap \mathbb{Z}^p$, where F^{-1} is the pre-image $\{v \in \mathbb{R}^h : F(v) \in P_{r,U}\}$.

Proof. An integral $v \in F^{-1}(P_{r,U}) \cap B$ is within the bounds B and satisfies that $(\hat{\omega}(v), \hat{z}(v))$ is a convex combination $\sum_{k \in [\ell]} \lambda^{(k)}(\omega^{(k)}, z^{(k)})$ of points in $S_{r,U}$. Lemma 23 shows that there exists a perfect $\hat{z}(v)$ -matching x with $c^\top x \leq f_{c,M}(\hat{z}(v)) \leq \sum_{k \in [\ell]} \lambda^{(k)} f_{c,M}(z^{(k)}) \leq \sum_{k \in [\ell]} \lambda^{(k)} \omega^{(k)} = \hat{\omega}(v)$. This witnesses that $v \in R$. The inclusion in the other direction makes use of the fact that $v \in R \cap B$ satisfies that $\|\hat{z}(v)\|_\infty \leq U$. The perfect $\hat{z}(v)$ -matching x witnessing $v \in R$ shows that $f_{c,M}(\hat{z}(v)) \leq c^\top x \leq \hat{\omega}(v)$ and thus that $F(v) = (\hat{\omega}(v), \hat{z}(v)) \in S_{r,U} \subseteq P_{r,U}$. ◁

An integral point in the polytope $F^{-1}(P_{r,U}) \cap B \cap \mathbb{Z}^p$ can be found in FPT time by an integer programming algorithm such as the algorithm by Reis and Rothvoss [45]⁴. For this, it is essential that one can implement a strong separation oracle over this polytope that runs in polynomial time, which follows from Lemma 25. To see this, first observe that testing whether a given fractional \tilde{v} is in $F^{-1}(P_{r,U}) \cap B$ is equivalent to testing whether $F(\tilde{v}) \in P_{r,U}$ and $\tilde{v} \in B$. Hyperplanes separating \tilde{v} from B are trivial to construct. Finally, a hyperplane $\alpha \in \mathbb{Q}^{1+m}$ separating $F(\tilde{v})$ from $P_{r,U}$ yields a hyperplane separating \tilde{v} from $F^{-1}(P_{r,U})$ as $(\alpha^\top L)\tilde{v} > (\alpha^\top L)v$ for $v \in F^{-1}(P_{r,U})$ where $L \in \mathbb{Z}^{(1+m) \times h}$ is the linear transformation that comprises the affine F .

If we find a feasible ILP, the integer programming algorithm can yield us a corresponding satisfying v . ◀

We note that the additional multiplicative 2^p contribution to the running time of the algorithm as a result of guessing the parity of y is asymptotically small compared to the running time of the current state-of-the-art IP algorithm by Reis and Rothvoss [45], which for our application may have a running time of $(\log p)^{\mathcal{O}(p)}$ times a polynomial of the input encoding length.

⁴ This state-of-the-art IP algorithm based on Dadush work [12] is randomized. However, the algorithm by Lenstra [33] shows that the integer programming problem can be solved deterministically, yielding a slower, but deterministic FPT algorithm.

5 Few arbitrary constraints

We now turn our focus on complicating constraints in ILP (W) instead of complicating variables. As mentioned, if the coefficients of the constraint matrix are encoded in binary, ILP (W) can encode the NP-hard subset sum problem. Therefore, we develop an algorithm scaling polynomially in $\Delta, \|c\|_\infty$ for a fixed h .

► **Theorem 3.** *ILP (W) can be solved with a randomized XP time algorithm in terms of h . More specifically, it can be solved in time $\tilde{\mathcal{O}}(\|c\|_\infty \log^2 \|c\|_\infty \cdot \Delta^h \log^5 \Delta) \cdot \mathcal{O}(n)^{10+h} \cdot \mathcal{O}(\Delta h m)^{8h+h^2} + \mathcal{O}(nm)$.*

To complement Theorem 3, we show that solving ILP (W) is W[1]-hard when parameterized by h even when $\Delta = 1$ in Theorem 4.

The essence of solving ILP (T) in XP time when W and c are encoded in unary is reducing the problem to a perfect b -matching problem with additional constraints and a polynomially bounded right-hand side b by using a bound on the Graver complexity of the constraint matrix and the Graver augmentation framework of Theorem 12. This perfect b -matching problem can then safely be pseudo-polynomially expanded to a perfect matching problem using the construction of Proposition 20. Finally, the h constraints can be condensed into one and the resulting problem can be solved using a known randomized algorithm for the exact matching problem.

As the key step in this algorithm is bounding b , we first discuss the Graver complexity of the constraint matrix and bound the proximity, i.e., the distance between optimal solutions to the LP relaxation of ILP (W) and nearby optimal integer solutions. Using Theorem 13 and the proof of a lemma from the study of n -fold ILPs [16], we obtain the bound given in Section 2.3.

► **Corollary 14.**

$$g_\infty \left(\binom{W}{M} \right) \leq 2(2\Delta h(2m+1) + 1)^h = \mathcal{O}(\Delta h m)^h.$$

Proof. Following the proof of Lemma 3 in [16] using the 1-norm bound on $\mathcal{G}(M)$ from Theorem 13, and then measuring the Graver complexity in terms of the ∞ -norm yields the result. ◀

Note that an instance of ILP (W) may be preprocessed so that $m \leq 2n$. Since every column has at most two nonzero entries, such preprocessing step which removes potential zero rows from M can be executed in $\mathcal{O}(nm)$ time. Except for the running time in Theorem 3, we will present the results of this section assuming $m = \mathcal{O}(n)$.

As shown in Proposition 18, the bound from Corollary 14 is asymptotically tight for constant h .

► **Proposition 18.** *For any fixed positive integer h , there exists an infinite family of constraint matrices $\binom{W}{M} \in \{0, 1, \Delta\}^{(h+m) \times n}$ of ILP (W) with Graver basis elements that have an ∞ -norm of order $\Omega(\Delta m)^h$.*

Proof. We construct a matrix that essentially matches the lower bound example

$$\begin{pmatrix} \Delta' & -1 & & & \\ & \Delta' & & & \\ & & \ddots & -1 & \\ & & & \Delta' & -1 \end{pmatrix}$$

in [30] for some large Δ' by using smaller coefficients of size $\Delta \leq \Delta' = \Delta k$ in the matrix W for some fixed $k \in \mathbb{Z}_{\geq 1}$. First, define the auxiliary $k \times k$ block U_z for $z \in \{0, 1\}$ by

$$U_z = \begin{pmatrix} -z & & & & & \\ 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & 1 & -1 \end{pmatrix} \quad (3)$$

which will ensure that the variables associated with this block have equal value. Then construct the constraint matrix

$$\begin{pmatrix} W \\ \bar{M} \end{pmatrix} = \begin{pmatrix} \Delta \mathbf{1}^\top & -1 & & & & \\ & \Delta \mathbf{1}^\top & -1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ \hline \bar{U}_0 & & & & \Delta \mathbf{1}^\top & -1 \\ & e_1 & U_1 & & & \\ & & & e_1 & \ddots & \\ & & & & \ddots & \ddots \\ & & & & & e_1 & U_1 \end{pmatrix}, \quad (4)$$

