

Parametric Integer Programming in Fixed Dimension

FRIEDRICH EISENBRAND AND GENNADY SHMONIN

*Institut für Mathematik
Universität Paderborn
D-33095 Paderborn
Germany*

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

Abstract

We consider the following problem: Given a rational matrix $A \in \mathbb{Q}^{m \times n}$ and a rational polyhedron $Q \subseteq \mathbb{R}^{m+p}$, decide if for all vectors $b \in \mathbb{R}^m$, for which there exists an integral $z \in \mathbb{Z}^p$ such that $(b, z) \in Q$, the system of linear inequalities $Ax \leq b$ has an integral solution. We show that there exists an algorithm that solves this problem in polynomial time if p and n are fixed. This extends a result of Kannan who established such an algorithm for the case when, in addition to p and n , the affine dimension of Q is fixed.

As an application of this result, we describe an algorithm to find the maximum difference between the optimum values of an integer program $\max\{cx : Ax \leq b, x \in \mathbb{Z}^n\}$ and its linear programming relaxation over all right-hand sides b , for which the integer program is feasible. The algorithm is polynomial if n is fixed. This is an extension of a recent result of Hoşten and Sturmfels who presented such an algorithm for integer programs in standard form.

1 Introduction

Central to this paper is the following *parametric integer linear programming (PILP)* problem:

Given a rational matrix $A \in \mathbb{Q}^{m \times n}$ and a rational polyhedron $Q \subseteq \mathbb{R}^{m+p}$, decide if for all $b \in \mathbb{R}^m$, for which there exists an integral $z \in \mathbb{Z}^p$ such that $(b, z) \in Q$, the system of linear inequalities $Ax \leq b$ has an integral solution.

In other words, we need to check that for all vectors b in the set

$$Q/\mathbb{Z}^p := \{b \in \mathbb{Q}^m : (b, z) \in Q \text{ for some } z \in \mathbb{Z}^p\}$$

the corresponding *integer linear programming* problem $Ax \leq b, x \in \mathbb{Z}^n$ has a feasible solution. The set Q/\mathbb{Z}^p is called the *integer projection* of Q . Using this notation, we can reformulate PILP as the problem of testing the following $\forall \exists$ -sentence:

$$\forall b \in Q/\mathbb{Z}^p \quad \exists x \in \mathbb{Z}^n : \quad Ax \leq b. \tag{1}$$

It is worth noticing that any polyhedron $Q \subseteq \mathbb{R}^m$ as well as the set of integral vectors in Q can be expressed by means of integer projections of polyhedra. Indeed,

$$Q = Q/\mathbb{Z}^0 \quad \text{and} \quad Q \cap \mathbb{Z}^m = \{(b, b) : b \in Q\}/\mathbb{Z}^m.$$

In its general form, PILP belongs to the second level of the polynomial hierarchy and is Π_2^p -complete; see (Stockmeyer, 1976) and (Wrathall, 1976). Kannan (1990) presented a polynomial algorithm to decide the sentence (1) in the case when n , p and the affine dimension of Q are fixed. This result was applied to deduce a polynomial algorithm that solves the Frobenius problem when the number of input integers is fixed, see (Kannan, 1992).

Kannan’s algorithm proceeds in several steps. We informally describe it at this point as a way to decide $\forall \exists$ -statements (1) in the case $p = 0$. First Kannan provides an algorithm which partitions the set of right-hand sides Q into polynomially many *integer projections of partially open polyhedra* S_1, \dots, S_t , where each S_i is obtained from a higher-dimensional polyhedron by projecting out a fixed number of integer variables. Each S_i is further equipped with a fixed number of mixed integer programs such that for each $b \in S_i$ the system $Ax \leq b$ is integer feasible, if and only if one of the fixed number of “candidate solutions” obtained from plugging b in these associated mixed integer programs, is a feasible integer point.

