Integer Points in a Parameterised Polyhedron

FRIEDRICH EISENBRAND AND GENNADY SHMONIN

Department of Mathematics
University of Paderborn
D-33095 Paderborn
Germany

{eisen,shmonin}@math.uni-paderborn.de

Abstract

The classical parameterised integer feasibility problem is as follows. Given a rational matrix $A \in \mathbb{Q}^{m\times n}$ and a rational polyhedron $Q \subseteq \mathbb{R}^m$, decide, whether there exists a point $b \in Q$ such that $Ax \leqslant b$ is integer infeasible. Our main result is a polynomial algorithm to solve a slightly more general parameterised integer feasibility problem if the number n of columns of A is fixed. This extends a result of Kannan (1992) who provided such an algorithm for the case in which additionally to n, also the affine dimension of the polyhedron Q has to be fixed.
As an application of our result, we describe an algorithm to find the maximum difference between the optimum values of an integer program $\max\{cx : Ax \leqslant b,\, x \in \mathbb{Z}^n\}$ and its linear programming relaxation, as the right-hand side b varies over all vectors, for which the integer program is feasible. The latter is an extension of a recent result of Hoşten and Sturmfels (2003) who presented such an algorithm for integer programs in standard form.

1 Introduction

Central to this paper is the following problem, which we call *parameterised integer feasibility* (*PIF*):

> Given rational matrices $A \in \mathbb{Q}^{m\times n}$ and $B \in \mathbb{Q}^{k\times n}$, rational affine transformations $\Phi : \mathbb{R}^l \to \mathbb{R}^m$ and $\Psi : \mathbb{R}^l \to \mathbb{R}^k$ and a rational polyhedron $Q \subseteq \mathbb{R}^l$, find $b \in Q$ such that the system of linear inequalities $Ax \leqslant \Phi(b)$ has an integer solution, but the system $Bx \leqslant \Psi(b)$ has no integer solution, or assert that no such b exists.

We assume here that Φ and Ψ are given by rational matrices and vectors respectively, i.e., $\Phi(x) = A_\Phi x + b_\Phi$ and $\Psi(x) = A_\Psi x + b_\Psi$, where the matrices A_Φ and A_Ψ and the vectors b_Φ and b_Ψ have suitable dimensions. We also assume that Q is given explicitly by a system of rational inequalities.

If there is no $b \in Q$ satisfying the above requirements, the existence of an integer solution of $Ax \leqslant \Phi(b)$ implies the existence of an integer solution to $Bx \leqslant \Psi(b)$ for all $b \in Q$. Thus a system $B'x \leqslant b'$ is integer infeasible if and only if the following instance of PIF has a solution

$$A = 0, \quad \Phi = 0, \quad B = B', \quad \Psi = I, \quad \text{and} \quad Q = \{b'\},$$

where I denotes the identity transformation. It follows that PIF is at least as hard as the classical integer feasibility problem.

Motivated by the celebrated result of Lenstra (1983), who established a polynomial-time algorithm for integer programming in fixed dimension, we restrict ourselves to the case in which the number of columns of the matrices A and B, hence the number of variables in the systems $Ax \leqslant \Phi(b)$ and $Bx \leqslant \Psi(b)$, is fixed. Our main result is a polynomial algorithm to solve PIF in this case.

Related work

The following is a slightly weaker version of PIF:

> Given a rational matrix $A \in \mathbb{Q}^{m\times n}$ and a rational polyhedron $Q \subseteq \mathbb{R}^m$, find $b \in Q$ such that the system of linear inequalities $Ax \leqslant b$ has no integer solution, or assert that no such b exists.

Equivalently, one needs to check the following statement: "for all $b \in Q$, there exists $x \in \mathbb{Z}^n$ such that $Ax \leqslant b$". This question was considered by Kannan (1990) and Kannan (1992) who presented a polynomial time algorithm to solve this only slightly weaker version of PIF if n *and* the affine dimension of Q are fixed. We generalise the algorithm of Kannan to handle also PIF if n is fixed; the dimension of Q does not need to be fixed anymore. However, we want to emphasise that we adopt many techniques and ideas of Kannan (1990) and Kannan (1992).

We apply our main result to find the maximum integer programming gap for a family of integer programs. The *integer programming gap* of an integer program is the difference between its optimum value and the optimum value of its linear programming relaxation. For a matrix A and an objective vector c, we denote by $g(A,c)$ the maximum integer programming gap of integer programs

$$\max\{cx : Ax \leqslant b\},$$

where b varies over all vectors, for which the integer program is feasible. Our algorithm finds $g(A,c)$ in polynomial time if the rank of A is fixed. This generalises a recent result of Hoşten and Sturmfels (2003), who proposed an algorithm to find $g(A,c)$ for integer programs in *standard form*, i.e.,

$$\max\{cx : Ax = b,\ x \geqslant 0,\ x \in \mathbb{Z}^n\}.$$

Their algorithm exploits short rational generating functions for certain lattice point problems, see Barvinok (1994) and Barvinok and Woods (2003), and runs in polynomial time if the number of columns in A is fixed. The latter implies also a fixed number of rows, as we can always assume A to have full row rank. We would like to point out that our approach does not rely on rational generating functions at all.

Barvinok and Woods (2003) present an algorithm to solve the counting problem for integer projections of integer points in polytopes. Their algorithm uses Kannan's partitioning algorithm, which we extend in this paper. In particular, the algorithm of Barvinok and Woods can be used to count the number of elements of the minimal Hilbert basis of a pointed cone in polynomial time, if the dimension is fixed. A polynomial test for the Hilbert basis property in fixed dimension was first presented by Cook et al. (1984).

Structure of this paper

We continue this section by introducing some basic definitions and notation that we use throughout the paper. Then we briefly describe the algorithm to solve integer programming in fixed dimension; our algorithm can be seen as its adaptation to the case of the varying right-hand side b. In particular, both algorithms exploit the so-called *flatness theorem*, which states that if a polyhedron contains no integer point, then it must be "flat" along some integral direction.

In Section 2 we consider *parameterised polyhedra*, which are defined by systems of linear inequalities $Ax \leqslant b$, where the matrix A is fixed but the right-hand side b is varying over some polyhedron Q. The polyhedra defined by different right-hand sides may have different widths and even different width directions. However we prove that the set of the right-hand sides can be decomposed into polynomially many partially open polyhedra such that the width direction remains the same, as b varies over one partially open polyhedron of the partition. This improves upon a result of Kannan (1992) (Lemma 3.1), since we compute the width directions *exactly* and *without any restriction* on the dimension of Q. In fact, this improvement is the main ingredient, which gives rise to the claimed generalisation of Kannan's algorithm.