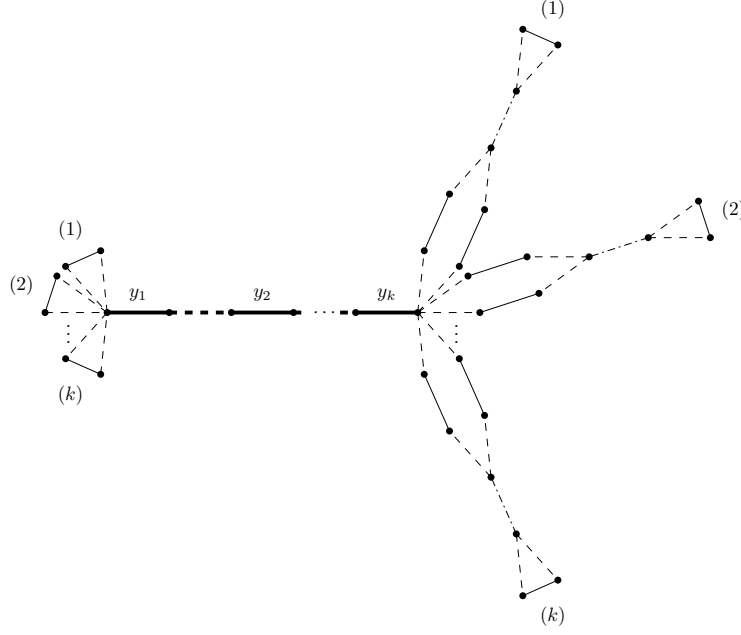
where e_1 denotes the unit vector $(1, 0, \dots, 0) \in \mathbb{Z}^k$.

Again, all kernel elements have last entry with absolute value equal to $(\Delta k)^h$ times the first entry. A nonzero kernel element with first entry 1 exists. Since $m = hk$, we obtain $k = m/h$ and the stated lower bound. Note that the constructed matrices use elements from $\{-1, 0, 1, \Delta\}$. However, they can be adapted to use only elements from $\{0, 1, \Delta\}$ using the ideas from Lemma 5, yielding the same asymptotic lower bound. ◀

As Theorem 12 requires the ILP to be bounded, we employ the proximity bound of $n \cdot \mathcal{O}(\Delta hm)^h$ from Corollary 15. Before showing how these two results allow us to reduce ILP (W) to finding solutions to a constrained b -matching problem with small b , we show that the proximity upper bound is tight when $n = \Theta(m)$ and h is constant.

► **Proposition 19.** *For any fixed positive integer h , there exists an infinite family of constraint matrices $\begin{pmatrix} W \\ \bar{M} \end{pmatrix} \in \{0, 1, \Delta\}^{(h+m) \times n}$ of ILP (W) and vertex LP relaxation solutions to the corresponding ILP (W) such that the nearest integral solution is at ∞ -norm distance $\Omega(m) \cdot \Omega(\Delta m)^h$.*

Proof. To show this, we use a large part of the matrix in Equation (4) and replace one block with the incidence matrix of a gadget graph. For this let $k \in \mathbb{Z}_{\geq 1}$ and construct the b -matching instance shown in Figure 7.



■ **Figure 7** A b -matching gadget showing that the 1-norm proximity for b -matching can be of order $\Omega(|V|^2)$. All vertices have $b_v = 1$ except for the vertices incident to y_j , which have $b_v = k$. The only integral solution assigns a value of zero to the dashed edges, a value of k to the edges y_j and a value of 1 to the other edges. On the other hand, a vertex fractional solution exists which assigns a value of zero to the full thick and dash-dotted edges, a value of k to the dashed thick edges, and $\frac{1}{2}$ to all other edges. The size of the gadget is given by $|V| = 12k$ and $|E| = -1 + 15k$.

Write the gadget perfect b -matching problem from Figure 7 as the ILP $\tilde{M}x = \tilde{b}$ for nonnegative integral x . Denote $\tilde{M} = (\tilde{M}_1 \quad \tilde{M}_2) \in \{0, 1\}^{12k \times (-1+15k)}$ where the columns of \tilde{M}_2 correspond to the variables y_1, \dots, y_k , which may be assigned value 0 in a vertex LP relaxation solution, but are forced to take value k in an integral solution. Finally, construct the system

$$\left(\begin{array}{cc|cccc} \Delta \mathbf{1}^\top & -1 & & & & \\ & & \Delta \mathbf{1}^\top & -1 & & \\ & & & & \ddots & \ddots \\ & & & & & \Delta \mathbf{1}^\top & -1 \\ \hline \tilde{M}_1 & \tilde{M}_2 & & & & & \\ & & e_1 & U_1 & & & \\ & & & & e_1 & \ddots & \\ & & & & & \ddots & \ddots \\ & & & & & & e_1 & U_1 \end{array} \right) x = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \tilde{b} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \quad x \geq \mathbf{0}$$

using the $k \times k$ auxiliary block U_1 from Equation (3). A vertex fractional solution exists with the last component of x being zero, whereas the only integral solution must assign $k \cdot (\Delta k)^h$ to this variable. We have that $m = 12k + (h-1)k$, i.e., $k = \Omega(m)$ and the lower bound follows. It can be verified that the -1 entries can be eliminated through the use of Lemma 5. ◀

Both Corollaries 14 and 15 can be used in conjunction with Theorem 12 to reduce the

problem of solving ILP (W) to computing a number of Graver-best steps.

► **Lemma 26.** *The ILP (W) can be solved by performing $\mathcal{O}(h^2 n \log(\|c\|_\infty \Delta h n) \log(\Delta h n))$ many Graver-best oracle queries for instances of ILP (W) with $n' = \mathcal{O}(n)$, $\|c'\|_\infty = \mathcal{O}(\|c\|_\infty)$ and W' having the same entries as W up to the insertion of an identity matrix; computing the LP relaxation of an asymptotically equally sized instance of ILP (W); and performing a preprocessing step which takes $\mathcal{O}((h^2 + m)n)$ time.*

Proof. We solve the LP relaxation of ILP (W) to obtain an optimal rational solution y^* . Using this and Corollary 15, we may impose bounds on x around y^* so that $\|u - l\|_\infty = n \cdot \mathcal{O}(\Delta h m)^h$. Then we translate the ILP with $x' = x - l$ so that $l' = \mathbf{0} \leq u' = u - l$. Note that we may assume that $\|(d, b)\|_\infty \leq \|A\|_\infty \|u\|_\infty \leq \Delta n \cdot n \cdot \mathcal{O}(\Delta h m)^h = \Delta n^2 \cdot \mathcal{O}(\Delta h m)^h$ or the ILP is infeasible.

