Parameterized algorithms for block-structured integer programs with large entries*

Jana Cslovjecsek[†] Martin Koutecký[‡] Alexandra Lassota[§] Michał Pilipczuk[¶]
Adam Polak[‖]

Abstract

We study two classic variants of block-structured integer programming. *Two-stage stochastic programs* are integer programs of the form $\{A_i\mathbf{x} + D_i\mathbf{y}_i = \mathbf{b}_i \text{ for all } i = 1, \ldots, n\}$, where A_i and D_i are bounded-size matrices. Intuitively, this form corresponds to the setting when after setting a small set of global variables $\mathbf{x}$, the program can be decomposed into a possibly large number of bounded-size subprograms. On the other hand, *n-fold programs* are integer programs of the form $\{\sum_{i=1}^{n} C_i\mathbf{y}_i = \mathbf{a} \text{ and } D_i\mathbf{y}_i = \mathbf{b}_i \text{ for all } i = 1, \ldots, n\}$, where again C_i and D_i are bounded-size matrices. This form is natural for knapsack-like problems, where we have a large number of variables partitioned into small-size groups, each group needs to obey some set of local constraints, and there are only a few global constraints that link together all the variables.

A line of recent work established that the optimization problem for both two-stage stochastic programs and n-fold programs is fixed-parameter tractable when parameterized by the dimensions of relevant matrices A_i, C_i, D_i and by the maximum absolute value of any entry appearing in the constraint matrix. A fundamental tool used in these advances is the notion of the *Graver basis* of a matrix, and this tool heavily relies on the assumption that all the entries of the constraint matrix are bounded.

In this work, we prove that the parameterized tractability results for two-stage stochastic and n-fold programs persist even when one allows large entries in the global part of the program. More precisely, we prove the following: In this work, we prove that the parameterized tractability results for two-stage stochastic and n-fold programs persist even when one allows large entries in the global part of the program. More precisely, we prove the following:

- The feasibility problem for two-stage stochastic programs is fixed-parameter tractable when parameterized by the dimensions of matrices A_i, D_i and by the maximum absolute value of the entries of matrices D_i. That is, we allow matrices A_i to have arbitrarily large entries.
- The linear optimization problem for n-fold integer programs that are uniform – all matrices C_i are equal – is fixed-parameter tractable when parameterized by the dimensions of matrices C_i and D_i and by the maximum absolute value of the entries of matrices D_i. That is, we require that $C_i = C$ for all $i = 1, \ldots, n$, but we allow C to have arbitrarily large entries.

In the second result, the uniformity assumption is necessary; otherwise the problem is NP-hard already when the parameters take constant values. Both our algorithms are weakly polynomial: the running time is measured in the total bitsize of the input.

In both results, we depart from the approach that relies purely on Graver bases. Instead, for two-stage stochastic programs, we devise a reduction to integer programming with a bounded number of variables using new insights about the combinatorics of integer cones. For n-fold programs, we reduce a given n-fold program to an exponential-size program with bounded right-hand sides, which we consequently solve using a reduction to mixed integer programming with a bounded number of integral variables.

*The full version of the paper can be accessed at https://arxiv.org/abs/2311.01890. Martin Koutecký was partially supported by Charles University project UNCE/SCI/004 and by the project 22-22997S of GA ČR. The work of Michał Pilipczuk on this manuscript is a part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, Grant Agreement No 948057 — BOBR. Part of this work was done when Adam Polak and Alexandra Lassota were at École Polytechnique Fédérale de Lausanne, supported by the Swiss National Science Foundation projects *Lattice Algorithms and Integer Programming* (185030) and *Complexity of Integer Programming* (CRFS-2_207365). Part of this work was carried out while Alexandra Lassota was affiliated with Max Planck Institute for Informatics, SIC, Saarbrücken, Germany.

[†]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[‡]Computer Science Institute, Charles University, Prague, Czech Republic
[§]Eindhoven University of Technology, Eindhoven, The Netherlands
[¶]Institute of Informatics, University of Warsaw, Poland
[‖]Department of Computing Sciences, Bocconi University, Milan, Italy

1 Introduction

We study two variants of integer programming problems, where the specific structure of the constraint matrix can be exploited for the design of efficient parameterized algorithms. *Two-stage stochastic programs* are integer programs of the following form:

$$(\spadesuit) \qquad \begin{aligned} &\mathbf{x} \in \mathbb{Z}^k_{\geqslant 0},\ \mathbf{y}_i \in \mathbb{Z}^k_{\geqslant 0}, && \text{and} \\ &A_i\mathbf{x} + D_i\mathbf{y}_i = \mathbf{b}_i && \text{for all } i = 1, 2, \ldots, n. \end{aligned}$$

Here, A_i, D_i are integer $k \times k$ matrices[1] and each $\mathbf{b}_i$ is an integer vector of length k. This form arises naturally when the given integer program can be decomposed into multiple independent subprograms on disjoint variable sets $\mathbf{y}_i$, except there are several global variables $\mathbf{x}$ that are involved in all the subprograms and thus link them. See the survey of Shultz et al. [41] as well as an exposition article by Gavenčiak et al. [19] for examples of applications.

We moreover study *n-fold programs* which are integer programs of the form

$$(\clubsuit) \qquad \begin{aligned} &\mathbf{y}_i \in \mathbb{Z}^k_{\geqslant 0}, && \\ &\sum_{i=1}^{n} C_i\mathbf{y}_i = \mathbf{a}, && \text{and} \\ &D_i\mathbf{y}_i = \mathbf{b}_i && \text{for all } i = 1, 2, \ldots, n, \end{aligned}$$

where again C_i, D_i are integer $k \times k$ matrices and $\mathbf{a}, \mathbf{b}_i$ are integer vectors of length k. These kind of programs appear for knapsack-like and scheduling problems, where blocks of variables $\mathbf{y}_i$ correspond to some independent local decisions (for instance, whether to pack an item into the knapsack or not), and there are only a few linear constraints that involve all variables (for instance, that the capacity of the knapsack is not exceeded). See [9, 14, 19, 23, 28, 30, 32, 33] for examples of n-fold programs appearing naturally "in the wild". Figure 1 depicts constraint matrices of two-stage stochastic and n-fold programs.

$$\begin{bmatrix} A_1 & D_1 & & & \\ A_2 & & D_2 & & \\ \vdots & & & \ddots & \\ A_n & & & & D_n \end{bmatrix} \qquad \begin{bmatrix} C_1 & C_2 & \ldots & C_n \\ D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_n \end{bmatrix} \qquad \begin{bmatrix} B & C_1 & C_2 & \ldots & C_n \\ A_1 & D_1 & & & \\ A_2 & & D_2 & & \\ \vdots & & & \ddots & \\ A_n & & & & D_n \end{bmatrix}$$

Figure 1: Constraint matrices in two-stage stochastic, n-fold, and 4-block integer programs, respectively. (The last kind will be discussed later.) Every block is a $k \times k$ matrix, where k is the parameter. Empty spaces denote blocks of zeroes.

Both for two-stage stochastic programs and for n-fold programs, we can consider two computational problems. The simpler *feasibility problem* just asks whether the given program has a *solution*: an evaluation of variables in nonnegative integers that satisfies all the constraints. In the harder *(linear) optimization problem*, we are additionally given an integer weight w_x for every variable x appearing in the program, and the task is to minimize $\sum_{x:\text{ variable}} w_x \cdot x$ over all solutions.

Two-stage stochastic and n-fold programs have recently gathered significant interest in the theoretical community for two reasons. On one hand, it turns out that for multiple combinatorial problems, their natural integer programming formulations take either of the two forms. On the other hand, one can actually design highly non-trivial fixed-parameter algorithms for the optimization problem for both two-stage stochastic and n-fold programs; we will review them in a minute. Combining this two points yields a powerful algorithmic technique that allowed multiple new tractability results and running times improvements for various problems of combinatorial optimization; see [9, 19, 23, 28, 29, 30, 32, 33, 37] for examples.

Delving more into technical details, if by Δ we denote the maximum absolute value of any entry in the constraint matrix, then the optimization problem for

[1] Reliance on square matrices is just for convenience of presentation. The setting where blocks are rectangular matrices with dimensions bounded by k can be reduced to the setting of $k \times k$ square matrices by just padding with 0s.