Section 3 states the main structural result of the paper, which allows us to simplify the search for an integer point in a parameterised polyhedron. Namely, we partition the set of the right-hand sides into polynomially many sets $S_1, \ldots, S_M$ such that, for each particular $b \in S_i$, we need to try only a constant number (if n is fixed) of "candidate" solutions—each defined by means of affine transformations of b (these transformations depend on the choice of S_i)—and if none of them satisfies the system $Ax \leqslant b$, then the system has no integer solution at all. The sets S_i are not polyhedra anymore but they can be represented as *integer projections* of polyhedra—the notion we define later—that makes it possible to deal with them by means of mixed-integer programming. The proof of this decomposition almost repeats the lines of the proof in Kannan (1992), but the stronger result on width directions of a parameterised polyhedron implies a stronger result here.

In Section 4 we describe the algorithm to solve PIF. Having a partition from Section 3, it is rather easy to develop such an algorithm, since we only need to check each set S_i in the partition independently and, for each such a set, solve a polynomial number of mixed integer programs with a fixed number of integer variables. At last, we show how this algorithm can be applied to find the maximum integer programming gap for a family of integer programs.

1.1 Basic definitions and notation

For sets V and W in the Euclidean space $\mathbb{R}^n$ and a number α we denote

$$V + W = \{v + w : v \in V,\, w \in W\} \quad \text{and} \quad \alpha W = \{\alpha w : w \in W\}.$$

If W contains the origin, then αW is just scaling of W by α. If V consists of one vector v only, we write

$$v + W = \{v + w : w \in W\}$$

and say that $v + W$ is the *translate* of W along the vector v. The symbol $\lceil \alpha \rceil$ denote the smallest integer greater than or equal to α, i.e., α *rounded up*. Similarly, $\lfloor \alpha \rfloor$ stands for the largest integer not exceeding α, hence α *rounded down*.

In the paper we establish a number of *polynomial-time algorithms*, i.e., the algorithms whose running time is bounded by some polynomial in the size of the input. Following the standard agreements, we define the *size* of a rational number $\alpha = p/q$, where $p \in \mathbb{Z}$ and $q \in \mathbb{N}$ are relatively prime, as the number of bits needed to write α in binary encoding:

$$\operatorname{size}(\alpha) = 1 + \lceil \log(|p|+1) \rceil + \lceil \log(q+1) \rceil.$$

The size of a rational vector $a = (\alpha_1, \ldots, \alpha_n)$ is the sum of the sizes of its components:

$$\operatorname{size}(a) = n + \sum_{i=1}^{n} \operatorname{size}(\alpha_i).$$

At last, the size of a rational matrix $A = (\alpha_{ij})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, is

$$\operatorname{size}(a) = mn + \sum_{i=1}^{m} \sum_{j=1}^{n} \operatorname{size}(\alpha_i).$$

An *open half-space* in $\mathbb{R}^n$ is the set of the form $\{x : ax < \beta\}$, where $a \in \mathbb{R}^n$ is a row-vector and β is a number. Similarly, the set $\{x : ax \leqslant \beta\}$ is called a *closed half-space*. A *partially open polyhedron* P is the intersection of finitely many closed or open half-spaces. If P can be defined by closed half-spaces only, we say that it is a *closed polyhedron*, or simply a *polyhedron*. We need the notion of a partially open polyhedron to be able to partition the space, that is definitely impossible by means of closed polyhedra only. At last, we say that a partially open polyhedron is *rational* if it can be defined by the system of linear inequalities with rational coefficients; the right-hand sides of these inequalities may be irrational.

Linear programming is about optimising a linear function cx over a given polyhedron $P \subseteq \mathbb{R}^n$:

$$\max\{cx : x \in P\} = -\min\{-cx : x \in P\}.$$

If the variables x are required to be integers, we have an *integer programming* problem

$$\max\{cx : x \in P \cap \mathbb{Z}^n\} = -\min\{-cx : x \in P \cap \mathbb{Z}^n\}.$$

For details on linear and integer programming, we refer to Schrijver (1986). Here we only mention that a linear programming problem can be solved in polynomial time, cf. Khachiyan (1979), while integer programming is *NP*-complete. However, if the number of variables is fixed, i.e., does not belong to the input, an integer programming problem can also be solved in polynomial time, which has been shown by Lenstra (1983). Moreover, Lenstra gave an algorithm to solve *mixed integer programming* with a fixed number of integer variables. We remark that both algorithms—of Khachiyan (1979) and of Lenstra (1983)—can be used to solve *decision versions* of integer and linear programming on partially open polyhedra, too.

In Section 3 we will construct a decomposition of a polyhedron into so-called integer projections of higher-dimensional polyhedra. The *integer projection* $W/\mathbb{Z}^l$ of a set $W \subseteq \mathbb{R}^{n+l}$ is defined by

$$W/\mathbb{Z}^l = \{x \in \mathbb{R}^n : (x, w) \in W \text{ for some } w \in \mathbb{Z}^l\}.$$

In other words, it is a set of points x for which there exists an integer $w \in \mathbb{Z}^l$ such that (x, w) belongs to W. Obviously, if $V = V'/\mathbb{Z}^{l_1}$ and $W = W'/\mathbb{Z}^{l_2}$ are integer projections of some sets in $\mathbb{R}^{n+l_1}$ and $\mathbb{R}^{n+l_2}$ respectively, then $V \cap W$ is the integer projection of a set in $\mathbb{R}^{n+l_1+l_2}$.

1.2 Flatness theorem and integer programming in fixed dimension

We briefly describe the algorithm to solve integer programming in fixed dimension, as its basic ideas will be used in the following sections. Intuitively, if a polyhedron contains no integer point, then it must be "flat" along some integral direction. We make this precise by introducing the notion of "lattice width." The *width* of a closed convex set K along a direction $c \in \mathbb{R}^n$ is defined as

$$w_c(K) = \max\{cx : x \in K\} - \min\{cx : x \in K\}. \tag{1}$$

The *lattice width* of K (with respect to the *standard lattice* $\mathbb{Z}^n$) is the minimum of its widths along all nonzero integral directions:

$$w(K) = \min\{w_c(K) : c \in \mathbb{Z}^n \setminus \{0\}\}.$$

An integral row-vector c attaining the above minimum is called a *width direction* of the set K. Clearly, $w(v + \alpha K) = \alpha w(K)$ for any rational vector v and any rational number α; moreover, both sets K and $v + \alpha K$ have the same width direction.

Applications of the concept of lattice width in algorithmic number theory and integer programming rely upon the *flatness theorem*, which goes back to Khinchin (1948) who first proved it for ellipsoids in $\mathbb{R}^n$. Here we state it for *convex bodies*, i.e., bounded closed convex sets of nonzero volume.

Theorem 1 (Flatness theorem). *There exists a constant $\omega(n)$, depending only on n, such that any convex body $K \subseteq \mathbb{R}^n$ with $w(K) \geqslant \omega(n)$ contains an integer point.*

The constant $\omega(n)$ in Theorem 1 is referred to as the *flatness constant*. The best known value for the flatness constant $\omega(n)$ is $O(n^{3/2})$, due to Banaszczyk et al. (1999), although a linear dependence on n is conjectured, e.g. by Kannan and Lovász (1988).