To decide now whether (1) holds, one searches within the sets S_i individually for a vector b for which $Ax \leq b$ has *no* integral solution. In other words, each of the candidate solutions associated to b must violate at least one of the inequalities in $Ax \leq b$. Since the number of candidate solutions is fixed, we can enumerate the choices to associate a violated inequality to each candidate solution. Each of these polynomially many choices yields now a mixed-integer program with a fixed number of integer variables. There exists a $b \in S_i$ such that $Ax \leq b$ has no integral solution if and only if one of these mixed-integer programs is feasible. The latter can be checked with the algorithm of Lenstra (1983) in polynomial time.

Contributions of this paper

We modify the algorithm of Kannan to run in polynomial time under the assumption that only n and p are fixed. This is achieved via providing an algorithm that computes for a matrix $A \in \mathbb{Q}^{m \times n}$ a set $D \subseteq \mathbb{Z}^n$ of integral directions with the following property: for each $b \in \mathbb{R}^m$, the lattice width (see Section 2) of the polyhedron $P_b = \{x : Ax \leq b\}$ is equal to the width of this polyhedron along one of the directions in D . This algorithm is described in Section 3 and runs in polynomial time if n is fixed. The strengthening of Kannan’s algorithm to decide $\forall \exists$ -statements of the form (1) if n and p is fixed follows then by using this result in the proof of Theorem 4.1 in (Kannan, 1992).

We then apply this result to find the maximum *integer programming gap* for a family of integer programs. The integer programming gap of an integer program

$$\max \{ cx : Ax \leq b, x \in \mathbb{Z}^n \} \tag{2}$$

is the difference

$$\max \{ cx : Ax \leq b \} - \max \{ cx : Ax \leq b, x \in \mathbb{Z}^n \}.$$

Given a rational matrix $A \in \mathbb{Q}^{m \times n}$ and a rational objective vector $c \in \mathbb{Q}^n$, $g(A, c)$ denotes the maximum integer programming gap of integer programs of the form (2), where the maximum is taken over all vectors b , for which the integer program (2) is feasible. Our algorithm finds $g(A, c)$ in polynomial time if n is fixed. This extends a recent result of Hoşten and Sturmfels (2003), who proposed an algorithm to find the maximum integer programming gap for a family of integer programs in *standard form* if n is fixed.

Related work

Kannan's algorithm is an extension of the polynomial algorithm for integer linear programming in fixed dimension by Lenstra (1983). Barvinok and Woods (2003) presented an algorithm for counting integral points in the integer projection Q/\mathbb{Z}^p of a polytope $Q \subseteq \mathbb{R}^{m+p}$. This algorithm runs in polynomial time if p and m are fixed, and uses Kannan's partitioning algorithm, which we extend in this paper. In particular, their algorithm can be applied to count the number of elements of the minimal Hilbert basis of a pointed cone in polynomial time if the dimension is fixed. We remark that a polynomial test for the Hilbert basis property in fixed dimension was first presented by Cook et al. (1984). Extensions of Barvinok's algorithm to compute counting functions for parametric polyhedra were presented in (Barvinok and Pommersheim, 1999; Verdoolaege et al., 2007) and in (Köppe and Verdoolaege, 2007). These counting functions are piecewise step-polynomials which involve roundup operations. With these functions at hand one can very efficiently compute the number of integer points in P_b via evaluation at b . It is however not known how to use such piecewise step-polynomials to decide $\forall\exists$ -statements efficiently in fixed dimension.

Hoşten and Sturmfels (2003) proposed an algorithm to find the maximum integer programming gap for a family of integer programs in *standard form*, i.e., $\max\{cx : Ax = b, x \geq 0, x \in \mathbb{Z}^n\}$. Their algorithm exploits short rational generating functions for certain lattice point problems, cf. Barvinok (1994) and Barvinok and Woods (2003), and runs in polynomial time if the number n of columns of A is fixed. However, the latter implies also a fixed number of rows in A , 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.