- two-stage stochastic programs (♠) can be solved in time $2^{\Delta^{\mathcal{O}(k^2)}} \cdot n \log^{\mathcal{O}(k^2)} n$ [11]; and
- n-fold programs (♣) can be solved in time $(k\Delta)^{\mathcal{O}(k^3)} \cdot n \log^{\mathcal{O}(k^2)} n$ [10].

The results above are in fact pinnacles of an over-a-decade-long sequence of developments, which gradually improved both the generality of the results and the running times [2, 10, 11, 14, 15, 16, 21, 24, 26, 35], as well as provided complexity lower bounds [22, 34]. We refer the interested reader to the monumental manuscript of Eisenbrand et al. [16] which provides a comprehensive perspective on this research area.

We remark that the tractability results presented above can be also extended to the setting where the global-local block structure present in two-stage stochastic and n-fold programs can be iterated further, roughly speaking to trees of bounded depth. This leads to the study of integer programs with bounded *primal* or *dual treedepth*, for which analogous tractability results have been established. Since these notions will not be of interest in this work, we refrain from providing further details and refer the interested reader to the works relevant for this direction [3, 4, 8, 10, 11, 15, 16, 26, 27, 34, 35].

All the abovementioned works, be it on two-stage stochastic or n-fold programs, or on programs of bounded primal or dual treedepth, crucially rely on one tool: the notion of the *Graver basis* of a matrix. Intuitively speaking, the Graver basis of a matrix A consists of "minimal steps" within the lattice of integer points belonging to the kernel of A. Therefore, the maximum norm of an element of the Graver basis reflects the "granularity" of this lattice. And so, the two fundamental observations underlying all the discussed developments are the following:

- in two-stage stochastic matrices (or more generally, matrices of bounded primal treedepth), the ℓ_∞ norm of the elements of the Graver basis is bounded in terms of k (the dimension of every block) and the maximum absolute value of any entry appearing in the matrix (see [16, Lemma 28]); and
- an analogous result holds for n-fold matrices (or more generally, matrices of bounded dual treedepth) and the ℓ_1 norm (see [16, Lemma 30]).

Based on these observations, various algorithmic strategies, including augmentation frameworks [24, 35] and proximity arguments [10, 11, 15], can be employed to solve respective integer programs.

The drawback of the Graver-based approach is that it heavily relies on the assumption that all the entries of the input matrices are (parametrically) bounded. Indeed, the norms of the elements of the Graver basis are typically at least as large as the entries, so lacking any upper bound on the latter renders the methodology inapplicable. This is in stark contrast with the results on fixed-parameter tractability of integer programming parameterized by the number of variables [12, 13, 18, 25, 38, 40], which rely on different tools and for which no bound on the absolute values of the entries is required. In fact, two-stage stochastic programs generalize programs with a bounded number of variables (just do not use variables $\mathbf{y}_i$), yet the current results for two-stage stochastic programs do *not* generalize those for integer programs with few variables, because they assume a bound on the absolute values of the entries.

The goal of this paper is to understand to what extent one can efficiently solve two-stage stochastic and n-fold programs while allowing large entries on input.

Our contribution. We prove that both the feasibility problem for two-stage stochastic programs and the optimization problem for uniform n-fold programs (that is, where $C_1 = C_2 = \ldots = C_n$) can be solved in fixed-parameter time when parameterized by the dimensions of the blocks and the maximum absolute value of any entry appearing in the diagonal blocks D_i. That is, we allow the entries of the global blocks (A_i and C_i, respectively) to be arbitrarily large, and in the case of n-fold programs, we require that all blocks C_i are equal. The statements below summarize our results. ($\|P\|$ denotes the total bitsize of a program P.)

THEOREM 1.1. *The feasibility problem for two-stage stochastic programs P of the form (♠) can be solved in time $f(k, \max_i \|D_i\|_\infty) \cdot \|P\|$ for a computable function f, where k is the dimension of all the blocks A_i, D_i.*

THEOREM 1.2. *The optimization problem for n-fold programs P of the form (♣) that are uniform (that is, satisfy $C_1 = \ldots = C_n$) can be solved in time $f(k, \max_i \|D_i\|_\infty) \cdot \|P\|^{\mathcal{O}(1)}$ for a computable function f, where k is the dimension of all the blocks C_i, D_i.*

The uniformity condition in Theorem 1.2 is necessary (unless $\mathsf{P} = \mathsf{NP}$), as one can very easily reduce SUBSET SUM to the feasibility problem for n-fold programs with $k = 2$ and D_i being $\{0,1\}$-matrices. Indeed, given an instance of SUBSET SUM consisting of positive integers $a_1, \ldots, a_n$ and a target integer t, we can write the following n-fold program on variables $y_1, \ldots, y_n, y'_1, \ldots, y'_n \in \mathbb{Z}_{\geqslant 0}$: $y_i + y'_i = 1$ for all $i = 1, \ldots, n$ and $\sum_{i=1}^n a_i y_i = t$. We remark that uniform n-fold programs are actually of the highest importance, as this form typically arises in applications. In fact, many of the previous works named such problems "n-fold", while our formulation (♣) would be called "generalized n-fold".

We also remark that the algorithm of Theorem 1.2 does not use the assumption that the number of *rows* of matrix C is bounded by k (formally, in the formal statement proved in the full version of this work, we do not consider this number among parameters). However, we stress that the bound on the number of *columns* of C is heavily exploited, which sets our approach apart from many of the previous works [10, 15, 35].

Further, observe that Theorem 1.1 applies only to the feasibility problem for two-stage stochastic programs. We actually do not know whether Theorem 1.1 can be extended to the optimization problem as well, and we consider determining this an outstanding open problem. We will discuss its motivation in more details later on. Also, we remark that Theorem 1.1 seems to be the first algorithm for feasibility of two-stage stochastic programs that achieves truly linear dependence of the running time on the total input size; the earlier algorithms of [11, 35] had at least some additional polylogarithmic factors.

Finally, note that the algorithms of Theorems 1.1 and 1.2 are not strongly polynomial (i.e., the running time depends on the total bitsize of the input, rather than is counted in the number of arithmetic operations), while the previous algorithms of [10, 11, 15, 35] for the stronger parameterization are. At least in the case of Theorem 1.1, this is justified, as the problem considered there generalizes integer programming parameterized by the number of variables, for which strongly polynomial FPT algorithms are not known.

Not surprisingly, the proofs of Theorems 1.1 and 1.2 depart from the by now standard approach through Graver bases; they are based on entirely new techniques, with some key Graver-based insight needed in the case of Theorem 1.2. In both cases, the problem is ultimately reduced to (mixed) integer programming with a bounded number of (integral) variables, and this allows us to cope with large entries on input. We expand the discussion of our techniques in Section 2, which contains a technical overview of the proofs.

4-block programs. Finally, we would like to discuss another motivation for investigating two-stage stochastic and n-fold programs with large entries, namely the open question about the parameterized complexity of 4-*block integer programming*. 4-block programs are programs in which the constraint matrix has the block-structured form depicted in the right panel of Figure 1; note that this form naturally generalizes both two-stage stochastic and n-fold programs. It is an important problem in the area to determine whether the feasibility problem for 4-block programs can be solved in fixed-parameter time when parameterized by the dimension of blocks k and the maximum absolute value of any entry in the input matrix. The question was asked by Hemmecke et al. [20], who proposed an XP algorithm for the problem. Improvements on the XP running time were reported by Chen et al. [7], and FPT algorithms for special cases were proposed by Chen et al. [5]; yet no FPT algorithm for the problem in full generality is known so far. We remark that recently, Chen et al. [6] studied the complexity of 4-block programming while allowing large entries in all the four blocks of the matrix. They showed that then the problem becomes NP-hard already for blocks of constant dimension, and they discussed a few special cases that lead to tractability.

We observe that in the context of the feasibility problem for uniform 4-block programs (i.e., with $A_i = A$ and $C_i = C$ for all $i = 1, \ldots, n$), it is possible to emulate large entries within the global blocks A, B, C using only small entries at the cost of adding a bounded number of auxiliary variables. This yields the following reduction, whose proof can be found in the full version of this work.