If an initial feasible solution is not known, we can find such solution by optimizing over a modified instance of ILP (W) with a given feasible solution such as done in [32]. This can be accomplished by building an instance of ILP (W) with

$$\begin{pmatrix} W' \\ M' \end{pmatrix} = \begin{pmatrix} W & I & \mathbf{0} \\ M & \mathbf{0} & I \end{pmatrix}$$

for $h + m$ new variables s corresponding to the right hand sides (d, b) . The objective corresponding to the original matching variables is set to $\mathbf{0}$. The variable bounds and objective coefficient corresponding to a variable s_j are set to $0 \leq s_j \leq (d, b)_j$ and 1 if $(d, b)_j \geq 0$ and $(d, b)_j \leq s_j \leq 0$ and -1 otherwise. All other parameters are left the same as for the restricted and translated ILP. As the lower bounds for x are 0, it is clear that $(\mathbf{0}, d, b)$ is a feasible solution to this auxiliary ILP unless the original ILP was trivially infeasible. Additionally, one can find a feasible solution x to the original ILP (W) by projecting the first n components of a solution to the auxiliary instance that has optimal value 0. If such solution does not exist, the original ILP is infeasible. The auxiliary instance has increased size $\|u' - l'\|_\infty = \Delta n^2 \cdot \mathcal{O}(\Delta h m)^h$, $n' = \mathcal{O}(h + n)$ and $\|c'\|_\infty = \mathcal{O}(\|c\|_\infty)$. Note that we can assume that $h \leq n$ after removing redundant constraints through Gaussian elimination in $\mathcal{O}(h^2 n)$ arithmetic operations.

We now inspect the number of Graver-best steps that the augmentation framework needs to compute for both finding the initial solution and improving it with respect to c . The difference in objective $c^\top x - c^\top x^*$ is at most $\|c\|_1 \|u - l\|_\infty = \|c\|_1 \cdot \Delta n^2 \cdot \mathcal{O}(\Delta h m)^h$. For this reason, the Graver augmentation framework needs to compute at most $\mathcal{O}(n \log(\Delta n^2 \cdot (\Delta h m)^h) \log(\|c\|_1 \Delta n^2 \cdot (\Delta h m)^h)) = \mathcal{O}(h^2 n \log(\|c\|_\infty \Delta h n) \log(\Delta h n))$ many Graver-best steps to find an optimal ILP solution by Theorem 12. ◀

We now proceed to reduce the problem of finding a Graver-best step to a constrained perfect matching problem. To do so, we first reduce to a constrained b -matching problem and then employ the pseudo-polynomial reduction to perfect matching from Proposition 20.

► **Lemma 27.** *Finding a Graver-best step for ILP (W) can be reduced to solving another instance of ILP (W) where $l' = \mathbf{0}$, $u' = \mathbf{1}$, $d' = d$, $b' = \mathbf{1}$, $\Delta' = \Delta$ and $n' = n^2 \cdot \mathcal{O}(\Delta h m)^{2h}$, $m' = n \cdot \mathcal{O}(\Delta h m)^h$, and M is the incidence matrix of a simple graph. This reduction can be performed in output-linear time.*

Proof. Corollary 14 shows that we may restrict to $\|l\|_\infty, \|u\|_\infty = \mathcal{O}(\Delta h m)^h$. We apply the reductions from Lemma 5. This yields an equivalent instance where b is bounded by $\|b\|_1 = \mathcal{O}(\|u - l\|_1) = \mathcal{O}(n \|u - l\|_\infty) = n \cdot \mathcal{O}(\Delta h m)^h$, and M is the incidence matrix of the

graph $G(M)$. We now further reduce to the case where $b = \mathbf{1}$ by using the graph $G(M)_b$ from Proposition 20. To accomplish this, let M' be the incidence matrix of $G(M)_b$ and construct an ILP of the form (W) with $b = \mathbf{1}$ and the restrictions from Lemma 5. For every copy j' of an edge j of $G(M)$, copy the corresponding column $W_{\cdot,j}$ and use it as the column for j' in W' . Copy the objective coefficients in the same way. This builds the correspondence $x_j = x_j^{(1)} + \dots + x_j^{(k)}$, where $x_j^{(1)}, \dots, x_j^{(k)}$ are the $k = b_{i_1} \cdot b_{i_2}$ copies of the variable x_j that corresponds with the edge connecting vertices i_1 and i_2 in $G(M)$. These problems are equivalent by Proposition 20. The graph $G(M)_b$ has $m' = \|b\|_1 = n \cdot \mathcal{O}(\Delta h m)^h$ vertices and $n' \leq m'^2$ edges. \blacktriangleleft

Finally, all h constraints can be condensed into a single constraint. This can be accomplished by representing the constraints in base B for a sufficiently large B , as done for special cases in [5, 44] (for $h = 2$ and $W \in \{0, 1\}^{h \times n}$ respectively).

► **Lemma 28.** *An instance of ILP (W) with*

- $W \in \mathbb{Z}_{\geq 0}^{h \times n}$,
- M being the incidence matrix of a simple graph,
- $l = \mathbf{0}, u = \mathbf{1}, b = \mathbf{1}$,

can be reduced to itself with $h' = 1, W' \in \mathbb{Z}_{\geq 0}^{1 \times n}$ in input+output linear time by only increasing $\Delta' = \Delta^h \cdot \mathcal{O}(m)^{h-1}$ and preserving c, M, l, u, b .

Proof. We condense all constraints of W into one. Note that $(m/2)\Delta$ is a trivial upper bound on $W_{i,\cdot}x$ for a perfect matching x . Therefore, we may assume that $0 \leq d_i \leq (m/2)\Delta$ for all i or the ILP is trivially infeasible. By working base $B := (m/2)\Delta + 1$, we observe that the h constraints

$$Wx = d$$

are equivalent to the single constraint

$$\left(\sum_{i \in [h]} B^{i-1} W_i \right) x = \sum_{i \in [h]} B^{i-1} d_i.$$

\blacktriangleleft

We can now employ the randomized exact matching algorithm by Mulmuley, Vazirani and Vazirani [39] to solve instances resulting from Lemma 28. For the sake of completeness, we present an algorithm in Lemma 29, which closely follows their ideas.

► **Lemma 29.** *A constrained minimum cost perfect matching problem, i.e., an instance of ILP (W) with*

- $c \geq \mathbf{0}, W \in \mathbb{Z}_{\geq 0}^{1 \times n}, h = 1$,
- M being the incidence matrix of a simple graph,
- $l = \mathbf{0}, u = \mathbf{1}, b = \mathbf{1}$,

can be solved with a randomized algorithm in

$$\|c\|_\infty \Delta n m^7 \log(\Delta m) \log(\|c\|_\infty \Delta n) \cdot (\log \log(\|c\|_\infty \Delta n))^{\mathcal{O}(1)}$$

time. In this case, an optimal solution is not computed.