Basic definitions and notation

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

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

It is easy to see that if W is a convex set containing the origin and $\alpha \leq 1$, then $\alpha W \subseteq W$. 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 $\lceil \alpha \rceil$ denotes 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 this paper we establish a number of *polynomial algorithms*, i.e., algorithms whose running time is bounded by a polynomial in the input size. Following the standard agreements, we define the *size* of a rational number $\alpha = p/q$, where $p, q \in \mathbb{Z}$ are relatively prime and $q > 0$, as the number of bits needed to write α in binary encoding:

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

The size of a rational vector $a = [a_1, \dots, a_n]$ is the sum of the sizes of its components:

$$\text{size}(a) := n + \sum_{i=1}^n \text{size}(a_i).$$

At last, the size of a rational matrix $A = [a_{ij}] \in \mathbb{Q}^{m \times n}$ is

$$\text{size}(A) := mn + \sum_{i=1}^m \sum_{j=1}^n \text{size}(a_{ij}).$$

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 $\{x : ax \leq \beta\}$ 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 means of 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 (this 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 and rational right-hand sides.

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

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

If x is required to be integral, it is an *integer linear 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 linear programming is NP-complete. However, if the number of variables is fixed, integer programming can also be solved in polynomial time, as was shown by Lenstra (1983). Moreover, Lenstra presented 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.

An integral square matrix U is called *unimodular* if $|\det(U)| = 1$. Clearly, if U is unimodular, then U^{-1} is also unimodular. A matrix of full row rank is said to be in *Hermite normal form* if it has the form $[H \ 0]$, where $H = [h_{ij}]$ is a square non-singular non-negative upper-triangular matrix such that $h_{ii} > h_{ij}$ for all $j > i$. Given a matrix A of full row rank, we can find in polynomial time a unimodular matrix U such that AU is in Hermite normal form; see (Kannan and Bachem, 1979). We remark that the Hermite normal form of an integral vector c is the vector αe_1 , where α is the greatest common divisor of the components of c and e_1 is the first unit vector. The unimodular matrix U such that $cU = \alpha e_1$ can be obtained directly while executing the Euclidean algorithm to compute the greatest common divisor.

2 Flatness theorem and integer programming in fixed dimension

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

$$w_c(K) := \max\{cx : x \in K\} - \min\{cx : x \in K\}. \quad (3)$$

The *lattice width* $w(K)$ of K (with respect to the *standard lattice* \mathbb{Z}^n) is the minimum of its widths along all non-zero 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 non-negative 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 non-zero volume.

Theorem 2.1 (Flatness theorem). *There is a constant $\omega(n)$, depending only on n , such that any convex body $K \subseteq \mathbb{R}^n$ with $w(K) \geq \omega(n)$ contains an integral point.*

The constant $\omega(n)$ in Theorem 2.1 is referred to as the *flatness constant*. The best known *upper bound* on $\omega(n)$ is $O(n^{3/2})$, cf. (Banaszczyk et al., 1999), although a linear dependence on n was conjectured, e.g., by Kannan and Lovász (1988). A linear *lower bound* on $\omega(n)$ was shown by Kantor (1999) and Sebő (1999).

Throughout this paper we will mostly deal with rational polyhedra rather than general convex bodies. In this case, assumptions of non-zero 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 $x + C$ trivially contains an integral point, for any $x \in P$ (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 $w(P) \geq \omega(n)$, then $P \cap B$, and hence P , contains an integral point by Theorem 2.1.

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

Lemma 2.2. *Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron of finite lattice width and let c be its width direction. Let*

$$\beta := \min \{ cx : x \in P \}. \tag{4}$$

Then P contains an integral point if and only if the polyhedron

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

contains an integral point.

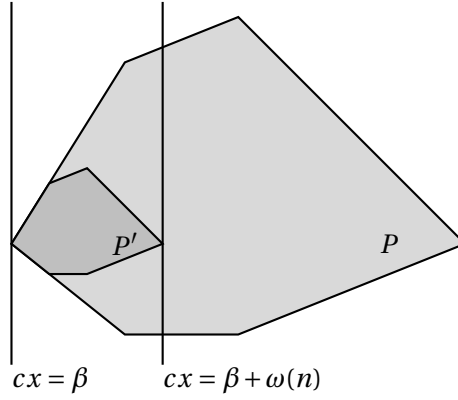


Figure 1: Illustration for the proof of Lemma 2.2

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

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

Suppose that $w(P) \geq \omega(n)$ and let $P = y + Q$, where y is an optimum solution of the linear program (4) and Q is the polyhedron containing the origin,

$$Q := \{x - y : x \in P\}.$$

We denote

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

In other words, Q is P translated to contain the origin, Q' is obtained from Q by scaling it down, and P' is Q' translated back to the original position. It is easy to see that

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

Since $\frac{\omega(n)}{w(P)} \leq 1$ and Q is convex, we have $Q' \subseteq Q$. This implies $P' \subseteq P$. Yet, we have $w(P) = w(Q)$, and therefore, $w(P') = w(Q') = \omega(n)$.