OBSERVATION 1. *Suppose the feasibility problem for uniform* 4-*block programs can be solved in time* $f(k, \Delta) \cdot \|P\|^{\mathcal{O}(1)}$ *for some computable function* f*, where* k *is the dimension of every block and* Δ *is the maximum absolute value of any entry in the constraint matrix. Then the feasibility problem for uniform* 4-*block programs can be also solved in time* $g(k, \max_i \|D_i\|_\infty) \cdot \|P\|^{\mathcal{O}(1)}$ *for some computable function* g *under the assumption that all the absolute values of the entries in matrices* A, B, C *are bounded by* n.

Consequently, to approach the problem of fixed-parameter tractability of 4-block integer programming, it is imperative to understand first the complexity of two-stage stochastic and n-fold programming with large entries allowed in the global blocks. And this is precisely what we do in this work.

We believe that the next natural step towards understanding the complexity of 4-block integer programming would be to extend Theorem 1.1 to the optimization problem; that is, to determine whether optimization of two-stage stochastic programs can be solved in fixed-parameter time when parameterized by k and $\max_i \|D_i\|_\infty$. Indeed, lifting the result from feasibility to the optimization problem roughly corresponds to adding a single constraint that links all the variables, and 4-block programs differ from two-stage stochastic programs precisely in that there may be up to k such additional linking constraints. Thus, we hope that the new approach to block-structured integer programming presented in this work may pave the way towards understanding the complexity of solving 4-block integer programs.

Acknowledgements. This research has been initiated during the trimester on Discrete Optimization at the Hausdorff Research Institute for Mathematics (HIM) in Bonn, Germany. We thank the organizers of the trimester for creating a friendly and motivating research environment. We also thank Eleonore Bach, Fritz Eisenbrand, and Robert Weismantel, for pointing us to the work of Aliev and Henk [1].

2 Overview

In this section we provide a technical overview of our results aimed at presenting the main ideas and new conceptual contributions. Complete and formal proofs of all our results can be found in the full version of this paper.

2.1 Two-stage stochastic programming. We start with an overview on the proof of Theorem 1.1. We will heavily rely on the combinatorics of integer and polyhedral cones, so let us recall basic definitions and properties.

Cones. Consider an integer matrix D with t columns and k rows. The *polyhedral cone* spanned by D is the set $\mathsf{cone}(D) \coloneqq \{D\mathbf{y} \colon \mathbf{y} \in \mathbb{R}^t_{\geqslant 0}\} \subseteq \mathbb{R}^k_{\geqslant 0}$, or equivalently, the set of all vectors in $\mathbb{R}^k_{\geqslant 0}$ expressible as nonnegative combinations of the columns of D. Within the polyhedral cone, we have the *integer cone* where we restrict attention to nonnegative integer combinations: $\mathsf{intCone}(D) \coloneqq \{D\mathbf{y} \colon \mathbf{y} \in \mathbb{Z}^t_{\geqslant 0}\} \subseteq \mathbb{Z}^k$. Finally, the *integer lattice* is the set $\mathsf{lattice}(D) \coloneqq \{D\mathbf{y} \colon \mathbf{y} \in \mathbb{Z}^t\} \subseteq \mathbb{Z}^k$ which comprises all integer combinations of columns of D with possibly negative coefficients.

Clearly, not every integer vector in $\mathsf{cone}(D)$ has to belong to $\mathsf{intCone}(D)$. It is not even necessarily the case that $\mathsf{intCone}(D) = \mathsf{cone}(D) \cap \mathsf{lattice}(D)$, as there might be vectors that can be obtained both as a nonnegative combination and as an integer combination of columns of D, but every such integer combination must necessarily contain negative coefficients. To see an example, note that in dimension $k = 1$, this is the Frobenius problem: supposing all entries of D are positive integers, the elements of $\mathsf{intCone}(D)$ are essentially all nonnegative numbers divisible by the gcd (greatest common divisor) of the entries of D, except that for small numbers there might be some aberrations: a positive integer of order $\mathcal{O}(\|D\|_\infty^2)$ may not be presentable as a nonnegative combination of the entries of D, even assuming it is divisible by the gcd of the entries of D.

However, the Frobenius example suggests that the equality $\mathsf{intCone}(D) = \mathsf{cone}(D) \cap \mathsf{lattice}(D)$ is almost true, except for aberrations near the boundary of $\mathsf{cone}(D)$. We forge this intuition into a formal statement presented below that says roughly the following: if one takes a look at $\mathsf{intCone}(D)$ at a large scale, by restricting attention to integer vectors $\mathbf{v} \in \mathbb{Z}^k$ with fixed remainders of entries modulo some large integer B, then $\mathsf{intCone}(D)$ behaves like a polyhedron. In the following, for a positive integer B and a vector $\mathbf{r} \in \{0, 1, \ldots, B-1\}^k$, we let $\Lambda^B_{\mathbf{r}}$ be the set of all vectors $\mathbf{v} \in \mathbb{Z}^k$ such that $\mathbf{v} \equiv \mathbf{r} \bmod B$, which means $v_i \equiv r_i \bmod B$ for all $i \in \{1, \ldots, k\}$.

Theorem 2.1. (Reduction to Polyhedral Constraints) *Let D be an integer matrix with t columns and k rows. Then there exists a positive integer B, computable from D, such that for every $\mathbf{r} \in \{0, 1, \ldots, B-1\}^k$, there exists a polyhedron $\mathcal{Q}_{\mathbf{r}}$ such that*

$$\Lambda^B_{\mathbf{r}} \cap \mathsf{intCone}(D) = \Lambda^B_{\mathbf{r}} \cap \mathcal{Q}_{\mathbf{r}}.$$

Moreover, a representation of such a polyhedron $\mathcal{Q}_{\mathbf{r}}$ can be computed given D and $\mathbf{r}$.

In other words, Theorem 2.1 states that if one fixes the remainders of entries modulo B, then membership in the integer cone can be equivalently expressed through a finite system of linear inequalities. Before we sketch the proof of Theorem 2.1, let us discuss how to use this to solve two-stage stochastic programs.

The algorithm. Consider a two-stage stochastic program $P = (A_i, D_i, \mathbf{b}_i \colon i \in \{1, \ldots, n\})$ such that blocks A_i, D_i are integer $k \times k$ matrices and all entries of blocks D_i are bounded in absolute value by Δ. The feasibility problem for P can be understood as the question about satisfaction of the following sentence, where all quantifications range over $\mathbb{Z}^k_{\geqslant 0}$:

$$\exists_{\mathbf{x}} \left(\bigwedge_{i=1}^{n} \exists_{\mathbf{y}_i}\ A_i\mathbf{x} + D_i\mathbf{y}_i = \mathbf{b}_i \right), \quad \text{or equivalently,} \quad \exists_{\mathbf{x}} \left(\bigwedge_{i=1}^{n} \mathbf{b}_i - A_i\mathbf{x} \in \mathsf{intCone}(D_i) \right). \tag{2.1}$$

Applying Theorem 2.1 to each matrix D_i yields a positive integer B_i. Note that there are only at most $(2\Delta+1)^{k^2}$ different matrices D_i appearing in P, which also bounds the number of different integers B_i. By replacing all B_is with their least common multiple, we may assume that $B_1 = B_2 = \ldots = B_n = B$. Note that B is bounded by a computable function of Δ and k.