Proof. Assign every edge j a random weight $w_j \leftarrow (nm + 1)c_j + Z_j$ where Z_j is sampled independently and uniformly from $\{1, 2, \dots, 2n\}$. By doing so, the isolation lemma [39] ensures that with probability at least $\frac{1}{2}$, there exists a unique minimum w -weighted perfect matching

satisfying the additional $Wx = d$ constraint, if such matching exists. To check for the existence of such unique minimum matching, we compute the Pfaffian of the $m \times m$ skew-symmetric Tutte matrix D where $D_{i_1, i_2} = 0$ if i_1 and i_2 are non-adjacent and $D_{i_1, i_2} = 2^{w_j} \cdot X^{W_{1j}}$ if j connects $i_1 < i_2$, where X is some indeterminate. It is known that if there is a unique minimum weight constrained perfect matching, the Pfaffian will contain a monomial $C \cdot X^{d_1}$ with a nonzero coefficient C [39]. In fact, the minimum weight perfect matching satisfying the additional constraint has a weight ω which is the least integer for which 2^ω divides C . As a perfect matching contains exactly $m/2$ edges, we have that the sum over Z_j where j ranges over this perfect matching is at most nm . Therefore, the minimum c -cost of the perfect matching can be recovered by computing the quotient of ω by $nm + 1$. If there is no constrained perfect matching, the coefficient C will always be zero. Therefore, we can compute the optimal objective of the ILP by computing the coefficient of X^{d_1} .

The Pfaffian of the $m \times m$ matrix can be computed in a division free way with m^4 ring operations [38]. As the ring elements can be large, these ring operations are expensive. A close inspection of the algorithm [38], see [36], reveals that the coefficients of the monomials in the computation can be bounded by $\mathcal{O}(m^m (2^{(nm+1)\|c\|_\infty + 2n})^m) = \mathcal{O}(2^{m \log m + 2nm + (nm+1)m\|c\|_\infty})$ and thus have encoding length bounded by $\bar{l} = \mathcal{O}(\|c\|_\infty nm^2)$. As we are solely interested in the coefficient C of the term $C \cdot X^{d_1}$, monomials containing a power of X greater than d_1 may be discarded during the computation of the Pfaffian. By again bounding $d_1 \leq (m/2)\Delta$, this shows that the polynomials have degree at most $\bar{d} := (m/2)\Delta$. We can apply an FFT based fast univariate polynomial multiplication algorithm (Corollary 8.27 [48]) on this to obtain a multiplication time of

$$\begin{aligned} & \bar{d}(\log \bar{d} + \bar{l}) \cdot \log(\bar{d}(\log \bar{d} + \bar{l})) \cdot (\log \log(\bar{d}(\log \bar{d} + \bar{l})))^{\mathcal{O}(1)} \\ &= \bar{l} \cdot \bar{d} \log \bar{d} \cdot \log(\bar{l} \cdot \bar{d} \log \bar{d}) \cdot (\log \log(\bar{l} \cdot \bar{d} \log \bar{d}))^{\mathcal{O}(1)} \\ &= \|c\|_\infty \Delta nm^3 \log(\Delta m) \cdot \log(\|c\|_\infty \Delta nm^3 \log(\Delta m)) \cdot (\log \log(\|c\|_\infty \Delta nm^3 \log(\Delta m)))^{\mathcal{O}(1)} \\ &= \|c\|_\infty \Delta nm^3 \log(\Delta m) \log(\|c\|_\infty \Delta n) \cdot (\log \log(\|c\|_\infty \Delta n))^{\mathcal{O}(1)}. \end{aligned}$$

Here, we have slightly loosely estimated $\bar{d}(\log \bar{d} + \bar{l}) \leq \bar{l} \cdot \bar{d} \log \bar{d}$ in order to improve readability. Additions can be performed in less asymptotic time. Multiplying with the $\mathcal{O}(m^4)$ arithmetic operations that need to be performed yields the stated running time. ◀

We can now combine the findings from this section to derive the claimed randomized XP time algorithm.

Proof of Theorem 3. We reduce solving ILP (W) to solving $\mathcal{O}(h^2 n \log(\|c\|_\infty \Delta hn) \log(\Delta hn))$ Graver-best oracle problems using Lemma 26. Then apply Lemmata 27 and 28. We find that we need to solve instances of size

$$\begin{aligned} m' &= n \cdot \mathcal{O}(\Delta hm)^h, \\ n' &= \mathcal{O}(m'^2), \\ \Delta' &= \Delta^h \cdot \mathcal{O}(m')^{h-1}, \\ \|c'\|_\infty &= \mathcal{O}(\|c\|_\infty). \end{aligned}$$

Finally, apply Lemma 29 to find a running time of

$$\begin{aligned}
& \|c\|_\infty \Delta' n' m'^7 \log(\Delta' m') \log(\|c\|_\infty \Delta' n') \cdot (\log \log(\|c\|_\infty \Delta' n'))^{\mathcal{O}(1)} \\
&= \Delta^h \cdot \mathcal{O}(m')^{h+8} \cdot \|c\|_\infty \log(\mathcal{O}(m')^h \cdot \Delta^h) \log(\|c\|_\infty \cdot \mathcal{O}(m')^{h+1} \cdot \Delta^h) \\
&\quad \cdot (\log \log(\|c\|_\infty \cdot \mathcal{O}(m')^{h+1} \cdot \Delta^h))^{\mathcal{O}(1)} \\
&= \Delta^h \cdot (n \cdot \mathcal{O}(\Delta h m)^h)^{h+8} \cdot \tilde{\mathcal{O}}(\|c\|_\infty \log \|c\|_\infty \cdot \log^2 \Delta) \cdot h^{\mathcal{O}(1)} \\
&= \tilde{\mathcal{O}}(\|c\|_\infty \log \|c\|_\infty \cdot \Delta^h \log^2 \Delta) \cdot \mathcal{O}(n)^{8+h} \cdot \mathcal{O}(\Delta h m)^{8h+h^2}
\end{aligned}$$

for finding the objective improvement of a single Graver-best step. Observe that polynomial factors in h are subsumed by an exponentially growing factor $\mathcal{O}(1)^h$. Note that we must obtain the Graver-best step x from Definition 11 in order to be able to apply the augmentation framework. To obtain this step, we perform a binary search on the value of each variable of x . We can recover a value of a variable x_j in a Graver-best step by restricting the domain of x_j to $[\bar{l}_j, \underline{u}_j]$ and seeing if this still yields an optimal solution with the same objective value. In this way, for each $j \in [n]$ only logarithmically many bound pairs $\bar{l}_j, \underline{u}_j$ need to be tried to fix the values of x_j and recover a solution. Using the imposed bounds on $\|u_j - l_j\|_\infty = \mathcal{O}(\Delta h m)^h$, this requires solving the Graver-best subproblem an additional $L = \mathcal{O}(n \cdot h \log(\Delta h m))$ many times. Now, it must be observed that a randomized algorithm is called a non-constant L number of times, of which each execution must be successful. In order to maintain a constant success probability, we must therefore attempt to solve each subproblem $\log L$ times. This yields a total running time of