Throughout this paper we will mostly deal with rational polyhedra rather than general convex bodies. It is easy to see that for this particular case assumptions of nonzero volume and boundedness can safely be removed from the theorem's statement. Indeed, if $P \subseteq \mathbb{R}^n$ is a rational polyhedron of zero volume, then it has width 0 along an integral direction orthogonal to its (rational) affine hull. Further, let C be the characteristic cone of P,

$$C = \{y : x + y \in P \text{ for all } x \in P\}.$$

If $C = \{0\}$, then P is already bounded. If C is full-dimensional, then the set $x + C$, for any $x \in P$, trivially contains an integer point (we can always allocate a unit box inside a full-dimensional cone). At last, if C is not full dimensional, then we can choose a sufficiently large box $B \subseteq \mathbb{R}^n$ such that $w(P) = w(P \cap B)$ and both P and $P \cap B$ have the same width direction, which is orthogonal to the (rational) affine hull of C. If $w(P) \geqslant \omega(n)$, then $P \cap B$, and hence P, contains an integer point by Theorem 1.

How can we use this theorem to check whether a given rational polyhedron contains an integer point ? The answer is in the following lemma, which is almost a direct consequence of the flatness theorem.

Lemma 2. *Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron of finite lattice width and let c be its width direction. Let $\beta = \min\{cx : x \in P\}$. Then $P \cap \mathbb{Z}^n \neq \emptyset$ if and only if the polyhedron*

$$P \cap \{x : \beta \leqslant cx \leqslant \beta + \omega(n)\}$$

contains an integer point.

Proof. If $w(P) < \omega(n)$, there is nothing to prove, as

$$P \subset \{x : \beta \leqslant cx < \beta + \omega(n)\}.$$

Suppose that $w(P) \geqslant \omega(n)$ and let $P = y + Q$, where y is an optimum solution of the linear program $\min\{cx : x \in P\}$ and Q is the polyhedron $Q = \{x - y : x \in P\}$. We denote

$$Q' = y + \frac{\omega(n)}{w(P)} Q.$$

Since $w(P) = w(Q)$, we have $w(Q') = \omega(n)$ and therefore it contains an integer point, say z. This integer point z also belongs to Q and

$$cz \leqslant cy + \omega(n),$$

that completes the proof. □

Suppose that we know a width direction c of a polyhedron $P = \{x : Ax \leqslant b\}$ in $\mathbb{R}^n$. Since c is integral, for any integer point $x \in P$ the scalar product cx must be integer. Together with Lemma 2, it allows to split the original problem into $\omega(n) + 1$ integer programming problems on lower-dimensional polyhedra

$$P \cap \{x : cx = \lceil \beta \rceil + j\}, \quad j = 0, \ldots, \omega(n)$$

where $\beta = \min\{cx : x \in P\}$.

The components of c must be relatively prime, as otherwise we could scale c obtaining a smaller width of P, and therefore its Hermite normal form is a unit row-vector $e_1 \in \mathbb{R}^n$. We can easily find a unimodular matrix U such that $cU = e_1$, introduce new variables $y = U^{-1}x$ and rewrite the original system of linear inequalities $Ax \leqslant b$ in the form $AUy \leqslant b$. Since U is unimodular, the system $Ax \leqslant b$ has an integer solution if and only if the system $AUy \leqslant b$ has an integer solution. But the equation $cx = \lceil \beta \rceil + i$ in the new coordinates has the form $y_1 = \lceil \beta \rceil + i$, where y_1 is the first component of y, thus y_1 can be eliminated. All together, we can proceed with a constant number of integer programming problems with smaller number of variables. If n is fixed, this gives us a polynomial-time algorithm.

Trying to generalise this approach for the case of varying b, gives rise to the following problems. First, the width directions of P_b depend on b and therefore can also vary. Furthermore, even if the same width direction c does not change, it is not a trivial task to to proceed recursively. The point is that β, defined to be the optimum value of the linear program $\min\{cx : x \in P_b\}$, also depends on b and therefore the hyperplanes $\{x : cx = \lceil \beta \rceil + j\}$ are not easy to construct. In the following sections we basically resolve these two problems and adapt the above algorithm for the case of varying b.

2 Lattice width of a parameterised polyhedron

A rational *parameterised polyhedron* P defined by a matrix $A \in \mathbb{Q}^{m \times n}$ is the family of polyhedra of the form

$$P_b = \{x : Ax \leqslant b\},$$

where the right-hand side vector b is allowed to vary over the whole space $\mathbb{R}^m$. We restrict our attention only to those b, for which P_b is nonempty. For each such b, there exists a width direction c of the polyhedron P_b. We aim to find a small set C of nonzero integral directions such that

$$w(P_b) = \min\{w_c(P_b) : c \in C\}$$

for all vectors b for which P_b is nonempty. Further on, the elements of the set C are referred to as *width directions* of the parameterised polyhedron P. It turns out that such a set can be computed in polynomial time when the rank of matrix A is fixed.

Let $A \in \mathbb{R}^{m\times n}$ be a matrix of full column rank. Given a subset of indices $N = \{i_1, \dots, i_n\} \subseteq \{1, \dots, m\}$, we denote by A_N the matrix composed of the rows $i_1, \dots, i_n$ of A. We say that N is a *basis* of A if A_N is invertible. Clearly, any matrix of full column rank has at least one basis. Each basis N defines a linear transformation

$$F_N : \mathbb{R}^m \to \mathbb{R}^n, \quad F_N b = A_N^{-1} b_N, \tag{2}$$

which maps right-hand sides b to the corresponding *basic solutions*. We can view F_N as an $n \times m$-matrix of rational numbers. If the point $F_N b$ satisfies the system $Ax \leqslant b$, then it is a vertex of the polyhedron $\{x : Ax \leqslant b\}$. From linear programming duality we know that the optimum value of any feasible linear program $\max\{cx : Ax \leqslant b\}$ is finite if and only if there exists a basis N such that $c = yA_N$ for some row-vector $y \geqslant 0$; in other words, c belongs to the cone generated by the rows of matrix A_N. Moreover, if it is finite, there exists a basis N such that the optimum value is attained at $F_N b$. It gives us the following simple lemma.

Lemma 3. *Let P be a parameterised polyhedron defined by a rational matrix A. If there exists a vector b' such that the polyhedron $P_{b'} = \{x : Ax \leqslant b'\}$ has infinite lattice width, then $w(P_b)$ is infinite for all b.*

Proof. Suppose that the lattice width of P_b is finite for some b and let $c \neq 0$ be the width direction. Then both linear programs in

$$\max\{cx : Ax \leqslant b\} \quad \text{and} \quad \min\{cx : Ax \leqslant b\}$$

are bounded and therefore there exist bases N_1 and N_2 of A such that c belongs to both cones $C_1 = \{yA_{N_1} : y \geqslant 0\}$ and $C_2 = \{-yA_{N_2} : y \geqslant 0\}$ generated by the rows of matrices A_{N_1} and $-A_{N_2}$ respectively. But then the linear programs

$$\max\{cx : Ax \leqslant b'\} \quad \text{and} \quad \min\{cx : Ax \leqslant b'\}$$

are also bounded, whence $w_c(P_{b'})$ is finite. □

The above Lemma shows that finite lattice width is a property of the matrix A. In particular P_0 has finite lattice width if and only if P_b has finite lattice width for all b and if P_0 has infinite lattice width, then P_b is integer feasible for all b. Since we can easily recognise whether P_0 has infinite lattice width, we shall further consider only those parameterised polyhedra, for which $w(P_0)$ is finite, and therefore $w(P_b)$ is finite for any b. We say in this case that the parameterised polyhedron P, defined by A has *finite lattice width*.