By Theorem 2.1, P' contains an integral point, say z . But then z also belongs to P and

$$cz \leq \max \{cx : x \in P'\} = \beta + \omega(n).$$

This completes the proof. □

Suppose that we know a width direction c of a polyhedron

$$P = \{x : Ax \leq b\} \subseteq \mathbb{R}^n. \quad (5)$$

Since c is integral, the scalar product cx must be an integer for any integral point $x \in P$. Together with Lemma 2.2, it allows us to split the original problem into $\omega(n) + 1$ integer programming problems on lower-dimensional polyhedra

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

where β is defined by (4).

The components of c must be relatively prime, as otherwise we could scale c , obtaining a smaller width of P . Therefore its Hermite normal form is a unit row-vector e_1 . We can 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 $Ax \leq b$ in the form $AUy \leq b$. Since U is unimodular, the system $Ax \leq b$ has an integral solution if and only if the system $AUy \leq b$ has an integral solution. But the equation $cx = \lceil \beta \rceil + j$ turns into $e_1y = \lceil \beta \rceil + j$. Thus, the first component of y can be eliminated. All together, we can proceed with a constant number of integer programming problems with a smaller number of variables. If n is fixed, this yields a polynomial algorithm.

An attempt to generalize this approach for the case of varying b gives rise to the following problems. First, the width directions of the polyhedron (5) depend on b and therefore can also vary. Furthermore, even if a width direction c remains the same, it is not a trivial task to proceed recursively. The point is that β , as it is defined in (4), also depends on b and the hyper-planes $\{x : cx = \lceil \beta \rceil + j\}$ are not easy to construct with β being a *function* of b . In the following sections we basically resolve these two problems and adapt the above algorithm for the case of varying b .

3 Lattice width of a parametric polyhedron

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

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

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

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

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

Let $A \in \mathbb{Q}^{m \times n}$ 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 non-singular. Clearly, any matrix of full column rank has at least one basis. Each basis N defines a linear transformation

$$F_N : \mathbb{R}^m \rightarrow \mathbb{R}^n, \quad F_N b = A_N^{-1} b_N, \quad (6)$$

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 $Ax \leq b$, then it is a vertex of the polyhedron $\{x : Ax \leq b\}$. From linear programming duality we know that the optimum value of any feasible linear program

$$\max \{cx : Ax \leq b\}$$

is finite if and only if there is a basis N such that $c = yA_N$ for some row-vector $y \geq 0$. In other words, c must belong to the cone generated by the rows of matrix A_N . Moreover, if it is finite, there is a basis N such that the optimum value is attained at $F_N b$. It gives us the following simple lemma.

Lemma 3.1. *Let P be a parametric polyhedron defined by a rational matrix A . If there exists a vector b' such that the polyhedron*

$$P_{b'} = \{x : Ax \leq b'\}$$

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 be a width direction. Then both linear programs

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

are bounded and therefore there are bases N_1 and N_2 of A such that c belongs to both cones

$$C_1 := \{yA_{N_1} : y \geq 0\} \quad \text{and} \quad C_2 := \{-yA_{N_2} : y \geq 0\} \quad (7)$$

generated by the rows of matrices A_{N_1} and $-A_{N_2}$, respectively. But then the linear programs

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

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

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 . Conversely, if P_0 has infinite lattice width, then P_b also has infinite lattice width and therefore contains an integral point for all b . We can easily recognize whether P_0 has infinite lattice width. For instance, we can enumerate all possible pairs of bases N_1 and N_2 and check if the cones (7) have a common integral vector. Further we shall not deal with this “trivial” case and shall consider only those parametric 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 parametric polyhedron P has *finite lattice width*.