$$\begin{aligned}
& nh \log(\Delta h m) \log(nh \log(\Delta h m)) \cdot \tilde{\mathcal{O}}(\|c\|_\infty \log \|c\|_\infty \cdot \Delta^h \log^2 \Delta) \cdot \mathcal{O}(n)^{8+h} \cdot \mathcal{O}(\Delta h m)^{8h+h^2} \\
&= \tilde{\mathcal{O}}(\|c\|_\infty \log \|c\|_\infty \cdot \Delta^h \log^3 \Delta) \cdot \mathcal{O}(n)^{9+h} \cdot \mathcal{O}(\Delta h m)^{8h+h^2}
\end{aligned}$$

to compute a Graver-best step.

To solve the original complete instance of ILP (W), we must compute as many as $K = \mathcal{O}(h^2 n \log(\|c\|_\infty \Delta h n) \log(\Delta h n))$ Graver-best steps. This yields a total running time of

$$\begin{aligned}
& h^2 n \log(\|c\|_\infty \Delta h n) \log(\Delta h n) \log(h^2 n \log(\|c\|_\infty \Delta h n) \log(\Delta h n)) \\
&\quad \cdot \tilde{\mathcal{O}}(\|c\|_\infty \log \|c\|_\infty \cdot \Delta^h \log^3 \Delta) \cdot \mathcal{O}(n)^{9+h} \cdot \mathcal{O}(\Delta h m)^{8h+h^2} \\
&= \tilde{\mathcal{O}}(\|c\|_\infty \log^2 \|c\|_\infty \cdot \Delta^h \log^5 \Delta) \cdot \mathcal{O}(n)^{10+h} \cdot \mathcal{O}(\Delta h m)^{8h+h^2}.
\end{aligned}$$

for solving all the required Graver-best oracle instances. After having preprocessed the instance as done in the proof of Lemma 26, which requires $\mathcal{O}((h^2 + m)n)$ arithmetic operations, solving the LP relaxation can be done within a running time that is dominated by the rest of the algorithm [47]. \blacktriangleleft

Observe that, by using the Graver augmentation framework, Theorem 13, Lemma 5 and a polynomial time algorithm for minimum cost perfect matching, one can rederive the fact that the generalized matching problem is in P, Theorem 1. Most of the steps taken in this reduction process from Lemma 5 reduce to the steps as listed by Schrijver [46]. The difference lies in that the application of a sensitivity result and minimum cost circulation algorithm are replaced by the Graver augmentation framework.

Instead of directly solving the constrained perfect matching problem via Lemma 29, we may further reduce the problem to the 0/1-weighted exact matching problem, which is to find a perfect matching with a target weight, assuming all edges have weight 0 or

1. This can be done by splitting the unary-encoded coefficients in W into multiple binary coefficients as shown in Lemma 30. This generalizes the reduction used in [44]. As the lemma additionally reveals that the $W[1]$ -hardness of solving ILP (W) persists even when we restrict to coefficients bounded by a constant Δ in Theorem 4, we provide an explicit proof.

► **Lemma 30.** *An instance of ILP (W) with finite l, u and $W \in \mathbb{Z}_{\geq 0}^{h \times n}$ can be reduced to itself in output-linear time where*

- $W' \in \{0, 1\}^{h' \times n'}$,
- $\|l'\|_1 = \mathcal{O}(\Delta \|l_1\|), \|u'\|_1 = \mathcal{O}(\Delta(\|u\|_1 + \|u - l\|_1))$,
- $n', m' = \mathcal{O}(\Delta n)$,
- $h' = h$,
- $d' = d$,
- $\|b'\|_1 = \mathcal{O}(\|b\|_1 + \Delta \|u\|_1)$,
- c' contains the same entries as c up to the insertion of zeros.

In addition, the following properties of an instance are preserved:

- M is the incidence matrix of a simple graph,
- $l = \mathbf{0}, u = \mathbf{1}, b = \mathbf{1}$.

Proof. We describe a reduction step to reduce the coefficients of W that are larger than 1 in a given column j by one. For this purpose, subdivide x_j into three variables $x_j^{(1)}, x_j^{(s)}, x_j^{(2)}$. In essence, the coefficients of W are reduced by splitting and distributing them over $x_j^{(1)}$ and $x_j^{(2)}$ as shown in Figure 8.

$$\begin{array}{c|ccc|c}
 \begin{array}{c} c_j \\ \hline W_{\cdot,j} \\ \hline \mathbf{0} \\ M_{i_1,j} = b_{i_1} \\ \hline \mathbf{0} \\ M_{i_2,j} = b_{i_2} \\ \hline \mathbf{0} \\ \hline [l_j, u_j] \end{array} & \rightarrow & \begin{array}{c} c_j \\ \hline (W_{\cdot,j} - \mathbf{1})_+ \\ \hline \mathbf{0} \\ M_{i_1,j} \\ \hline \mathbf{0} \\ 0 \\ \hline 1 \\ \hline 0 \end{array} & \begin{array}{c} 0 \\ \hline 0 \\ \hline \mathbf{0} \\ 0 \\ \hline \mathbf{0} \\ 1 \\ \hline 1 \end{array} & \begin{array}{c} 0 \\ \hline \min\{W_{\cdot,j}, \mathbf{1}\} \\ \hline \mathbf{0} \\ 0 \\ \hline \mathbf{0} \\ 0 \\ \hline 1 \\ \hline 1 \end{array} \\
 & & \begin{array}{c} [l_j, u_j] \\ \hline [0, u_j - l_j] \\ \hline [l_j, u_j] \end{array} & & \begin{array}{c} = b_{i_1} \\ \\ \\ = b_{i_2} \\ \\ = u_j \\ = u_j \end{array}
 \end{array}$$

■ **Figure 8** Coefficients of the complicating constraints can be reduced by subdividing the variables and distributing the coefficient contributions among the new copies.