Now, suppose that P_b is nonempty and let c be a nonzero integral direction such that $w_c(P_b) = w(P_b)$. Then there exist two bases N_1 and N_2 such that

$$\max\{cx : Ax \leqslant b\} = cF_{N_1} b \quad \text{and} \quad \min\{cx : Ax \leqslant b\} = cF_{N_2} b \tag{3}$$

and c belongs to both cones $C_1 = \{yA_{N_1} : y \geqslant 0\}$ and $C_2 = \{-yA_{N_1} : y \geqslant 0\}$ generated by the rows of the matrices A_{N_1} and $-A_{N_2}$ respectively. In fact, equations (3) hold for any vector c

from $C_1 \cap C_2$. Thus, the lattice width of P_b is equal to the optimum value of the following optimisation problem:

$$\min\{c(F_{N_1} - F_{N_2})b : c \in C_1 \cap C_2 \cap \mathbb{Z}^n \setminus \{0\}\}. \tag{4}$$

The latter is an integer programming problem, since cones C_1 and C_2 can be represented by some systems of inequalities, $cD_1 \leqslant 0$ and $cD_2 \leqslant 0$ respectively, while the origin can be cut off by a single inequality, for example, $cD_1\mathbf{1} \leqslant -1$, where $\mathbf{1}$ denotes the n-dimensional all-one vector. Hence, the optimum value of (4) is attained at some vertex of the integer hull of the pointed polyhedron

$$\{c : cD_1 \leqslant 0,\, cD_2 \leqslant 0,\, cD_1\mathbf{1} \leqslant -1\}. \tag{5}$$

For fixed n, the number of these vertices is polynomial in the input size and they all can be computed in polynomial time, see Cook et al. (1992). This is summarised in the following lemma.

Lemma 4. *There exists an algorithm that takes as input a rational matrix $A \in \mathbb{Q}^{m\times n}$ of full column rank, defining a parameterised polyhedron P of finite lattice width, and computes a set of triples (F_i, G_i, c_i) of linear transformations $F_i, G_i : \mathbb{R}^m \to \mathbb{R}^n$ and a nonzero row-vector $c_i \in \mathbb{Z}^n$, $i = 1, \ldots, M$, such that for all b, for which P_b is nonempty,*

(a) F_i and G_i provide, respectively, an upper and lower bound on the value of the linear function $c_i x$ in P_b, i.e., for all i,

$$c_i G_i b \leqslant \min\{c_i x : x \in P_b\} \leqslant \max\{c_i x : x \in P_b\} \leqslant c_i F_i b,$$

(b) the lattice width of P_b is attained along the direction c_i for some $i \in \{1, \ldots, M\}$ and can be expressed as

$$w(P_b) = \min_i c_i(F_i - G_i)b.$$

The number M satisfies the bound

$$M \leqslant m^n(2n+1)^n(24n^5\phi)^{n-1}, \tag{6}$$

where ϕ is the maximum size of a column in A. The algorithm runs in polynomial time if n is fixed.

Proof. In the first step of the algorithm we enumerate all possible bases of A; since A is of full column rank, there exists at least one basis, but the total number of possible bases is at most $m^{n/2}$. The algorithm iterates over all unordered pairs of bases and for each such a pair $\{N_1, N_2\}$ does the following. Let

$$C_1 = \{yA_{N_1} : y \geqslant 0\} \quad \text{and} \quad C_2 = \{-yA_{N_2} : y \geqslant 0\}$$

be the simplicial cones generated by the rows of matrices A_{N_i} and $-A_{N_j}$ respectively. These cones can be represented by systems of linear inequalities, $cD_1 \leqslant 0$ and $cD_2 \leqslant 0$ respectively, each of which consists of n inequalities and the size of each inequality is bounded by $4n^2\phi$, see Theorem 10.2 of Schrijver (1986). As the cone $C_1 \cap C_2$ is pointed, the origin can be cut off by a single inequality; for example, $cD_1\mathbf{1} \leqslant -1$, where $\mathbf{1}$ stands for the n-dimensional all-one vector; the size of the latter inequality is bounded by $4n^3\phi$. Thus, there are $2n+1$

inequalities in (5) and the size of each is bounded by $4n^3\phi$. This implies that the number of vertices of the integer hull of (5) is at most $2\big(2n+1\big)^n\big(24n^5\phi\big)^{n-1}$, see Cook et al. (1992), and they all can be computed in polynomial time if n is fixed, by exploiting the algorithm for integer programming in fixed dimension; see Hartmann (1989) for details. The algorithm then outputs the triple (F_{N_1}, F_{N_2}, c) for each vertex c of the integer hull of (5), where F_{N_1} and F_{N_2} are the linear transformations defined by (2). Since there are at most $m^n/2$ unordered pairs of bases and, for each pair, the algorithm returns at most $2\big(2n+1\big)^n\big(24n^5\phi\big)^{n-1}$ triples, the total number of triples satisfies (6), as required. Both parts of the theorem follow directly from our previous explanation. □

The bound (6) can be rewritten as

$$M = O\big(m^n\phi^{n-1}\big)$$

for fixed n. Clearly, the greatest common divisor of the components of any direction c_i obtained by the algorithm must be equal to 1, as otherwise it would not be a vertex of (5). This implies, in particular, that the Hermite normal form of any of these vectors is just the first unit vector e_1 in $\mathbb{R}^n$.

It is also worth mentioning that if (F_i, G_i, c_i) is a triple attaining the minimum in Part (b) of Lemma 4, then we have

$$w(P_b) \leqslant \max\big\{c_i x : x \in P_b\big\} - \min\big\{c_i x : x \in P_b\big\} \leqslant c_i F_i b - c_i G_i b = w(P_b),$$

hence Part (a), when applied to this triple, turns into

$$\min\big\{c_i x : x \in P_b\big\} = c_i G_i b \quad \text{and} \quad \max\big\{c_i x : x \in P_b\big\} = c_i F_i b.$$

For our further purposes, however, it is more suitable to have a *unique* width direction for all polyhedra P_b with varying b. In fact, using Lemma 4, we can partition the set of the right-hand sides into a number of partially open polyhedra, such that the width direction remains the same for all b belonging to the same cell of the partition.