Consider a hypothetical solution $\mathbf{x}, (\mathbf{y}_i : i \in \{1, \ldots, n\})$ to P. We guess, by branching into B^k possibilities, a vector $\mathbf{r} \in \{0, 1, \ldots, B-1\}^k$ such that $\mathbf{x} \equiv \mathbf{r} \bmod B$. Having fixed $\mathbf{r}$, we know how the vectors $\mathbf{b}_i - A_i\mathbf{x}$ look like modulo B, hence by Theorem 2.1, we may replace the assertion $\mathbf{b}_i - A_i\mathbf{x} \in \mathsf{intCone}(D_i)$ with the assertion $\mathbf{b}_i - A_i\mathbf{x} \in \mathcal{Q}_{\mathbf{r}_i}$, where $\mathbf{r}_i \in \{0, 1, \ldots, B-1\}^k$ is the unique vector such that $\mathbf{b}_i - A_i\mathbf{r} \equiv \mathbf{r}_i \bmod B$. Thus, (2.1) can be rewritten to the sentence

$$\bigvee_{\mathbf{r} \in \{0,1,\ldots,B-1\}^k} \exists_{\mathbf{x}} \ (\mathbf{x} \equiv \mathbf{r} \bmod B) \wedge \left(\bigwedge_{i=1}^{n} \mathbf{b}_i - A_i\mathbf{x} \in \mathcal{Q}_{\mathbf{r}_i} \right),$$

which is equivalent to

$$\bigvee_{\mathbf{r} \in \{0,1,\ldots,B-1\}^k} \exists_{\mathbf{x}} \exists_{\mathbf{z}} \ (\mathbf{x} = B \cdot \mathbf{z} + \mathbf{r}) \wedge \left(\bigwedge_{i=1}^{n} \mathbf{b}_i - A_i\mathbf{x} \in \mathcal{Q}_{\mathbf{r}_i} \right). \tag{2.2}$$

Verifying satisfiability of (2.2) boils down to solving B^k integer programs on $2k$ variables $\mathbf{x}$ and $\mathbf{z}$ and linearly FPT many constraints, which can be done in linear fixed-parameter time using standard algorithms, for instance that of Kannan [25].

We remark that the explanation presented above highlights that Theorem 2.1 can be understood as a quantifier elimination result in the arithmetic theory of integers. This may be of independent interest, but we do not pursue this direction in this work.

Reduction to polyhedral constraints. We are left with sketching the proof of Theorem 2.1. Let $\mathcal{Z} := \Lambda^B_{\mathbf{r}} \cap \mathsf{intCone}(D)$. Our goal is to understand that $\mathcal{Z}$ can be expressed as the points of $\Lambda^B_{\mathbf{r}}$ that are contained in some polyhedron $\mathcal{Q} = \mathcal{Q}_{\mathbf{r}}$.

The first step is to understand $\mathsf{cone}(D)$ itself as a polyhedron. This understanding is provided by a classic theorem of Weyl [42]: given D, one can compute a set of integer vectors $\mathcal{F} \subseteq \mathbb{Z}^k$ such that

$$\mathsf{cone}(D) = \{\mathbf{v} \in \mathbb{R}^k \mid \langle \mathbf{f}, \mathbf{v} \rangle \geqslant 0 \text{ for all } \mathbf{f} \in \mathcal{F}\}.$$

Here, $\langle \cdot, \cdot \rangle$ denotes the scalar product in $\mathbb{R}^k$. We will identify vectors $\mathbf{f} \in \mathcal{F}$ with their associated linear functionals $\mathbf{v} \mapsto \langle \mathbf{f}, \mathbf{v} \rangle$. Thus, $\mathsf{cone}(D)$ comprises all vectors $\mathbf{v}$ that have nonnegative evaluations on all functionals in $\mathcal{F}$. It is instructive to also think of the elements of $\mathcal{F}$ as of the facets of $\mathsf{cone}(D)$ understood as a polyhedron, where the functional associated with $\mathbf{f} \in \mathcal{F}$ measures the distance from the corresponding facet.

Recall that in the context of Theorem 2.1, we consider vectors of $\Lambda^B_{\mathbf{r}}$, that is, vectors $\mathbf{v} \in \mathbb{Z}^k$ such that $\mathbf{v} \equiv \mathbf{r} \bmod B$. Then $\langle \mathbf{f}, \mathbf{v} \rangle \equiv \langle \mathbf{f}, \mathbf{r} \rangle \bmod B$ for every $\mathbf{f} \in \mathcal{F}$, hence we can find a unique integer $p_{\mathbf{f}} \in \{0, 1, \ldots, B-1\}$, $p_{\mathbf{f}} \equiv \langle \mathbf{f}, \mathbf{r} \rangle \bmod B$, such that $\langle \mathbf{f}, \mathbf{v} \rangle \equiv p_{\mathbf{f}} \bmod B$ for all $\mathbf{v} \in \Lambda^B_{\mathbf{r}}$. Now $\langle \mathbf{f}, \mathbf{v} \rangle$ is also nonnegative provided $\mathbf{v} \in \mathsf{cone}(D)$, hence

$$\langle \mathbf{f}, \mathbf{v} \rangle \in \{p_{\mathbf{f}}, p_{\mathbf{f}} + B, p_{\mathbf{f}} + 2B, \ldots\} \qquad \text{for all } \mathbf{f} \in \mathcal{F} \text{ and } \mathbf{v} \in \Lambda^B_{\mathbf{r}} \cap \mathsf{cone}(D).$$

Now comes the key distinction about the behavior of $\mathbf{v} \in \Lambda^B_{\mathbf{r}} \cap \mathsf{cone}(D)$ with respect to $\mathbf{f} \in \mathcal{F}$: we say that $\mathbf{f}$ is *tight* with respect to $\mathbf{v}$ if $\langle \mathbf{f}, \mathbf{v} \rangle = p_{\mathbf{f}}$, and is *not tight* otherwise, that is, if $\langle \mathbf{f}, \mathbf{v} \rangle \geqslant p_{\mathbf{f}} + B$. Recall that in the context of Theorem 2.1, we are eventually free to choose B to be large enough. Intuitively, this means that if $\mathbf{f}$ is not tight for $\mathbf{v}$, then $\mathbf{v}$ lies far from the facet corresponding to $\mathbf{f}$ and there is a very large slack in the constraint posed by $\mathbf{f}$ understood as a functional. On the other hand, if $\mathbf{f}$ is tight with respect to $\mathbf{v}$, then $\mathbf{v}$ is close to the boundary of $\mathsf{cone}(D)$ at the facet corresponding to $\mathbf{f}$, and there is a potential danger of observing Frobenius-like aberrations at $\mathbf{v}$.

Thus, the set $\mathcal{R} := \Lambda^B_{\mathbf{r}} \cap \mathsf{cone}(D)$ can be partitioned into subsets $\{\mathcal{R}_{\mathcal{G}} : \mathcal{G} \subseteq \mathcal{F}\}$ defined as follows: $\mathcal{R}_{\mathcal{G}}$ comprises all vectors $\mathbf{v} \in \mathcal{R}$ such that $\mathcal{G}$ is exactly the set of functionals $\mathbf{f} \in \mathcal{F}$ that are tight with respect to $\mathbf{v}$. Our goal is to prove that each set $\mathcal{R}_{\mathcal{G}}$ behaves uniformly with respect to $\mathcal{Z}$: it is either completely disjoint or completely contained in $\mathcal{Z}$. To start the discussion, let us look at the particular case of $\mathcal{R}_{\mathcal{G}}$ for $\mathcal{G} = \emptyset$. These are vectors that are deep inside $\mathsf{cone}(D)$, for which no functional in $\mathcal{F}$ is tight. For these vectors, we use the following lemma, which is the cornerstone of our proof.

Lemma 2.1. (Deep-in-the-Cone Lemma, simplified version) *There exists a constant M, depending only on D, such that the following holds. Suppose $\mathbf{v} \in \mathsf{cone}(D) \cap \mathbb{Z}^k$ is such that $\langle \mathbf{f}, \mathbf{v} \rangle > M$ for all $\mathbf{f} \in \mathcal{F}$. Then $\mathbf{v} \in \mathsf{intCone}(D)$ if and only if $\mathbf{v} \in \mathsf{lattice}(D)$.*

Proof. The left-to-right implication is obvious, hence let us focus on the right-to-left implication. Suppose then that $\mathbf{v} \in \mathsf{lattice}(D)$.