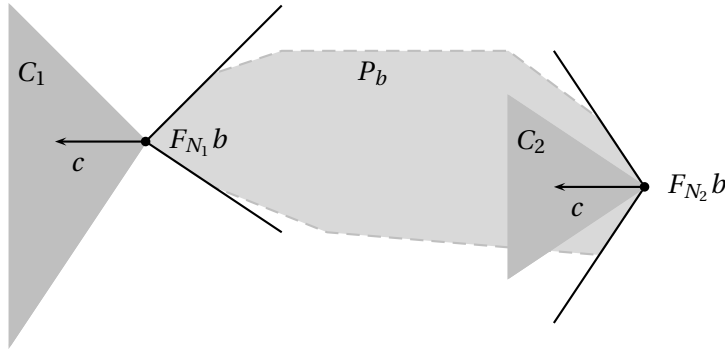


Figure 2: The width direction c and the two cones C_1 and C_2 .

Now, suppose that P_b is non-empty and let c be its width direction. Then there are two bases N_1 and N_2 such that

$$\max\{cx : Ax \leq b\} = cF_{N_1}b \quad \text{and} \quad \min\{cx : Ax \leq b\} = cF_{N_2}b \quad (8)$$

and c belongs to the cones C_1 and C_2 defined by (7), see Figure 2. In fact, equations (8) hold for any vector c in $C_1 \cap C_2$. Thus, the lattice width of P_b is equal to the optimum value of the following optimization problem:

$$\min\{c(F_{N_1} - F_{N_2})b : c \in C_1 \cap C_2 \cap \mathbb{Z}^n \setminus \{0\}\}. \quad (9)$$

The latter can be viewed as an integer programming problem. Indeed, the cones C_1 and C_2 can be represented by some systems of linear inequalities, say $cD_1 \leq 0$ and $cD_2 \leq 0$, respectively, where $D_1, D_2 \in \mathbb{Z}^{n \times n}$. The minimum (9) is taken over all integral vectors c satisfying $cD_1 \leq 0$ and $cD_2 \leq 0$, except the origin. Since both cones C_1 and C_2 are simplicial, i.e., generated by n linearly independent vectors, the origin is a vertex of $C_1 \cap C_2$ and therefore can be cut off by a single inequality, for example, $cD_1 \mathbf{1} \leq -1$, where $\mathbf{1}$ denotes the n -dimensional all-one vector. It is important that all other integral vectors c in $C_1 \cap C_2$ satisfy this inequality and therefore remain feasible. Thus, the problem (9) can be rewritten as

$$\min \{ c(F_{N_1} - F_{N_2})b : cD_1 \leq 0, cD_2 \leq 0, cD_1 \mathbf{1} \leq -1, c \in \mathbb{Z}^n \}.$$

For a given b , this is an integer programming problem. Therefore, the optimum value of (9) is attained at some vertex of the integer hull of the underlying polyhedron

$$\{ c : cD_1 \leq 0, cD_2 \leq 0, cD_1 \mathbf{1} \leq -1 \} \quad (10)$$

Shevchenko (1981) and Hayes and Larman (1983) proved that the number of vertices of the integer hull of a rational polyhedron is polynomial in fixed dimension. Tight bounds for number were presented in (Cook et al., 1992) and (Bárány et al., 1992). This gives rise to the next lemma.

Lemma 3.2. *There is an algorithm that takes as input a rational matrix $A \in \mathbb{Q}^{m \times n}$ of full column rank, which defines a parametric polyhedron P of finite lattice width, and computes a set of triples (F_i, G_i, c_i) of rational linear transformations $F_i, G_i : \mathbb{R}^m \rightarrow \mathbb{R}^n$ and a non-zero integral row-vector $c_i \in \mathbb{Z}^n$ ($i = 1, \dots, t$) satisfying the following properties. For all b , for which P_b is non-empty,*

(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 ,

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

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

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

(c) The number t of the triples satisfies the bound

$$t \leq 2m^{2n} (2n+1)^n (24n^5 \phi)^{n-1}, \quad (11)$$

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 . Observe that there is at least one basis, since A is of full column rank. On the other hand, the total number of possible bases is at most m^n . Hence, the number of possible pairs of bases is bounded by m^{2n} . The algorithm iterates over all unordered pairs of bases and for each such pair $\{N_1, N_2\}$ does the following.