Again let $M_{i_1,j}, M_{i_2,j}$ denote the potentially nonzero coefficients with absolute value at most 1 in M of x_j (interpreting $i_1 = i_2$ when there is a single coefficient with absolute value 2). We implement the following modifications:

- Add the constraints $x_j^{(1)} + x_j^{(s)} = u_j, x_j^{(2)} + x_j^{(s)} = u_j$ and domains $x_j^{(1)}, x_j^{(2)} \in [l_j, u_j]$ and $x_j^{(s)} \in [0, u_j - l_j]$ as done in the proof of Lemma 5.
- Replace the terms $M_{i_1,j}x_j$ and $M_{i_2,j}x_j$ with $M_{i_1,j}x_j^{(1)}$ and $M_{i_2,j}x_j^{(2)}$ respectively.
- Assign the objective coefficient c_j to $x_j^{(1)}$ and assign all other new variables an objective coefficient of 0.
- Set the column of W' associated with $x_j^{(s)}$ to $\mathbf{0}$, but use the other two variables to reduce the coefficients in W . In particular, assign $(W_{\cdot,j} - \mathbf{1})_+$ and $\min\{W_{\cdot,j}, \mathbf{1}\}$ to $x_j^{(1)}$ and $x_j^{(2)}$ so that $W_{\cdot,j}x = W_{\cdot,j}x_j^{(1)} = (W_{\cdot,j} - \mathbf{1})_+x_j^{(1)} + \min\{W_{\cdot,j}, \mathbf{1}\}x_j^{(2)}$.

XX:30 Parameterized Algorithms for Matching IPs with Additional Rows and Columns

This reduces the coefficients in the j -th column by one. By exhaustively applying this reduction rule on all columns j , all coefficients of W can be made binary. ◀

This technique can also be used to reduce the objective coefficients to a binary vector when it is given in unary. In this way, we find that solving ILP (W) is equivalent to the 0/1-weighted exact matching problem.

► **Proposition 31.** *If the objective coefficients and constraints are encoded in unary and h is fixed, then solving ILP (W) is polynomially equivalent to the 0/1-weighted exact matching problem.*

Proof. Reduce the problem to finding Graver-best steps using Lemma 26 and for finding such step, perform binary search on the objective. The objective requirement can be encoded as an additional constraint and results in feasibility subproblems with $c = \mathbf{0}$. Then apply Lemmata 27, 28, and 30 successively to find a polynomially equivalent 0/1-weighted exact matching instance. ◀

In this way, a polynomial time deterministic algorithm for 0/1-weighted exact matching yields a deterministic algorithm to solve ILP (W). The question of whether such algorithm exists is a long standing open question.

Additionally, if one is able to solve minimum cost perfect matching with one 0/1 side constraint, Lemmata 26–28 and 30 show that ILP (W) is similarly solvable. However, to the best of our knowledge, the complexity of this problem is unknown. In particular, it suffices to focus on maximum weight perfect matching under a 0/1 budget constraint $\sum_{j \in [n]} w_j x_j \leq B$ for $w \in \{0, 1\}^n$, as one can reduce to this problem by decreasing the costs associated with edges with $w_j = 1$ by a sufficiently large number $(m/2)\|c\|_\infty + 1$ so that the budget constraint is forced to be met with equality.

Having finished the discussion of problems related to ILP (W), we now complement Theorem 3 with the hardness result listed in Theorem 4. For this, we employ the known hardness of ILP with coefficients encoded in unary [13]. This problem was shown to be strongly W[1]-hard parameterized by the number of constraints by Dvorák et al. [13] through a reduction from the multicolored clique problem. In their reduction they use binary indicator variables together with variables taking the values of (the sum of two) numbers that have a one-to-one correspondence with vertices of the graph. For this, they use a Sidon sequence of which the elements may be bounded by $\mathcal{O}(n^2)$. This justifies the restriction on the variables x in Theorem 16 of $\mathbf{0} \leq x \leq u$ where u is in unary.

► **Theorem 16** (Theorem 22 in [13]). *Determining the feasibility of the system*

$$\{Wx = d : x \in \mathbb{Z}_{\geq 0}^n\},$$

where $W \in \mathbb{Z}_{\geq 0}^{h \times n}$ is encoded in unary and $d \in \{0, 1\}^h$, is W[1]-hard parameterized by h . This hardness persists when restricting x to $\mathbf{0} \leq x \leq u$ where u is encoded in unary.

The unary upper bounds on the variables, combined with Lemma 30 allow to derive the W[1]-hardness of solving ILP (W) parameterized by h for the restricted case of a binary constrained perfect matching problem on a bipartite graph for which many width parameters are constant. To arrive at the claimed graph class, we avoid the pseudo-polynomial reduction from Lemma 27 and first model each variable of the ILP with pseudo-polynomially many binary variables, before subdividing the corresponding edges to reduce the coefficients of W .

► **Theorem 4.** *Solving ILP (W) encoded in unary is $W[1]$ -hard parameterized by h . This hardness persists when restricting to M being the incidence matrix of a graph which is the disjoint union of even length cycles and $W \in \{0, 1\}^{h \times n}$, $c = \mathbf{0}$, $l = \mathbf{0}$, $u = \mathbf{1}$, $b = \mathbf{1}$.*

Proof. We reduce an ILP instance

$$\{Wx = d : \mathbf{0} \leq x \leq u, x \in \mathbb{Z}^n\}$$

of Theorem 16 to one of ILP (W). Observe that it can immediately be cast into the form (W) by setting $c = \mathbf{0}$, $M = \mathbf{O}$ and $b = \mathbf{0}$. To arrive at the listed class of incidence matrix, we add u_j redundant quadrangles for every variable x_j as shown in Figure 9.

$$\begin{array}{cccccc} \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \vdots \\ \begin{array}{|c|} \hline W_{\cdot,j} \\ \hline \end{array} & \begin{array}{|c|} \hline W_{\cdot,j} \\ \hline \end{array} & \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \vdots \\ \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \begin{array}{|c|} \hline \mathbf{0} \\ \hline \end{array} & \vdots \\ \rightarrow & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \vdots \\ & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \vdots \\ & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \vdots \\ & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \begin{array}{|c|} \hline 1 \\ \hline \end{array} & \vdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots \\ \begin{array}{|c|} \hline [0, u_j] \\ \hline \end{array} & \begin{array}{|c|} \hline [0, 1] \\ \hline \end{array} & \begin{array}{|c|} \hline [0, 1] \\ \hline \end{array} & \begin{array}{|c|} \hline [0, 1] \\ \hline \end{array} & \begin{array}{|c|} \hline [0, 1] \\ \hline \end{array} & \vdots \end{array} \left. \vphantom{\begin{array}{cccccc} \end{array}} \right\} u_j \text{ times}$$

■ **Figure 9** A redundant quadrangle can be added for each variable to ensure that M is an incidence matrix of a graph.

That is, we remove x_j and for every $k \in [u_j]$ we add the variables $x_j^{(1,k)}, x_j^{(2,k)}, x_j^{(3,k)}, x_j^{(4,k)}$ with zero objective coefficients. The variable $x_j^{(1,k)}$ is assigned the original column $W_{\cdot,j}$ in W , whereas all other variables are assigned zero columns. We set all variable domains to $[0, 1]$. Finally, the variables are given suitable coefficients in M by adding the constraints $x_j^{(1,k)} + x_j^{(2,k)} = x_j^{(2,k)} + x_j^{(3,k)} = x_j^{(3,k)} + x_j^{(4,k)} = x_j^{(4,k)} + x_j^{(1,k)} = 1$. In this way, the old variable x_j corresponds to $\sum_{k \in [u_j]} x_j^{(1,k)}$ in the new constrained perfect matching problem.