Theorem 5. *There exists an algorithm that, given a rational matrix $A \in \mathbb{Q}^{m\times n}$ of full column rank, defining a parameterised polyhedron P of finite lattice width, and a rational partially open polyhedron $Q \subseteq \mathbb{R}^m$ such that P_b is nonempty for all $b \in Q$, partitions Q into a number of partially open polyhedra $Q_1, \dots, Q_M$ and finds, for each i, a triple (F_i, G_i, c_i) of linear transformations $F_i, G_i : \mathbb{R}^m \to \mathbb{R}^n$ and a nonzero row-vector $c_i \in \mathbb{Z}^n$, such that*

$$\min\big\{c_i x : x \in P_b\big\} = c_i G_i b, \quad \max\big\{c_i x : x \in P_b\big\} = c_i F_i b,$$

and

$$w(P_b) = w_{c_i}(P_b) = c_i(F_i - G_i)b$$

for all $b \in Q_i$. If n is fixed, the algorithm runs in polynomial time and $M = O\big(m^n\phi^{n-1}\big)$, where ϕ is the maximum size of a column in A.

Remark. The statement of Theorem 5 is very analogous to Lemma 3.1 of Kannan (1992). However, there are several crucial differences. First, the number of cells in the partition obtained by Kannan's algorithm is exponential in n and the affine dimension j_0 of the polyhedron Q. Our algorithm yields a partition, which is exponential in n only, hence polynomial if n is fixed. Also our algorithm runs in polynomial time if n is fixed but j_0 may vary. At last, the algorithm of Kannan associates, to each cell Q_i of the partition, a direction c_i such that for each $b \in Q_i$, either $w_{c_i}(P_b) \leqslant 1$ or $w_{c_i}(P_b) \leqslant 2\,w(P_b)$. In contrast, we compute the *exact* width direction for each cell in the partition.

Proof of Theorem 5. First, we exploit the algorithm of Lemma 5 to obtain the triples (F_i, G_i, c_i), $i = 1, \dots, M$, with $M = O\big(m^n \phi^{n-1}\big)$, providing us the width directions of the parameterised polyhedron P. For each $i = 1, \dots, M$, we define a partially open polyhedron Q_i by the inequalities

$$\begin{aligned} c_i(F_i - G_i)b &< c_j(F_j - G_j)b, \quad j = 1, \dots, i-1, \\ c_i(F_i - G_i)b &\leqslant c_j(F_j - G_j)b, \quad j = i+1, \dots, M. \end{aligned}$$

Thus,

$$\min_j c_j(F_j - G_j)b = c_i(F_i - G_i)b$$

for all $b \in Q_i$. We claim that the intersections of the partially open polyhedra Q_i with Q give the required partition.

Indeed, let $b \in Q$ and let μ be the minimal value of $c_i(F_i - G_i)b$, $i = 1, \dots, M$. Let I denote the set of indices i with $c_i(F_i - G_i)b = \mu$. Then $b \in Q_{i_0}$, where i_0 is the smallest index in I. Yet, suppose that $b \in Q$ belongs to two partially open polyhedra, say Q_i and Q_j. Without loss of generality, we can assume $i < j$. But then we have

$$c_i(F_i - G_i)b \leqslant c_j(F_j - G_j)b < c_i(F_i - G_i)b,$$

where the first inequality is due to the fact $b \in Q_i$ and the second inequality follows from $b \in Q_j$; both together are a contradiction.

For the width directions, Lemma 4 implies that

$$w(P_b) = \min_j c_j(F_j - G_j)b = c_i(F_i - G_i)b$$

for all $b \in Q_i \cap Q$. This completes the proof. □

Now, having computed the partition of Q into partially open polyhedra $Q_1, \dots, Q_M$ and the corresponding width directions c_i, we can guarantee that, while b remains in the same polyhedron Q_i, the width direction of P_b is c_i. This solves the first problem of adapting the algorithm for integer programming in fixed dimension to the case of varying b, as stated in the introduction. However, we still need to deal with the hyperplanes $\{x : c_i x = \lceil \beta \rceil + j\}$, where β is the optimum value of the linear program $\min\{c_i x : x \in P_b\}$. As mentioned above, β can be expressed as a linear transformation of b, namely $\beta = c_i G_i b$, but $\lceil \beta \rceil$ is no more a linear function of b, that makes the recursion complicated. In the next section we describe how to tackle this problem and prove the main structural result of the paper.

3 Partitioning theorem

The following structural result, as well as its proof, is mostly taken from Kannan (1992). However, our stronger result on width directions of a parameterised polyhedron from Section 2 leads to the stronger result in this structural theorem: again, the parameter b is allowed to vary over a polyhedron of a variable dimension.

Theorem 6. *There exists an algorithm that, given a rational matrix $A \in \mathbb{Q}^{m \times n}$ of full column rank, defining a parameterised polyhedron P of finite lattice width, and a rational partially open polyhedron $Q \subseteq \mathbb{R}^m$ such that P_b is nonempty for all $b \in Q$, computes a partition of Q into sets $S_1, \dots, S_M$, each being the integer projection of a partially open polyhedron, $S_i =$*

$S'_i/\mathbb{Z}^{l_i}$, and finds, for each i, a number of unimodular transformations $U_{ij}:\mathbb{R}^n\to\mathbb{R}^n$ and affine transformations $T_{ij}:\mathbb{R}^m\to\mathbb{R}^n$, $j=1,\dots,K_i$, such that, for any $b\in S_i$, $P_b\cap\mathbb{Z}^n\neq\emptyset$ if and only if P_b contains $U_{ij}\lceil T_{ij}b\rceil$ for some index j.

If n is fixed, then the algorithm runs in polynomial time and the following bounds hold:

$$M=O\big((m^n\phi^{n-1})^{n\Omega(n)}\big),\quad l_i=O\big(\Omega(n)\big),\quad K_i=O\big(2^{n^2/2}\Omega(n)\big),\quad i=1,\dots,M,$$

where ϕ denotes the maximum size of a column in A and $\Omega(n)=\prod_{i=1}^n\omega(n)$.

Before we present the proof of this theorem, we informally discuss why it is useful. Suppose n is fixed and we want to decide whether there exists a $b\in Q$ such that $Ax\leqslant b$ is integer infeasible. The algorithm of Theorem 6 returns us a partition of Q into polynomially many sets S_i, each being the integer projection of some partially open polyhedron $S'_i\subseteq\mathbb{R}^{n+l_i}$. If n is fixed, then l_i is bounded by a constant. This means that S_i can be modeled in an extended space as the solutions of a mixed integer program with a fixed number of integer variables. Then the theorem states further that, in order to find an integer point in P_b, we need to consider K_i (a *fixed* number of) candidate solutions $U_{ij}\lceil T_{ij}b\rceil$. Notice that each of these candidate solutions for a given b can be modeled with a fixed number of integer variables too, as $T_{ij}b\in\mathbb{R}^n$ and n is fixed. We want to check whether each of these candidate solutions does not satisfy $Ax\leqslant b$. In this case, each of the candidate solutions violates at least one constraint of $Ax\leqslant b$. Since the number of candidate solutions K_i is fixed, we can check the $\binom{m}{K_i}$ many ways, in which a candidate solution is associated with a constraint to be violated. All together we can answer the question whether there exists a $b\in Q$ with $Ax\leqslant b$ being integer infeasible by solving a *polynomial number* of mixed integer programs with a fixed number of integer variables. In Section 4 we describe this again in more detail and in a slightly more general form.