Let C_1 and C_2 be the corresponding simplicial cones, defined by (7). These cones can be represented by systems of linear inequalities, $cD_1 \leq 0$ and $cD_2 \leq 0$ respectively, where $D_1, D_2 \in \mathbb{Z}^{n \times n}$ and the size of each inequality is bounded by $4n^2 \phi$, see (Schrijver, 1986, Theorem 10.2). As the origin is

a vertex of the cone $C_1 \cap C_2$, it can be cut off by a single inequality; for example, $cD_1\mathbf{1} \leq -1$, 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 exactly $2n + 1$ inequalities in (10) and the size of each inequality is bounded by $4n^3\phi$. This implies that the number of vertices of the integer hull of (10) is at most $2(2n + 1)^n(24n^5\phi)^{n-1}$, cf. (Cook et al., 1992), and they all can be computed in polynomial time if n is fixed, cf. (Hartmann, 1989). The algorithm then outputs the triple (F_{N_1}, F_{N_2}, c) for each vertex c of the integer hull of (10), where F_{N_1} and F_{N_2} are the linear transformations defined by (6). Since there are at most m^{2n} unordered pairs of bases and, for each pair, the algorithm returns at most $2(2n + 1)^n(24n^5\phi)^{n-1}$ triples, the total number of triples satisfies (11), as required. Parts (a) and (b) of the theorem follow directly from our previous explanation. \square

The bound (11) can be rewritten as

$$t = O(m^{2n}\phi^{n-1})$$

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 the integer hull of (10). 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 3.2, then we have

$$w(P_b) \leq \max\{c_i x : x \in P_b\} - \min\{c_i x : x \in P_b\} \leq c_i F_i b - c_i G_i b = w(P_b).$$

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

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

For our further purposes, it is more suitable to have a *unique* width direction for all polyhedra P_b with varying b . In fact, using Lemma 3.2, we can partition the set of the right-hand sides b into a number of partially open polyhedra such that the width direction remains the same for all b belonging to the same region of the partition.

Theorem 3.3. *Let P be a parametric polyhedron of finite lattice width, defined by a matrix $A \in \mathbb{Q}^{m \times n}$ of full column rank. Let $Q \subseteq \mathbb{R}^m$ be a rational partially open polyhedron such that P_b is non-empty for all $b \in Q$. We can compute—in polynomial time, if n is fixed—a partition of Q into a number of partially open polyhedra Q_1, \dots, Q_t and, for each i , find a triple (F_i, G_i, c_i) of linear transformations $F_i, G_i : \mathbb{R}^m \rightarrow \mathbb{R}^n$ and a non-zero vector $c_i \in \mathbb{Z}^n$, such that*

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

and

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

for all $b \in Q_i$. If ϕ denotes the maximum size of a column in A , then $t = O(m^{2n}\phi^{n-1})$.

Remark. The statement of Theorem 3.3 is very analogous to (Kannan, 1992, Lemma 3.1). However, there are several crucial differences. First, the number of regions 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 that 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 width directions c_i obtained by the Kannan's algorithm satisfy, for any $b \in Q_i$,

$$\text{either } w_{c_i}(P_b) \leq 1 \quad \text{or} \quad w_{c_i}(P_b) \leq 2w(P_b)$$

In contrast, we compute the *exact* width direction for each region in the partition. While the exact computation of width directions does not help much from the algorithmic point of view, removing the restriction on the dimension of Q turns out to be the main step towards the claimed generalization of Kannan's algorithm.