Let $\mathbf{w} = \sum_{\mathbf{d} \in D} L \cdot \mathbf{d}$, where the summation is over the columns of D and L is a positive integer to be fixed later. Observe that for every $\mathbf{f} \in \mathcal{F}$, we have $\langle \mathbf{f}, \mathbf{v} - \mathbf{w} \rangle > M - L \cdot \sum_{\mathbf{d} \in D} \langle \mathbf{f}, \mathbf{d} \rangle$. Therefore, if we choose M to be not smaller than $L \cdot \max_{\mathbf{f} \in \mathcal{F}} \|\mathbf{f}\|_1 \cdot \|D\|_\infty$, then we are certain that $\langle \mathbf{f}, \mathbf{v} - \mathbf{w} \rangle \geqslant 0$ for all $\mathbf{f} \in \mathcal{F}$, and hence $\mathbf{v} - \mathbf{w} \in \mathsf{cone}(D)$. Consequently, we can write $\mathbf{v} - \mathbf{w} = D\mathbf{y}$ for some $\mathbf{y} \in \mathbb{R}^t_{\geqslant 0}$. Let $\mathbf{y}' \in \mathbb{Z}^t_{\geqslant 0}$ be such that $y'_i = \lfloor y_i \rfloor$ for all $i \in \{1, \ldots, t\}$, and let $\mathbf{v}' = \mathbf{w} + D\mathbf{y}'$. Then

$$\|\mathbf{v} - \mathbf{v}'\|_\infty = \|D(\mathbf{y} - \mathbf{y}')\|_\infty \leqslant t \cdot \|D\|_\infty.$$

On the other hand, we clearly have $\mathbf{v}' \in \mathsf{intCone}(D)$ and by assumption, $\mathbf{v} \in \mathsf{lattice}(D)$. It follows that $\mathbf{v} - \mathbf{v}' \in \mathsf{lattice}(D)$. From standard bounds, see e.g. [39], it follows that there exists $\mathbf{z} \in \mathbb{Z}^t$ with $\mathbf{v} - \mathbf{v}' = D\mathbf{z}$ such that $\|\mathbf{z}\|_1$ is bounded by a function of D and $\|\mathbf{v} - \mathbf{v}'\|_\infty$, which in turn is again bounded by a function of D as explained above. (Note here that t is the number of columns of D, hence it also depends only on D.) This means that if we choose L large enough depending on D, we are certain that $\|\mathbf{z}\|_1 \leqslant L$. Now, it remains to observe that

$$\mathbf{v} = \mathbf{w} + D\mathbf{y}' + (\mathbf{v} - \mathbf{v}') = D(L \cdot \mathbf{1} + \mathbf{y}' + \mathbf{z}),$$

where $\mathbf{1}$ denotes the vector of t ones, and that all the entries of $L \cdot \mathbf{1} + \mathbf{y}' + \mathbf{z}$ are nonnegative integers. This proves that $\mathbf{v} \in \mathsf{intCone}(D)$. ∎

We remark that the statement of Lemma 2.1 actually follows from results present in the literature, concerning the notion of *diagonal Frobenius numbers*. See the work of Aliev and Henk [1] for a broader discussion and pointers to earlier works. As we will discuss in a moment, in this work we actually use a generalization of Lemma 2.1.

Consider any $\mathbf{u}, \mathbf{v} \in \mathcal{R}$. Since all the entries of $\mathbf{u} - \mathbf{v}$ are divisible by B, it is not hard to prove the following: if we choose B to be a large enough factorial, then $\mathbf{u} \in \mathsf{lattice}(D)$ if and only if $\mathbf{v} \in \mathsf{lattice}(D)$. Hence, from Lemma 2.1 it follows that $\mathcal{R}_\emptyset$ is either entirely disjoint or entirely contained in $\mathcal{Z}$.

A more involved reasoning based on the same fundamental ideas, but using a generalization of Lemma 2.1, yields the following lemma, which tackles also the case when some functionals of $\mathcal{F}$ are tight with respect to the considered vectors.

Lemma 2.2. *Suppose $\mathbf{u}, \mathbf{v} \in \mathcal{R}$ are such that for every $\mathbf{f} \in \mathcal{F}$, if $\mathbf{f}$ is tight with respect to $\mathbf{u}$, then $\mathbf{f}$ is also tight with respect to $\mathbf{v}$. Then $\mathbf{u} \in \mathcal{Z}$ implies $\mathbf{v} \in \mathcal{Z}$.*

We remark that the proof of Lemma 2.2 actually requires more work and more ideas than those presented in the proof of Lemma 2.1. In essence, one needs to partition functionals that are tight with respect to $\mathbf{u}$ into those that are very tight (have very small $p_\mathbf{f}$) and those that are only slightly tight (have relatively large $p_\mathbf{f}$) in order to create a sufficient gap between very tight and slightly tight functionals. Having achieved this, a delicate variant of the reasoning from the proof of Lemma 2.1 can be applied. It is important that whenever a functional $\mathbf{f} \in \mathcal{F}$ is tight with respect to both $\mathbf{u}$ and $\mathbf{v}$, we actually know that $\langle \mathbf{f}, \mathbf{u} \rangle = \langle \mathbf{f}, \mathbf{v} \rangle = p_\mathbf{f}$. Note that this is exactly the benefit achieved by restricting attention to the vectors of $\Lambda^B_\mathbf{r}$.

Using Lemma 2.2, we can immediately describe how the structure of $\mathcal{Z}$ relates to that of $\mathcal{R}$.

Corollary 2.1. *For every $\mathcal{G} \subseteq \mathcal{F}$, either $\mathcal{R}_\mathcal{G} \cap \mathcal{Z} = \emptyset$ or $\mathcal{R}_\mathcal{G} \subseteq \mathcal{Z}$. Moreover, if $\mathcal{R}_\mathcal{G} \subseteq \mathcal{Z}$ and $\mathcal{R}_\mathcal{G}$ is non-empty, then $\mathcal{R}_{\mathcal{G}'} \subseteq \mathcal{Z}$ for all $\mathcal{G}' \subseteq \mathcal{G}$.*

Corollary 2.1 suggests now how to define the polyhedron $\mathcal{Q}$. Namely, $\mathcal{Q}$ is defined as the set of all $\mathbf{v} \in \mathbb{R}^k$ satisfying the following linear inequalities:

- inequalities $\langle \mathbf{f}, \mathbf{v} \rangle \geqslant 0$ for all $\mathbf{f} \in \mathcal{F}$ that define $\mathsf{cone}(D)$; and
- for every $\mathcal{G} \subseteq \mathcal{F}$ such that $\mathcal{R}_\mathcal{G}$ is nonempty and $\mathcal{R}_\mathcal{G} \cap \mathcal{Z} = \emptyset$, the inequality

$$\sum_{\mathbf{g} \in \mathcal{G}} \langle \mathbf{g}, \mathbf{v} \rangle \geqslant 1 + \sum_{\mathbf{g} \in \mathcal{G}} p_\mathbf{g}.$$

In essence, the inequalities from the second point "carve out" those parts $\mathcal{R}_{\mathcal{G}}$ that should not be included in $\mathcal{Z}$. We note that computing the inequalities defining $\mathcal{Q}$ requires solving several auxiliary integer programs to figure out for which $\mathcal{G} \subseteq \mathcal{F}$ the corresponding inequality should be included.

It is now straightforward to verify, using all the accumulated observations, that indeed $\mathcal{Z} = \mathcal{R} \cap \mathcal{Q}$ as required. This concludes a sketch of the proof of Theorem 2.1.

2.2 n-fold programming. We now give an overview of the proof of Theorem 1.2. For simplicity, we make the following assumptions.

- We focus on the feasibility problem instead of optimization. At the very end, we will remark on what additional ideas are needed to also tackle the optimization problem.
- We assume that all the diagonal blocks D_i are equal: $D_i = D$ for all $i \in \{1, \ldots, n\}$, where D is a $k \times k$ integer matrix with $\|D\|_\infty \leqslant \Delta$. This is only a minor simplification because there are only $(2\Delta+1)^{k^2}$ different matrices D_i with $\|D_i\|_\infty \leqslant \Delta$, and in the general case, we simply treat every such possible matrix "type" separately using the reasoning from the simplified case.

Breaking up bricks. Basic components of the given n-fold program $P = (C, D, \mathbf{a}, \mathbf{b}_i : i \in \{1, \ldots, n\})$ are *bricks*: programs $D\mathbf{y}_i = \mathbf{b}_i$ for $i \in \{1, \ldots, n\}$ that encode local constraints on the variables $\mathbf{y}_i$. While the entries of D are bounded in absolute values by the parameter Δ, we do not assume any bound on the entries of vectors $\mathbf{b}_i$. This poses an issue, as different bricks may possibly have very different behaviors.