Proof of Theorem 6. The proof proceeds by induction on n. First, suppose that $n=1$. The algorithm of Theorem 5 partitions Q into a polynomial amount of partially open polyhedra $Q_1,\dots,Q_M$ and computes for each $i\in\{1,\dots,M\}$ a triple (F_i,G_i,c_i) such that

$$\min\{c_ix: x\in P_b\}=c_iG_ib,\quad \max\{c_ix: x\in P_b\}=c_iF_ib,$$

holds. Notice that c_i is necessarily 1. Thus, we set $S_i=Q_i$ and assign to each S_i one transformation $T_{i1}:\mathbb{R}^m\to\mathbb{R}$ defined by $T_{i1}b=G_ib$. For $b\in S_i$ one has that P_b contains an integer point if and only if $\lceil G_ib\rceil$ is contained in P_b. The unimodular transformation U_{i1} is simply the identity.

Now, we consider a general n. Again, by applying the algorithm of Theorem 5, we obtain a partition of Q into partially open polyhedra Q_i and the corresponding triples (F_i,G_i,c_i) such that

$$w(P_b)=c_i(F_i-G_i)b$$

for all $b\in Q_i$; moreover, c_iG_ib gives the minimal value of the linear function c_ix in P_b, while c_iF_ib is its maximum value. Further on, we consider one particular cell of this partition, and (we hope it does not confuse the reader) denote it by Q. Let (F,G,c) be the corresponding triple. After applying an appropriate unimodular transformation, we may assume that c is the first unit vector e_1. This is feasible, since we can transform the candidate solutions $U_{ij}\lceil T_{ij}b\rceil$ back with the inverse of this unimodular transformation from the left.

Let P' denote the lower-dimensional parameterised polyhedron, derived from P by moving the variable x_1 to the right-hand side:

$$P'_{b-a_1x_1}=\{x': A'x'\leqslant b-a_1x_1\},$$

where A' stands for the matrix A after removing the first column a_1, x_i is the i-th component of x and $x' = [x_2, \ldots, x_n]$. The polyhedron P_b contains an integer point if and only if $P'_{b-a_1x_1}$ contains an integer point for some integer value of x_1. On the other hand, from Lemma 2 it follows that we need to consider only those values of x_1 that satisfy $e_1Gb \leqslant x_1 \leqslant e_1Gb + \omega(n)$. As b varies over Q and x_1 varies from e_1Gb to $e_1Gb + \omega(n)$, the vector $b - a_1x_1$ varies over the polyhedron

$$Q' = \{b - a_1x_1 : b \in Q,\, e_1Gb \leqslant x_1 \leqslant e_1Gb + \omega(n),\, x_1 \leqslant e_1Fb\},$$

where the last inequality ensures that we do not leave the feasible region.

As P' has a smaller dimension than P, we can use the induction hypothesis to obtain a partition of Q' into sets $R_1, \ldots, R_{M'}$, where $R_i = R'_i/\mathbb{Z}^{l_i}$, where R'_i are partially open polyhedra, and the corresponding collections of unimodular transformations U_{ij} and affine transformations T_{ij} such that $P'_{b'}$, with $b' \in R_i$, contains an integer point if and only if it contains $U_{ij}\lceil T_{ij}b'\rceil$ for some j. By induction, we also have

$$M' = O\big((m^{n-1}\phi^{n-2})^{(n-1)\Omega(n-1)}\big), \quad l_i = O\big(\Omega(n-1)\big), \quad i = 1, \ldots, M',$$

and, for each i, the number of unimodular transformations U_{ij} and affine transformations T_{ij} is $O\big(2^{(n-1)^2/2}\Omega(n-1)\big)$.

Recall that we are interested in integer points of the polyhedra $P'_{b-a_1x_1}$ for at most $\omega(n)+1$ different values of x_1, namely, $x_1 = \lceil e_1Gb\rceil + j$, $j = 0, \ldots, \omega(n)$. Consequently, we need to consider transformations U_{ij} and T_{ij} corresponding to these particular values of x_1. However, the vectors $b - \big(\lceil e_1Gb\rceil + j\big)a_1$ may happen to lie in different parts of the partition of Q'. We define our partition as follows. For every ordered tuple $I = \langle i_0, \ldots, i_{|I|-1}\rangle$ of at most $\omega(n)+1$ indices from $\{1, \ldots, M'\}$, we define S_I as the set of all $b \in Q$ such that

$$\begin{aligned} &b - \big(\lceil e_1Gb\rceil + j\big)a_1 \in R_{i_j}, && j = 0, \ldots, |I|-1 \\ &e_1Gb + j > e_1Fb, && j \geqslant |I|. \end{aligned}$$

The second constraint is equivalent to $b - \big(\lceil e_1Gb\rceil + j\big)a_1 \notin Q'$.

These sets S_I are integer projections of some higher-dimensional partially open polyhedra, $S_I = S'_I/\mathbb{Z}^{l_I}$. Indeed, $\lceil e_1Gb\rceil$ can be expressed as an integer variable z_0 satisfying the constraint

$$e_1Gb \leqslant z_0 < e_1Gb + 1,$$

while each of the conditions $b - \big(\lceil e_1Gb\rceil + j\big)a_1 \in R_{i_j}$ is equivalent to

$$(b, z_0, z) \in R'_{i_j}$$

for some integral vector $z \in \mathbb{Z}^{l_{i_j}}$. Hence we also have

$$l_I \leqslant 1 + (\omega(n)+1)O\big(\Omega(n-1)\big) = O\big(\Omega(n)\big).$$

The number of these sets is roughly

$$O\Big((m^n\phi^{n-1})\big((m^{n-1}\phi^{n-2})^{(n-1)\Omega(n-1)}\big)^{\omega(n)}\Big) = O\big((m^n\phi^{n-1})^{n\Omega(n)}\big).$$

At last, they form a partition of Q. Indeed, for each $b \in Q$, its translate $b - \big(\lceil e_1Gb\rceil + j\big)a_1$, belongs to Q' and lies in some set R_i, unless $e_1Gb + j > e_1Fb$ or $j > \omega(n)$. Consequently, there exists a tuple I such that $b \in S_I$. Similarly, b cannot lie in several sets S_I, as in this case

$b - (\lceil e_1 G b \rceil + j) a_1$, for some j, would belong to several sets R_i, which is impossible, since the R_i form a partition of Q'.