Proof of Theorem 3.3. First, we exploit the algorithm of Lemma 3.2 to obtain the triples (F_i, G_i, c_i) , $i = 1, \dots, t$, with $t = O(m^n \phi^{n-1})$. These triples provide the possible width directions of the parametric polyhedron P . For each $i = 1, \dots, t$, we define a partially open polyhedron Q_i by the inequalities

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

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 minimum value of $c_i(F_i - G_i)b$, $i = 1, \dots, t$. 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

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

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 3.2 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. □

Theorem 3.3 provides a unique width direction for each region Q_i of the partition. This resolves the first problem of adapting the algorithm for integer linear programming in fixed dimension to the case of varying b , which was addressed in the introduction. However, we still need to deal with the hyper-planes $\{x : c_i x = \lceil \beta \rceil + j\}$, 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 , which makes the recursion complicated. Kannan (1992) showed how to tackle this problem. We discuss it in the next section.

4 Partitioning theorem and parametric integer programming

The proof of the following structural result follows from the proof of (Kannan, 1992, Theorem 4.1) if it is combined with Theorem 3.3.

Theorem 4.1. *Let P be a parametric polyhedron of finite lattice width, defined by a rational matrix $A \in \mathbb{Q}^{m \times n}$ of full column rank. Let $Q \subseteq \mathbb{R}^m$ be a rational partially open polyhedron such that P_b is non-empty for all $b \in Q$. We can compute—in polynomial time, if n is fixed—a partition of Q into sets S_1, \dots, S_t , and for each i , find a number of unimodular transformations $U_{ij} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and affine transformations $T_{ij} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, $j = 1, \dots, k_i$ such that*

- (a) *each S_i is the integer projection of a partially open polyhedron, $S_i = S'_i / \mathbb{Z}^{l_i}$;*
- (b) *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 ;*
- (c) *the following bounds hold:*

$$t = O((m^{2n} \phi^{n-1})^{n\bar{\omega}(n)}), \quad l_i = O(\bar{\omega}(n)), \quad k_i = O(2^{n^2/2} \bar{\omega}(n)), \quad i = 1, \dots, t,$$

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

We do not repeat Kannan's proof here but give an intuition of why it is true in dimension 2. By Theorem 3.3 we can assume that the width-direction is invariant for all $b \in Q$ and by applying a unimodular transformation we can further assume that this width-direction is the first unit-vector e_1 , see Figure 3.

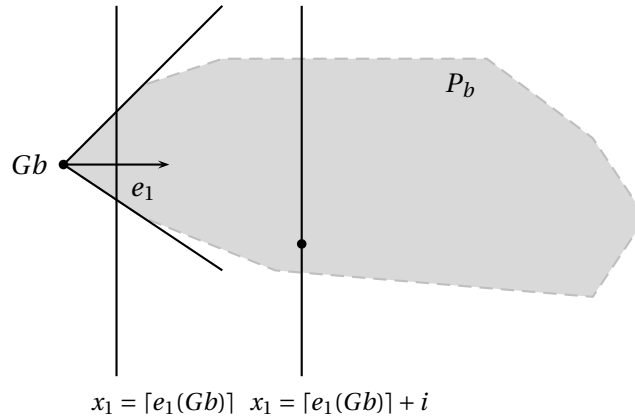


Figure 3: Illustration of Theorem 4.1 in dimension 2.

Lemma 2.2 tells that P_b contains an integral point if and only if there exists an integral point on the lines $x_1 = \lceil e_1(Gb) \rceil + j$ for $j = 0, \dots, \omega(2)$. The intersections of these lines with the polyhedron P_b are 1-dimensional polyhedra. Some of the constraints $ax \leq \beta$ of $Ax \leq b$ are “pointing upwards”, i.e., $ae_2 < 0$, where e_2 is the second unit-vector. Let $a_1x_1 + a_2x_2 \leq \beta$ be a constraint pointing upwards such that the intersection point $(\lceil e_1(Gb) \rceil + j, y)$ of $a_1x_1 + a_2x_2 = \beta$ with the line $x_1 = \lceil e_1(Gb) \rceil + j$ has the largest second component. The line $x_1 = \lceil e_1(Gb) \rceil + j$ contains an integral point in P_b if and only if $(\lceil e_1(Gb) \rceil + j, \lceil y \rceil)$ is contained in P_b . This point is illustrated in Figure 3. By choosing the

highest constraint pointing upwards for each line $x_1 = \lceil e_1(Gb) \rceil + j$, we partition the set of right-hand sides into polynomially many integer projections of partially open polyhedra.