The key idea in our approach is to simplify the program P by iteratively breaking up every brick $D\mathbf{y} = \mathbf{b}$ into two bricks $D\mathbf{y} = \mathbf{b}'$ and $D\mathbf{y} = \mathbf{b}''$ with strictly smaller right-hand sides $\mathbf{b}', \mathbf{b}''$, until eventually, we obtain an equivalent n-fold program P' in which all right-hand sides have ℓ_∞-norms bounded in terms of the parameters. The following lemma is the crucial new piece of technology used in our proof. (Here, we use the *conformal order* on $\mathbb{Z}^k$: we write $\mathbf{u} \sqsubseteq \mathbf{v}$ if $|\mathbf{u}[i]| \leqslant |\mathbf{v}[i]|$ and $\mathbf{u}[i] \cdot \mathbf{v}[i] \geqslant 0$ for all $i \in \{1, \ldots, k\}$.)

LEMMA 2.3. (BRICK DECOMPOSITION LEMMA) *There exists a function $g(k, \Delta) \in 2^{(k\Delta)^{\mathcal{O}(k)}}$ such that the following holds. Let D be an integer matrix with t columns and k rows and all absolute values of its entries bounded by Δ. Further, let $\mathbf{b} \in \mathbb{Z}^k$ be an integer vector such that $\|\mathbf{b}\|_\infty > g(k, \Delta)$. Then there are non-zero vectors $\mathbf{b}', \mathbf{b}'' \in \mathbb{Z}^k$ such that:*

- *$\mathbf{b}', \mathbf{b}'' \sqsubseteq \mathbf{b}$ and $\mathbf{b} = \mathbf{b}' + \mathbf{b}''$; and*
- *for every $\mathbf{v} \in \mathbb{Z}^{\mathbf{y}}_{\geqslant 0}$ satisfying $D\mathbf{v} = \mathbf{b}$, there exist $\mathbf{v}', \mathbf{v}'' \in \mathbb{Z}^{\mathbf{y}}_{\geqslant 0}$ such that*

$$\mathbf{v} = \mathbf{v}' + \mathbf{v}'', \qquad D\mathbf{v}' = \mathbf{b}', \qquad \textit{and} \qquad D\mathbf{v}'' = \mathbf{b}''.$$

In other words, Lemma 2.3 states that the brick $D\mathbf{y} = \mathbf{b}$ can be broken into two new bricks $D\mathbf{y}' = \mathbf{b}'$ and $D\mathbf{y}'' = \mathbf{b}''$ with conformally strictly smaller $\mathbf{b}', \mathbf{b}''$ so that *every* potential solution $\mathbf{v}$ to $D\mathbf{y} = \mathbf{b}$ can be decomposed into solutions $\mathbf{v}', \mathbf{v}''$ to the two new bricks. It is easy to see that this condition implies that in P, we may replace the brick $D\mathbf{y} = \mathbf{b}$ with $D\mathbf{y}' = \mathbf{b}'$ and $D\mathbf{y}'' = \mathbf{b}''$ without changing feasibility or, in the case of the optimization problem, the minimum value of the optimization goal. In the latter setting, both new bricks inherit the optimization vector $\mathbf{c}_i$ from the original brick.

Before we continue, let us comment on the proof of Lemma 2.3. We use two ingredients. The first one is the following fundamental result of Klein [26]. (Here, for a multiset of vectors A, by $\sum A$ we denote the sum of all the vectors in A.)

LEMMA 2.4. (KLEIN LEMMA, VARIANT FROM [11]) *Let $T_1, \ldots, T_n$ be non-empty multisets of vectors in $\mathbb{Z}^k$ such that $\sum T_1 = \sum T_2 = \ldots = \sum T_n$ and all vectors contained in all multisets $T_1, \ldots, T_n$ have ℓ_∞-norm bounded by Δ. Then there are non-empty multisets $S_1 \subseteq T_1, \ldots, S_n \subseteq T_n$, each of size at most $2^{\mathcal{O}(k\Delta)^k}$, such that $\sum S_1 = \sum S_2 = \ldots = \sum S_n$.*

In the context of the proof of Lemma 2.3, we apply Lemma 2.4 to the family of all multisets T that consist of columns of D and satisfy $\sum T = \mathbf{b}$. By encoding multiplicities, such multisets correspond to vectors $\mathbf{v} \in \mathbb{Z}^k_{\geqslant 0}$ satisfying $D\mathbf{v} = \mathbf{b}$. (We hide here some technicalities regarding the fact that this family is infinite.) By Lemma 2.4, from each such multiset T, we can extract a submultiset S of bounded size such that all the submultisets S sum up to the same vector $\mathbf{b}'$. Denoting $\mathbf{b}'' = \mathbf{b} - \mathbf{b}'$, this means that every vector $\mathbf{v} \in \mathbb{Z}^k_{\geqslant 0}$ satisfying $D\mathbf{v} = \mathbf{b}$ can be decomposed as $\mathbf{v} = \mathbf{v}' + \mathbf{v}''$ with $\mathbf{v}', \mathbf{v}'' \in \mathbb{Z}^k_{\geqslant 0}$ so that $D\mathbf{v}' = \mathbf{b}'$ and $D\mathbf{v}'' = \mathbf{b}''$. Namely, $\mathbf{v}'$ corresponds to the vectors contained in S and $\mathbf{v}''$ corresponds to the vectors contained in $T - S$, where T is the multiset corresponding to $\mathbf{v}$.

There is an issue in the above reasoning: we do not obtain the property $\mathbf{b}', \mathbf{b}'' \sqsubseteq \mathbf{b}$, which will be important in later applications of Lemma 2.3. To bridge this difficulty, we apply the argument above exhaustively to decompose $\mathbf{b}$ as $\mathbf{b}_1 + \ldots + \mathbf{b}_m$, for some integer m, so that every vector $\mathbf{b}_i$ has the ℓ_∞-norm bounded by $2^{\mathcal{O}(k\Delta)^k}$ and every vector $\mathbf{v} \in \mathbb{Z}^k_{\geqslant 0}$ satisfying $D\mathbf{v} = \mathbf{b}$ can be decomposed as $\mathbf{v} = \mathbf{v}_1 + \ldots + \mathbf{v}_m$ where $\mathbf{v}_i \in \mathbb{Z}^k_{\geqslant 0}$ satisfies $D\mathbf{v}_i = \mathbf{b}_i$. Then, we treat vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m$ with the following lemma.

Lemma 2.5. *Let $\mathbf{u}_1, \ldots, \mathbf{u}_m$ be vectors in $\mathbb{Z}^k$ of ℓ_∞-norm bounded by Ξ, and let $\mathbf{b} = \sum_{i=1}^m \mathbf{u}_i$. Then the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ can be grouped into non-empty groups $U_1, \ldots, U_\ell$, each of size at most $\mathcal{O}(\Delta)^{2^{k-1}}$, so that $\sum U_i \sqsubseteq \mathbf{b}$ for all $i = 1, \ldots, \ell$.*

More precisely, Lemma 2.5 allows us to group vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m$ into groups of bounded size so that the sum within each group is sign-compatible with $\mathbf{b}$. Assuming $\|\mathbf{b}\|_\infty$ is large enough, there will be at least two groups. Then, any non-trivial partition of the groups translates into a suitable decomposition $\mathbf{b} = \mathbf{b}' + \mathbf{b}''$ with $\mathbf{b}', \mathbf{b}'' \sqsubseteq \mathbf{b}$.

The proof of Lemma 2.5 is by induction on k and uses arguments similar to standard proofs of Steinitz Lemma. This concludes a sketch of the proof of Lemma 2.3.

Once Lemma 2.3 is established, it is natural to use it iteratively: break $\mathbf{b}$ into $\mathbf{b}', \mathbf{b}''$, then break $\mathbf{b}'$ into two even smaller vectors, and so on. By applying the argument exhaustively, eventually we obtain a collection of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m \sqsubseteq \mathbf{b}$ such that $\mathbf{b} = \mathbf{b}_1 + \ldots + \mathbf{b}_m$, $\|\mathbf{b}_i\|_\infty \leqslant 2^{(k\Delta)^{\mathcal{O}(k)}}$ for all $i \in \{1, \ldots, m\}$, and every $\mathbf{v} \in \mathbb{Z}^k_{\geqslant 0}$ satisfying $D\mathbf{v} = \mathbf{b}$ can be decomposed as $\mathbf{v} = \mathbf{v}_1 + \ldots + \mathbf{v}_m$ with $\mathbf{v}_i \in \mathbb{Z}^k_{\geqslant 0}$ and $D\mathbf{v}_i = \mathbf{b}_i$ for all $i \in \{1, \ldots, m\}$. We call such a collection a *faithful decompostion* of $\mathbf{b}$ of *order* $2^{(k\Delta)^{\mathcal{O}(k)}}$.