We now employ Lemma 30 to reduce the coefficients of W . Observe that the reduction steps subdivides an edge into three edges. Therefore, the obtained M is the incidence matrix of a graph that is the disjoint union of even length cycles. ◀

This shows that, unlike n -fold ILP, the constrained matching problem is unlikely to become FPT when the coefficient size is bounded.

6 Concluding notes and open problems

We study the complexity of ILP parameterized by the size of variable or constraint backdoors to the generalized matching problem, called p and h respectively. In particular, we show that solving ILPs is in FPT when parameterized by p . We study the convexity of degree sequences in the light of SBO jump M -convex functions to express the non-backdoor variables efficiently as polyhedral constraints. Additionally, we present a randomized XP time algorithm to solve ILPs for fixed h when the objective and constraint matrix are encoded in unary. This algorithm employs a Graver basis augmentation procedure to reduce the problem to the exact matching problem. Finally, we match this latter result by showing that ILP is $W[1]$ -hard parameterized by h .

Having studied the case where $p > 0, h = 0$ and where $p = 0, h > 0$, a natural open question is whether ILP can still be solved in polynomial time when it is a generalized matching problem with a fixed number of both additional variables and constraints. This can be seen as an analogue of 4-block n -fold ILP in the context of block structured ILPs, which are known to be solvable in slice-wise polynomial time [28]. Since the restricted case of $p = 0, h > 0$ in the context of generalized matching is $W[1]$ -hard in terms of h , the question is whether $p > 0, h > 0$ also admits a randomized XP time algorithm when c, A are encoded in unary. We note that providing a polynomial Graver-complexity upper bound for ILP (T) will suffice to obtain this result. In such case, the Graver-complexity of the entire constraint matrix will also be polynomially bounded and a Graver-best step can be computed by brute forcing the values of the p arbitrary variables, after which a problem of ILP (W) can be solved efficiently.

In addition, aside from the long standing open question of whether the exact matching problem is in P, a natural open question that arises from Theorem 3 is whether the exponential dependence on the encoding length of c is necessary. To answer this question positively, it suffices to restrict to 0/1-budgeted perfect matching with its objective in encoded binary.

Finally, we suspect that some intermediate steps used in obtaining Theorem 2 can be refined. That is, we suspect that Lemma 23 holds for general jump M-convex functions. In addition, it may be possible to make a combinatorial separation algorithm for $P_{r,U}$ in a more direct fashion as done in [49].

References

- 1 Richard P. Anstee and Yunsun Nam. Convexity of degree sequences. *Journal of Graph Theory*, 30(2):147–156, 1999.
- 2 Egon Balas and William R. Pulleyblank. The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, 13(4):495–516, 1983.
- 3 Max Bannach, Sebastian Berndt, Marten Maack, Matthias Mnich, Alexandra Lassota, Malin Rau, and Malte Skambath. Solving packing problems with few small items using rainbow matchings. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science, MFCS*, pages 11:1–11:14, 2020.
- 4 John Bartholdi, Craig A Tovey, and Michael A Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.
- 5 André Berger, Vincenzo Bonifaci, Fabrizio Grandoni, and Guido Schäfer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Mathematical Programming*, 128(1-2):355–372, 2011.
- 6 Sebastian Berndt, Matthias Mnich, and Tobias Stamm. New support size bounds and proximity bounds for integer linear programming. In *Proceedings of the 49th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM*, pages 82–95, 2024.
- 7 André Bouchet and William H. Cunningham. Delta-matroids, jump systems, and bisubmodular polyhedra. *SIAM Journal on Discrete Mathematics*, 8(1):17–32, 1995.
- 8 Marcin Brianski, Martin Koutecký, Daniel Král’, Kristýna Pekárková, and Felix Schröder. Characterization of matrices with bounded graver bases and depth parameters and applications to integer programming. *Mathematical Programming*, 208(1-2):497–531, 2024.
- 9 Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13(2):258–273, 1992.
- 10 William J. Cook, Albertus M. H. Gerards, Alexander Schrijver, and Éva Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34(3):251–264, 1986.
- 11 Jana Cslovjceksek, Martin Koutecký, Alexandra Lassota, Michal Pilipczuk, and Adam Polak. Parameterized algorithms for block-structured integer programs with large entries. In *Pro-*

- ceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 740–751, 2024.
- 12 Daniel Dadush. *Integer programming, lattice algorithms, and deterministic volume estimation*. Georgia Institute of Technology, 2012.
 - 13 Pavel Dvořák, Eduard Eiben, Robert Ganian, Dusan Knop, and Sebastian Ordyniak. The complexity landscape of decompositional parameters for ILP: programs with few global variables and constraints. *Artificial Intelligence*, 300:103561, 2021.
 - 14 Jack Edmonds and Ellis L. Johnson. Matching: A well-solved class of integer linear programs. In *Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications*, pages 89–92, 1970.
 - 15 Jack Edmonds and Ellis L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5(1):88–124, 1973.
 - 16 Friedrich Eisenbrand, Christoph Hunkenschröder, and Kim-Manuel Klein. Faster algorithms for integer programs with block structure. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 49:1–49:13, 2018.
 - 17 Friedrich Eisenbrand, Christoph Hunkenschröder, Kim-Manuel Klein, Martin Koutecký, Asaf Levin, and Shmuel Onn. An algorithmic theory of integer programming, 2019. [arXiv:1904.01361](#).
 - 18 Friedrich Eisenbrand and Thomas Rothvoss. A parameterized linear formulation of the integer hull, 2025. [arXiv:2501.02347](#).
 - 19 T. Ermolieva, Y Ermoliev, P. Havlik, A. Lessa-Derci-Augustynczik, N. Komendantova, T. Kahil, J. Balkovic, R. Skalsky, C. Folberth, P. S. Knopov, and G. Wang. Connections between robust statistical estimation, robust decision-making with two-stage stochastic optimization, and robust machine learning problems. *Cybernetics and Systems Analysis*, 59(3):385–397, 2023.
 - 20 Michael R. Fellows, Daniel Lokshantov, Neeldhara Misra, Frances A. Rosamond, and Saket Saurabh. Graph layout problems parameterized by vertex cover. In *Proceedings of the 19th International Symposium on Algorithms and Computation, ISAAC*, pages 294–305, 2008.
 - 21 Jirí Fiala, Tomas Gavenciak, Dusan Knop, Martin Koutecký, and Jan Kratochvíl. Parameterized complexity of distance labeling and uniform channel assignment problems. *Discrete Applied Mathematics*, 248:46–55, 2018.
 - 22 Samuel Fiorini, Gwenaél Joret, Stefan Weltge, and Yelena Yuditsky. Integer programs with bounded subdeterminants and two nonzeros per row. In *Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 13–24, 2021.
 - 23 Serge Gaspers and Stefan Szeider. Backdoors to satisfaction. In *The Multivariate Algorithmic Revolution and Beyond*, pages 287–317. Springer, 2012.
 - 24 Tomáš Gavenciak, Martin Koutecký, and Dušan Knop. Integer programming in parameterized complexity: Five miniatures. *Discrete Optimization*, 44(Part 1):100596, 2022.
 - 25 Michel X. Goemans and Thomas Rothvoss. Polynomiality for bin packing with a constant number of item types. *Journal of the ACM*, 67(6):38:1–38:21, 2020.
 - 26 Jens Gramm, Rolf Niedermeier, and Peter Rossmanith. Fixed-parameter algorithms for CLOSEST STRING and related problems. *Algorithmica*, 37(1):25–42, 2003.
 - 27 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.
 - 28 Raymond Hemmecke, Matthias Köppe, and Robert Weismantel. A polynomial-time algorithm for optimizing over N -fold 4-block decomposable integer programs. In *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization, IPCO*, pages 219–229, 2010.
 - 29 Raymond Hemmecke, Matthias Köppe, and Robert Weismantel. Graver basis and proximity techniques for block-structured separable convex integer minimization problems. *Mathematical Programming*, 145(1-2):1–18, 2014.
 - 30 Christoph Hunkenschröder, Kim-Manuel Klein, Martin Koutecký, Alexandra Lassota, and Asaf Levin. Tight lower bounds for block-structured integer programs. In *Proceedings of*