Now, we need to construct the appropriate transformations for each set S_I. Let $I = \langle i_0, \ldots, i_{N-1} \rangle$ and let S_I be the corresponding set in our partition and $b \in S_I$. If P_b contains an integer point x, then it contains one with

$$x_1 = \lceil e_1 G b \rceil + j \quad \text{for some } j = 0, \ldots, N-1.$$

For this x_1, the polyhedron $P'_{b - a_1 x_1}$ contains an integer point x', defined by

$$x' = U_{i_j k} \lceil T_{i_j k}(b - a_1 x_1) \rceil$$

for some index k. Equivalently,

$$U_{i_j k}^{-1} x' = \lceil T_{i_j k} b - T_{i_j k} a_1 x_1 \rceil.$$

To prove the induction step, we need to move x_1 to the left-hand side of the above equation. First, since x_1 is integer, the product $\lfloor T_{i_j k} a_1 \rfloor x_1$ is integer and rounding will not affect it. Hence, we get

$$\lfloor T_{i_j k} a_1 \rfloor x_1 + U_{i_j k}^{-1} x' = \lceil T_{i_j k} b - \{T_{i_j k} a_1\} x_1 \rceil,$$

where $\{T_{i_j k} a_1\}$ denotes the fractional part of the vector $T_{i_j k} a_1$. Recall that we consider

$$x_1 = \lceil e_1 G b \rceil + j. \tag{7}$$

Therefore,

$$\lfloor T_{i_j k} a_1 \rfloor x_1 + U_{i_j k}^{-1} x' = \lceil T_{i_j k} b - (e_1 G b + j)\{T_{i_j k} a_1\} - \gamma \{T_{i_j k} a_1\} \rceil \tag{8}$$

for some $0 \leqslant \gamma < 1$. Observe that $T_{i_j k} b - (e_1 G b + j)\{T_{i_j k} a_1\}$ is an affine transformation of b only; let us denote it by $T_{Ijk} b$. Furthermore, each component of the vector $\gamma \{T_{i_j k} a_1\}$ lies between 0 and 1; thus each component of the right-hand side vector in (8) can actually take only two values. Therefore, we can try all possibilities; namely, we have to consider equations

$$\lfloor T_{i_j k} a_1 \rfloor x_1 + U_{i_j k}^{-1} x' = \lceil T_{Ijk} b - v \rceil, \tag{9}$$

where $v \in \mathbb{Z}^{n-1}$ satisfies the bounds $0 \leqslant v \leqslant 1$. Obviously, there are only 2^{n-1} such vectors, which is a constant if n is fixed. The right-hand side in these equations depends now only on b. Combining them with (7), we obtain the required formula for an integer point in P_b, since the transformation

$$U_{Ijk}^{-1} = \begin{bmatrix} 1 & 0 \\ \lfloor T_{i_j k} a_1 \rfloor & U_{i_j k}^{-1} \end{bmatrix}$$

is obviously unimodular.

The above construction must be repeated for all $j = 0, \ldots, N-1$ and all indices k, thus we obtain at most

$$O\big(2^{(n-1)^2/2} \Omega(n-1)\big)\big(\omega(n) + 1\big) 2^{n-1} = O\big(2^{n^2/2} \Omega(n)\big)$$

pairs of transformations for each set S_I. As explained earlier, if there is an integer point in P_b, then there is one satisfying (7), for some j, and (8), for some k, which is equivalent to (9), for some v. This completes the proof. □

4 Applications

Now, we are ready to tackle PIF, the problem stated in the beginning of the paper:

> Given rational matrices $A \in \mathbb{Q}^{m\times n}$ and $B \in \mathbb{Q}^{k\times n}$, rational affine transformations $\Phi : \mathbb{R}^l \to \mathbb{R}^m$ and $\Psi : \mathbb{R}^l \to \mathbb{R}^k$ and a rational polyhedron $Q \subseteq \mathbb{R}^l$, find $b \in Q$ such that the system of linear inequalities $Ax \leqslant \Phi(b)$ has an integer solution, but the system $Bx \leqslant \Psi(b)$ has no integer solution, or assert that no such b exists.

The idea is now clear: First we run the algorithm of Theorem 6 on input B and $\Psi(Q)$. Then we consider each set S_i returned by the algorithm of Theorem 6 independently. For each b such that $\Psi(b) \in S_i$ we have a fixed number of candidate solutions for the system $Bx \leqslant \Psi(b)$, defined via unimodular and affine transformations as $U_{ij}\lceil T_{ij}\Psi(b)\rceil$. Each rounding operation can be expressed by introducing an integral vector: $z = \lceil T_{ij}\Psi(b)\rceil$ is equivalent to $T_{ij}\Psi(b) \leqslant z < T_{ij}\Psi(b)+\mathbf{1}$. We need only a constant number of integer variables to express all candidate solutions plus a fixed number of integer variables to represent the integer projections $S_i = S_i'/\mathbb{Z}^{l_i}$. It remains to solve a number of mixed-integer programs, to which we also include the constraints $Ax \leqslant \Phi(b)$, in order to check whether there exists $b \in Q$ such that all candidate solutions violate $Bz \leqslant \Psi(b)$, while the system $Ax \leqslant \Phi(b)$ has an integer solution.

Theorem 7. *There exists an algorithm that, given rational matrices $A \in \mathbb{Q}^{m\times n}$ and $B \in \mathbb{Q}^{k\times n}$, rational affine transformations $\Phi : \mathbb{R}^l \to \mathbb{R}^m$ and $\Psi : \mathbb{R}^l \to \mathbb{R}^k$ and a rational polyhedron $Q \subseteq \mathbb{R}^l$, finds $b \in Q$ such that the system $Ax \leqslant \Phi(b)$ has an integer solution, but the system $Bx \leqslant \Psi(b)$ has no integer solution, or assert that no such b exists. The algorithm runs in polynomial time if n is fixed.*

Proof. Let P be a parameterised polyhedron defined by the matrix B. The system $Bx \leqslant \Psi(b)$ has no integer solution if and only if $P_{\Psi(b)} \cap \mathbb{Z}^n$ is empty. First, we exploit the Fourier-Motzkin elimination procedure to construct the polyhedron $Q' \subseteq \mathbb{R}^k$ of the right-hand sides b', for which the system $Bx \leqslant b'$ has a fractional solution. For each inequality $ab' \leqslant \beta$, defining the polyhedron Q', we can solve the following mixed-integer program

$$b \in Q, \quad a\Psi(b) > \beta, \quad Ax - \Phi(b) \leqslant 0, \quad x \in \mathbb{Z}^n,$$

and if any of these problems has a feasible solution (x, b), then b is the answer to the original problem. Hence, we can terminate and output this b.

Now, we can assume that for all $b \in Q$ the system $Bx \leqslant \Psi(b)$ has a fractional solution. By applying the algorithm of Theorem 6, we construct a partition of $\Psi(Q)$ into the sets $S_1, \dots, S_M$, where each S_i is the integer projection of a partially open polyhedron, $S_i = S_i'/\mathbb{Z}^{l_i}$. Since n is fixed, the l_i are bounded by some constant, where $i = 1, \dots, M$. Furthermore, for each i, the algorithm constructs unimodular transformations U_{ij} and affine transformations T_{ij}, $j = 1, \dots, K$, such that $P_{\Psi(b)}$, with $\Psi(b) \in S_i$, contains an integer point if and only if $U_{ij}\lceil T_{ij}\Psi(b)\rceil \in P_b$ for some j. Again, K is fixed for a fixed n.