There is an important technical caveat here. Observe that the size m of a faithful decomposition of a right-hand side $\mathbf{b}$ can be as large as $\Omega(\|\mathbf{b}\|_1)$, which is *exponential* in the bitsize of the program P. So we cannot hope to compute a faithful decomposition explicitly within the target time complexity. However, observe that all vectors $\mathbf{b}_i$ in a faithful decomposition $\mathcal{B}$ are bounded in ℓ_∞-norm by $\Xi \coloneqq 2^{(k\Delta)^{\mathcal{O}(k)}}$, and there are only at most $(2\Xi + 1)^k$ different such vectors. Therefore, $\mathcal{B}$ can be encoded by storing, for each vector $\mathbf{b}'$ present in $\mathcal{B}$, the multiplicity of $\mathbf{b}'$ in $\mathcal{B}$. Thus, describing $\mathcal{B}$ takes $2^{(k\Delta)^{\mathcal{O}(k)}} \cdot \log \|\mathbf{b}\|_\infty$ bits.

With this encoding scheme in mind, we show that a faithful decomposition $\mathcal{B}$ of a given vector $\mathbf{b}$ of order at most Ξ can be computed in fixed-parameter time $f(\Delta, k) \cdot (\log \|\mathbf{b}\|_\infty)^{\mathcal{O}(1)}$, for a computable function f. For this, we show that one can extract parts of the decomposition in "larger chunks", at each step reducing the ℓ_1-norm of the decomposed vector by a constant fraction; this gives a total number of steps logarithmic in $\|\mathbf{b}\|_1$. In each step, to extract the next large chunk of the decomposition, we use the fixed-parameter algorithm for optimization problems definable in Presburger arithmetic, due to Koutecký and Talmon [36]. We remark that in our context, this tool could be also replaced by the fixed-parameter algorithm of Eisenbrand and Shmonin [17] for $\forall\exists$ integer programming.

Reduction to (mixed) integer programming with few variables. With faithful decompositions understood, we can compute, for every right-hand side $\mathbf{b}_i$ part of P, a faithful decomposition $\{\mathbf{b}_i^1, \ldots, \mathbf{b}_i^{m_i}\}$ of $\mathbf{b}_i$. This allows us to construct an equivalent (in terms of feasibility and optimization) n-fold program P' by replacing each brick $D\mathbf{y}_i = \mathbf{b}_i$ with bricks $D\mathbf{y}_i^j = \mathbf{b}_i^j$ for $j \in \{1, \ldots, m_i\}$. Thus, the program P' has an exponential number of bricks, but can be computed and described concisely: all right-hand sides are bounded in the ℓ_∞-norm by at most Ξ, so for every potential right-hand side $\mathbf{b}$, we just write the multiplicity in which $\mathbf{b}$ appears in P'. We remark that such *high-multiplicity encoding* of n-fold integer programs has already been studied by Knop et al. [31].

For convenience, let $\mathsf{RHS} \coloneqq \{-\Xi, \ldots, \Xi\}^k$ be the set of all possible right-hand sides, and for $\mathbf{b} \in \mathsf{RHS}$, by $\mathsf{count}[\mathbf{b}]$ we denote the multiplicity of $\mathbf{b}$ in P'.

It is now important to better understand the set of solutions to a single brick $D\mathbf{y} = \mathbf{b}$ present in P'. Here comes a key insight stemming from the theory of Graver bases: as (essentially) proved by Pottier [39], every solution $\mathbf{w} \in \mathbb{Z}^k_{\geqslant 0}$ to $D\mathbf{w} = \mathbf{b}$ can be decomposed as $\mathbf{w} = \widehat{\mathbf{w}} + \mathbf{g}_1 + \ldots + \mathbf{g}_\ell$, where

- $\widehat{\mathbf{w}} \in \mathbb{Z}^k_{\geqslant 0}$ is a *base solution* that also satisfies $D\widehat{\mathbf{w}} = \mathbf{b}$, but $\|\widehat{\mathbf{w}}\|_\infty$ is bounded by a function of Δ and $\|\mathbf{b}\|_\infty$, and

- $\mathbf{g}_1, \ldots, \mathbf{g}_\ell \in \mathbb{Z}^k_{\geqslant 0}$ are elements of the Graver basis of D.

Here, the *Graver basis* of D consists of all conformally-minimal non-zero vectors $\mathbf{g}$ satisfying $D\mathbf{g} = \mathbf{0}$. In particular, it is known that the Graver basis is always finite and consists of vectors of ℓ_∞ norm bounded by $(2k\Delta + 1)^k$ [15]. The decomposition explained above will be called a *Graver decomposition* of $\mathbf{w}$.

For $\mathbf{b} \in \mathsf{RHS}$, let $\mathsf{Base}[\mathbf{b}]$ be the set of all possible base solutions $\widehat{\mathbf{w}}$ to $D\mathbf{y} = \mathbf{b}$. As $\|\mathbf{b}\|_\infty \leqslant \Xi$ and Ξ is bounded by a function of the parameters under consideration, it follows that $\mathsf{Base}[\mathbf{b}]$ consists only of vectors of bounded ℓ_∞-norms, and therefore it can be efficiently constructed.

Having this understanding, we can write an integer program M with few variables that is equivalent to P'. The variables are as follows:

- for every $\mathbf{b} \in \mathsf{RHS}$ and $\widehat{\mathbf{w}} \in \mathsf{Base}[\mathbf{b}]$, we introduce a variable $\zeta^{\mathbf{b}}_{\widehat{\mathbf{w}}} \in \mathbb{Z}_{\geqslant 0}$ that signifies how many times in total $\widehat{\mathbf{w}}$ is used in the Graver decompositions of solutions to individual bricks.
- for every nonnegative vector $\mathbf{g}$ in the Graver basis of D, we introduce a variable $\delta_{\mathbf{g}} \in \mathbb{Z}_{\geqslant 0}$ signifying how many times in total $\mathbf{g}$ appears in the Graver decompositions of solutions to individual bricks.

Note that since program P' is uniform, the guessed base solutions and elements of the Graver basis can be assigned to *any* brick with the same effect on the linking constraints of P'. Hence, it suffices to verify the cardinalities and the total effect on the linking constrains of P', yielding the following constraints of M:

- the translated linking constraints: $\sum_{\mathbf{b} \in \mathsf{RHS}} \sum_{\widehat{\mathbf{w}} \in \mathsf{Base}[\mathbf{b}]} \zeta^{\mathbf{b}}_{\widehat{\mathbf{w}}} \cdot C\widehat{\mathbf{w}} + \sum_{\mathbf{g} \in \mathsf{Graver}(D), \mathbf{g} \geqslant 0} \delta_{\mathbf{g}} \cdot C\mathbf{g} = \mathbf{a}$.
- for every $\mathbf{b} \in \mathsf{RHS}$, the cardinality constraint $\sum_{\widehat{\mathbf{w}} \in \mathsf{Base}[\mathbf{b}]} \zeta^{\mathbf{b}}_{\widehat{\mathbf{w}}} = \mathsf{count}[\mathbf{b}]$.

Noting that the number of variables of M is bounded in terms of the parameters, we may apply any fixed-parameter algorithm for integer programming parameterized by the number of variables, for instance that of Kannan [25], to solve M. This concludes the description of the algorithm for the feasibility problem.

In the case of the optimization problem, there is an issue that the optimization vectors $\mathbf{c}_i$ may differ between different bricks, and there may be as many as n different such vectors. While the Graver basis elements can be always greedily assigned to bricks in which their contribution to the optimization goal is the smallest, this is not so easy for the base solutions, as every brick may accommodate only one base solution. We may enrich M by suitable assignment variables $\omega^{\mathbf{b},i}_{\widehat{\mathbf{w}}}$ to express how many base solutions of each type are assigned to bricks with different optimization vectors; but this yields as many as $\Omega(n)$ additional variables. Fortunately, we observe that in the enriched program M, if one fixes any integral valuation of variables $\zeta^{\mathbf{b}}_{\widehat{\mathbf{w}}}$ and $\delta_{\mathbf{g}}$, the remaining problem on variables $\omega^{\mathbf{b},i}_{\widehat{\mathbf{w}}}$ corresponds to a flow problem, and hence its constraint matrix is totally unimodular. Thus, we may solve M as a mixed integer program where variables $\omega^{\mathbf{b},i}_{\widehat{\mathbf{w}}}$ are allowed to be fractional. The number of integral variables is bounded in terms of parameters, so we may apply the fixed-parameter algorithm for mixed integer programming of Lenstra [38].