- the 25th International Conference on Integer Programming and Combinatorial Optimization, *IPCO*, pages 224–237, 2024.
- 31 Klaus Jansen, Kim-Manuel Klein, Marten Maack, and Malin Rau. Empowering the configuration-ip: new PTAS results for scheduling with setup times. *Math. Program.*, 195(1):367–401, 2022.
 - 32 Klaus Jansen, Alexandra Lassota, and Lars Rohwedder. Near-linear time algorithm for n-fold ILPs via color coding. *SIAM Journal on Discrete Mathematics*, 34(4):2282–2299, 2020.
 - 33 Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
 - 34 Dusan Knop, Martin Koutecký, and Matthias Mnich. Voting and bribing in single-exponential time. *ACM Transactions on Economics and Computation*, 8(3):12:1–12:28, 2020.
 - 35 Yusuke Kobayashi. Optimal general factor problem and jump system intersection. In *Proceedings of the 24th International Conference on Integer Programming and Combinatorial Optimization, IPCO*, pages 291–305, 2023.
 - 36 Alexandra Lassota, Aleksander Lukaszewicz, and Adam Polak. Tight vector bin packing with few small items via fast exact matching in multigraphs. In *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 87:1–87:15, 2022.
 - 37 László Lovász. The factorization of graphs. II. *Acta Mathematica Academiae Scientiarum Hungarica*, 23(1-2):223–246, 1972.
 - 38 Meena Mahajan, P. R. Subramanya, and V. Vinay. A combinatorial algorithm for pfaffians. In *Proceedings of the 5th International Computing and Combinatorics Conference, COCOON*, pages 134–143, 1999.
 - 39 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.
 - 40 Kazuo Murota. M-convex functions on jump systems: A general framework for minsquare graph factor problem. *SIAM Journal on Discrete Mathematics*, 20(1):213–226, 2006.
 - 41 Kazuo Murota and Ken’ichiro Tanaka. A steepest descent algorithm for M-convex functions on jump systems. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E89-A(5):1160–1165, 2006.
 - 42 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
 - 43 Manfred W. Padberg and M. R. Rao. Odd minimum cut-sets and b -matchings. *Mathematics of Operations Research*, 7(1):67–80, 1982.
 - 44 Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29(2):285–309, 1982.
 - 45 Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming. In *Proceedings of the 64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 974–988, 2023.
 - 46 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.
 - 47 Éva Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986.
 - 48 Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, third edition, 2013.
 - 49 Fan Zhang. A separation algorithm for b -matching degree-sequence polyhedra. *Mathematics of Operations Research*, 28(1):92–102, 2003.

A A direct MILP approach to exploit variable backdoors

Recent independent and concurrent research by Eisenbrand and Rothvoss [18] shows that the integer hull of arbitrary polyhedra $Ax \leq b$ can be expressed as a system that depends linearly on b as long as b has a fixed remainder modulo a large integer depending only on the dimensions of A and the coefficient size. They use this to optimize two-stage stochastic

integer programming with large coefficients in the constraint matrix corresponding to the variable backdoor in FPT time. To accomplish this, they replace the non-backdoor variables with real variables and replace the constraints on those variables with their parametric versions of the integer hull. Their strategy reveals an additional way in which Theorem 2 can be derived, which we shortly discuss.

Consider the perfect b -matching problem with additional variables resulting from Lemma 5. Let $P(z)$ be the polyhedron $\{x \in \mathbb{R}^n : Mx = z, x \geq 0\}$ and $P_I(z)$ its integer hull $\text{conv}(P(z) \cap \mathbb{Z}^n)$. It is known that $P_I(z)$ is given by

$$P_I(z) = \{x \in \mathbb{R}^n : Mx = z, x \geq 0, Q_b x \leq q\},$$

where $Q_b x \leq q$ consists of the constraints $\sum_{j \in \delta(U)} x_j \geq 1$ for all $U \subseteq [m]$ such that $\sum_{i \in U} z_i$ is odd. See Corollary 31.2a in [46]. Here $\delta(U)$ are the edges connecting U and its complement. These constraints are identical for all $z \equiv r \pmod{2}$ for a fixed remainder r , which justifies labeling our system as $Q_r x \leq q$.

After guessing the remainder $r' \equiv y \pmod{2}$, we may solve the b -matching variant of ILP (T) by solving the problem

$$\min\{c^\top(y, x) \mid y \in 2\mathbb{Z}^p + r', \mathbf{0} \leq y \leq g, x \in P_I(b - Ty)\},$$

as after finding an optimal y , we can obtain vertex solutions x that are integral. This is equivalent to

$$\min\{c^\top(2v + r', x) \mid v \in \mathbb{Z}, \mathbf{0} \leq 2v + r' \leq g, x \in P_I(b - T(2v + r'))\}.$$

As $z = b - T(2v + r') \equiv r \pmod{2}$ for all v for suitably chosen r , we may restrict to solving

$$\min\{c^\top(2v + r', x) \mid v \in \mathbb{Z}, \mathbf{0} \leq 2v + r' \leq g, Mx = b - T(2v + r'), x \geq 0, Q_r x \leq q\}.$$

This mixed integer linear program can be solved in FPT time parameterized by p as a result of Lenstra's algorithm [33] as long as we can optimize any linear function over its linear relaxation in polynomial time. The latter is possible by using the ellipsoid method [27] and Padberg's and Rao odd minimum cut algorithm [43], which allows to efficiently separate the inequalities in $Q_r x \leq q$.