The algorithm will consider each index i independently. For a given i, S_i can be described as the set of vectors b such that

$$(\Psi(b), z) \in S_i'$$

has a solution for some integer $z \in \mathbb{Z}^{l_i}$, which can be expressed in terms of linear constraints, as S_i' is a partially open polyhedron. Let $x_j = U_{ij}\lceil T_{ij}\Psi(b)\rceil$. The points x_j can be described by linear inequalities as

$$T_{ij}\Psi(b) \leqslant z_j < T_{ij}\Psi(b) + \mathbf{1},$$

$$x_j = U_{ij} z_j,$$

where $\mathbf{1}$ is the all-one vector. Then b is a solution to our problem if and only if $x_j \notin P_{\Psi(b)}$ for all j. In this case, each x_j violates at least one constraint in the system $Bx \leqslant \Psi(b)$. We consider all possible tuples I of K constraints from $Bx \leqslant \Psi(b)$. Obviously, there are only m^K such tuples, that is, polynomially many in the input size. For each such a tuple, we solve the mixed-integer program

$$\begin{aligned}
&Ax \leqslant \Phi(b), && \\
&(\Psi(b), z) \in S'_i, && \\
&T_{ij}\Psi(b) \leqslant z_j < T_{ij}\Psi(b) + \mathbf{1}, && j = 1, \ldots, K, \\
&x_j = U_{ij} z_j, && j = 1, \ldots, K, \\
&a_{i_j} x_j > \Psi_{i_j}(b), && j = 1, \ldots, K,
\end{aligned}$$

where $a_{i_j} x \leqslant \Psi_{i_j}(b)$ is the j-th constraint in the chosen tuple. Each such a mixed-integer program can be solved in polynomial time, since the number of integer variables is fixed (in fact, there are at most $(K+1)n + l_i$ integer variables).

If there exists a feasible solution b to one of these mixed-integer programs, this b is also a solution to our original problem, hence we terminate and output b. If all these mixed-integer programs are infeasible, there is no solution to the problem. □

Integer programming gaps

Now, we describe how Theorem 7 can be applied to compute the maximum integer programming gap for a family of integer programs. Let $A \in \mathbb{Q}^{m \times n}$ be a rational matrix and let $c \in \mathbb{Q}^n$ be a rational vector. Let us consider the integer programs of the form

$$\max\{cx : Ax \leqslant b,\ x \in \mathbb{Z}^n\}, \tag{10}$$

where b is varying over $\mathbb{R}^m$. The corresponding linear programming relaxations are then

$$\max\{cx : Ax \leqslant b\}. \tag{11}$$

Consider the following system of inequalities:

$$\begin{aligned} cx &\geqslant \beta, \\ Ax &\leqslant b. \end{aligned}$$

Given a vector b and a number β, there exists a feasible fractional solution of the above system if and only if the linear program (11) is feasible and its value is at least β. The set of pairs $(\beta, b) \in \mathbb{R}^{m+1}$, for which the above system has a fractional solution, is a polyhedron in $\mathbb{R}^m$ and can be computed by means of Fourier-Motzkin elimination, in polynomial time if n if fixed. Let Q denote this polyhedron.

Suppose that we suspect the maximum integer programming gap to be smaller than γ. This means that, whenever β is an optimum value of (11), the integer program (10) must have a solution of value at least $\beta - \gamma$. Equivalently, the system

$$\begin{aligned} cx &\geqslant \beta - \gamma, \\ Ax &\leqslant b, \end{aligned} \tag{12}$$

must have an integer solution. If there exists $(b,\beta)\in Q$ such that (12) has no integer solution, the integer programming gap is bigger than γ. We also need to ensure that, for a given b, the integer program is feasible, i.e., the system $Ax \leqslant b$ has a solution in integer variables.

Now, this is exactly the question for the algorithm of Theorem 7: Is there a $(\beta,b)\in Q'$ such that the system (12) has no integer solution, but there exists $y\in\mathbb{Z}^n$ such that $Ay\leqslant b$? Here $Q' = Q-\gamma(1,0)$ is the appropriate translate of the set Q. If the algorithm returns some $(\beta-\gamma,b)$, then the integer program (10), with the right-hand side b, has no solution of value greater than $\beta-\gamma$, while being feasible. From the other hand, $(\beta,b)\in Q$, thus the corresponding linear solution has optimum value at least β. We can conclude that the maximum integer programming gap is greater than γ. This gives us the following theorem.

Theorem 8. *There exists an algorithm that, given a rational matrix $A\in\mathbb{R}^{m\times n}$, a rational row-vector $c\in\mathbb{Q}^n$ and a number γ, checks whether the maximum integer programming gap for the integer programs* (10) *defined by A and c is bigger than γ. The algorithm runs in polynomial time if the rank of A is fixed.*

Using binary search, we can also find the *minimum* possible value for γ, hence the maximum integer programming gap.

References

Banaszczyk, W., Litvak, A. E., Pajor, A., and Szarek, S. J. (1999). The flatness theorem for nonsymmetric convex bodies via the local theory of Banach spaces. *Mathematics of Operations Research*, 24(3):728–750.

Barvinok, A. (1994). A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19(4):769–779.

Barvinok, A. and Woods, K. M. (2003). Short rational generating functions for lattice point problems. *Journal of the American Mathematical Society*, 16(4):957–979.

Cook, W. J., Hartmann, M. E., Kannan, R., and McDiarmid, C. (1992). On integer points in polyhedra. *Combinatorica*, 12(1):27–37.

Cook, W. J., Lovász, L., and Schrijver, A. (1984). A polynomial-time test for total dual integrality in fixed dimension. In Korte, B. H. and Ritter, K., editors, *Mathematical programming at Oberwolfach II*, volume 22 of *Mathematical Programming Study*. North-Holland, Amsterdam.

Hartmann, M. E. (1989). *Cutting Planes and the Complexity of the Integer Hull.* PhD thesis, Department of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.

Hoşten, S. and Sturmfels, B. (2003). Computing the integer programming gap. To appear in *Combinatorica.*

Kannan, R. (1990). Test sets for integer programs, ∀∃ sentences. In Cook, W. J. and Seymour, P. D., editors, *Polyhedral Combinatorics*, volume 1 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 39–47. American Mathematical Society, Providence, RI. Proceedings of a workshop on polyhedral combinatorics, held in the Headquarters Plaza Hotel, Morristown, NJ, June 12–16, 1989.

Kannan, R. (1992). Lattice translates of a polytope and the Frobenius problem. *Combinatorica*, 12(2):161–177.

Kannan, R. and Lovász, L. (1988). Covering minima and lattice-point-free convex bodies. *The Annals of Mathematics, 2nd Series*, 128(3):577–602.

Khachiyan, L. G. (1979). A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096.

Khinchin, A. Y. (1948). A quantitative formulation of Kronecker's theory of approximation. *Izvestiya Akademii Nauk SSSR. Seriya Matematicheskaya*, 12:113–122.

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

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, Chichester. A Wiley-Interscience publication.