References

[1] I. Aliev and M. Henk. Feasibility of integer knapsacks. *SIAM J. Optim.*, 20(6):2978–2993, 2010.

[2] M. Aschenbrenner and R. Hemmecke. Finiteness theorems in stochastic integer programming. *Found. Comput. Math.*, 7(2):183–227, 2007.

[3] M. Briański, M. Koutecký, D. Král', K. Pekárková, and F. Schröder. Characterization of matrices with bounded Graver bases and depth parameters and applications to integer programming. In *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022*, volume 229 of *LIPIcs*, pages 29:1–29:20. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2022.

[4] T. F. N. Chan, J. W. Cooper, M. Koutecký, D. Král, and K. Pekárková. Matrices of optimal tree-depth and a row-invariant parameterized algorithm for integer programming. *SIAM J. Comput.*, 51(3):664–700, 2022.

[5] H. Chen, L. Chen, and G. Zhang. FPT algorithms for a special block-structured integer program with applications in scheduling. *CoRR*, abs/2107.01373, 2021.

[6] H. Chen, L. Chen, and G. Zhang. Block-structured integer programming: Can we parameterize without the largest coefficient? *Discret. Optim.*, 46:100743, 2022.

[7] L. Chen, M. Koutecký, L. Xu, and W. Shi. New bounds on augmenting steps of block-structured integer programs. In *28th Annual European Symposium on Algorithms, ESA 2020*, volume 173 of *LIPIcs*, pages 33:1–33:19. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2020.

[8] L. Chen and D. Marx. Covering a tree with rooted subtrees — parameterized and approximation algorithms. In *29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 2801–2820. SIAM, 2018.

[9] L. Chen, D. Marx, D. Ye, and G. Zhang. Parameterized and approximation results for scheduling with a low rank processing time matrix. In *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017*, volume 66 of *LIPIcs*, pages 22:1–22:14. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2017.
[10] J. Cslovjecsek, F. Eisenbrand, C. Hunkenschröder, L. Rohwedder, and R. Weismantel. Block-structured integer and linear programming in strongly polynomial and near linear time. In *32nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, pages 1666–1681. SIAM, 2021.
[11] J. Cslovjecsek, F. Eisenbrand, M. Pilipczuk, M. Venzin, and R. Weismantel. Efficient sequential and parallel algorithms for multistage stochastic integer programming using proximity. In *29th Annual European Symposium on Algorithms, ESA 2021*, volume 204 of *LIPIcs*, pages 33:1–33:14. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2021.
[12] D. Dadush. *Integer Programming, Lattice Algorithms, and Deterministic Volume Estimation.* PhD thesis, Georgia Institute of Technology, USA, 2012.
[13] D. Dadush, F. Eisenbrand, and T. Rothvoss. From approximate to exact integer programming. In *24th International Conference on Integer Programming and Combinatorial Optimization, IPCO 2023*, volume 13904 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 2023.
[14] J. A. De Loera, R. Hemmecke, S. Onn, and R. Weismantel. N-fold integer programming. *Discrete Optimization*, 5(2):231–241, 2008. In Memory of George B. Dantzig.
[15] F. Eisenbrand, C. Hunkenschröder, and K. Klein. Faster algorithms for integer programs with block structure. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 49:1–49:13. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2018.
[16] F. Eisenbrand, C. Hunkenschröder, K. Klein, M. Koutecký, A. Levin, and S. Onn. An algorithmic theory of integer programming. *CoRR*, abs/1904.01361, 2019.
[17] F. Eisenbrand and G. Shmonin. Parametric integer programming in fixed dimension. *Math. Oper. Res.*, 33(4):839–850, 2008.
[18] A. Frank and É. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.
[19] T. Gavenčiak, M. Koutecký, and D. Knop. Integer programming in parameterized complexity: Five miniatures. *Discret. Optim.*, 44(Part):100596, 2022.
[20] R. Hemmecke, M. Köppe, and R. Weismantel. Graver basis and proximity techniques for block-structured separable convex integer minimization problems. *Math. Program.*, 145(1-2):1–18, 2014.
[21] R. Hemmecke, S. Onn, and L. Romanchuk. n-fold integer programming in cubic time. *Math. Program.*, 137(1-2):325–341, 2013.
[22] K. Jansen, K. Klein, and A. Lassota. The double exponential runtime is tight for 2-stage stochastic ILPs. *Math. Program.*, 197(2):1145–1172, 2023.
[23] K. Jansen, K. Klein, M. Maack, and M. Rau. Empowering the configuration-IP: new PTAS results for scheduling with setup times. *Math. Program.*, 195(1):367–401, 2022.
[24] K. Jansen, A. Lassota, and L. Rohwedder. Near-linear time algorithm for n-fold ILPs via color coding. *SIAM J. Discret. Math.*, 34(4):2282–2299, 2020.
[25] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.
[26] K. Klein. About the complexity of two-stage stochastic IPs. *Math. Program.*, 192(1):319–337, 2022.
[27] K. Klein and J. Reuter. Collapsing the tower — On the complexity of multistage stochastic IPs. In *33rd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*, pages 348–358. SIAM, 2022.
[28] D. Knop and M. Koutecký. Scheduling meets n-fold integer programming. *J. Sched.*, 21(5):493–503, 2018.
[29] D. Knop and M. Koutecký. Scheduling kernels via configuration LP. In *30th Annual European Symposium on Algorithms, ESA 2022*, volume 244 of *LIPIcs*, pages 73:1–73:15. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2022.
[30] D. Knop, M. Koutecký, A. Levin, M. Mnich, and S. Onn. Parameterized complexity of configuration integer programs. *Oper. Res. Lett.*, 49(6):908–913, 2021.
[31] D. Knop, M. Koutecký, A. Levin, M. Mnich, and S. Onn. High-multiplicity N-fold IP via configuration LP. *Math. Program.*, 200(1):199–227, 2023.
[32] D. Knop, M. Koutecký, and M. Mnich. Combinatorial n-fold integer programming and applications. *Math. Program.*, 184(1):1–34, 2020.
[33] D. Knop, M. Koutecký, and M. Mnich. Voting and bribing in single-exponential time. *ACM Trans. Economics and Comput.*, 8(3):12:1–12:28, 2020.
[34] D. Knop, M. Pilipczuk, and M. Wrochna. Tight complexity lower bounds for integer linear programming with few constraints. *ACM Trans. Comput. Theory*, 12(3):19:1–19:19, 2020.
[35] M. Koutecký, A. Levin, and S. Onn. A parameterized strongly polynomial algorithm for block structured integer

programs. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 85:1–85:14. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2018.
[36] M. Koutecký and N. Talmon. Multi-party campaigning. In *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 5506–5513. AAAI Press, 2021.
[37] M. Koutecký and J. Zink. Complexity of scheduling few types of jobs on related and unrelated machines. In *31st International Symposium on Algorithms and Computation, ISAAC 2020*, volume 181 of *LIPIcs*, pages 18:1–18:17. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2020.
[38] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
[39] L. Pottier. Minimal solutions of linear diophantine systems: Bounds and algorithms. In *4th International Conference on Rewriting Techniques and Applications, RTA-91*, volume 488 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 1991.
[40] V. Reis and T. Rothvoss. The subspace flatness conjecture and faster integer programming. *CoRR*, abs/2303.14605, 2023.
[41] R. Schultz, L. Stougie, and M. H. van der Vlerk. Two-stage stochastic integer programming: a survey. *Statistica Neerlandica*, 50(3):404–416, 1996.
[42] H. Weyl. Elementare Theorie der konvexen Polyeder. *Commentarii Mathematici Helveticii*, 7:290–306, 1935.