In order to express the candidate solution $(\lceil e_1(Gb) \rceil + j, \lceil y \rceil)$ in the form described in the theorem, observe that

$$y = (\beta - a_1 x_1) / a_2.$$

Since x_1 is an integer and

$$x_1 = \lceil e_1(Gb) \rceil + j = e_1(Gb) + j + \gamma$$

for some $\gamma \in [0, 1)$, we can rewrite the equation $\lceil y \rceil = \lceil (\beta - a_1 x_1) / a_2 \rceil$ as

$$\lfloor a_1 / a_2 \rfloor x_1 + \lceil y \rceil = \lceil \beta / a_2 - \{a_1 / a_2\} x_1 \rceil = \lceil \beta / a_2 - \{a_1 / a_2\} (e_1(Gb) + j) - \{a_1 / a_2\} \gamma \rceil,$$

where $\{a_1 / a_2\}$ denotes the fractional part of a_1 / a_2 . Since $\{a_1 / a_2\} \gamma$ lies between 0 and 1, it suffices to check independently two different possibilities, namely,

$$\lfloor a_1 / a_2 \rfloor x_1 + \lceil y \rceil = \lceil \beta / a_2 - \{a_1 / a_2\} (e_1(Gb) + j) \rceil$$

and

$$\lfloor a_1 / a_2 \rfloor x_1 + \lceil y \rceil = \lceil \beta / a_2 - \{a_1 / a_2\} (e_1(Gb) + j) - 1 \rceil.$$

Combined with $x_1 = \lceil e_1(Gb) \rceil + j$, each of the above equations yields a unimodular system with respect to the variables x_1 and $\lceil y \rceil$, with the right-hand side being the round-up of an affine transformation of b . We refer the reader to (Kannan, 1992) to see the complete proof for arbitrary dimension.

$\forall \exists$ -statements

Theorem 4.1 gives rise to a polynomial algorithm for testing sentences of the form

$$\forall b \in Q / \mathbb{Z}^p \quad \exists x \in \mathbb{Z}^n : Ax \leq b, \tag{12}$$

when p and n are fixed. This algorithm was first described by Kannan (1992) but he required, in addition, the affine dimension of Q to be fixed. Our improvement follows basically from the improvement in the partitioning theorem, while the algorithm itself remains exactly the same. We describe it here for the sake of completeness. First observe that we can assume that A has full column rank. Otherwise we can apply a unimodular transformation of A from the right to obtain a matrix $[A' \mid 0]$, where A' has full column rank.

The idea is as follows: First we run the algorithm of Theorem 4.1 on input A and $Q' \subseteq \mathbb{R}^m$, where Q' is the set of vectors b , for which the system $Ax \leq b$ has a solution. Then we consider each set S_i returned by the algorithm of Theorem 4.1 independently. For each $b \in S_i$ we have a fixed number of candidate solutions for the system $Ax \leq b$, defined via unimodular and affine transformations as $U_{ij} \lceil T_{ij} b \rceil$. Each rounding operation can be expressed by introducing an integral vector: $z = \lceil T_{ij} b \rceil$ is equivalent to $T_{ij} b \leq z < T_{ij} b + \mathbf{1}$. 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 $S_i = S'_i / \mathbb{Z}^{h_i}$. It remains to solve a number of mixed-integer programs, to which we also include the constraints $(b, y) \in Q, y \in \mathbb{Z}^p$.

Theorem 4.2. *There is an algorithm that, given a rational matrix $A \in \mathbb{Q}^{m \times n}$ and a rational polyhedron $Q \subseteq \mathbb{R}^{m+p}$, decides the sentence (12). The algorithm runs in polynomial time if p and n are fixed.*

Proof. Let P be a parametric polyhedron defined by the matrix A . First, we exploit the Fourier–Motzkin elimination procedure to construct the polyhedron $Q' \subseteq \mathbb{R}^m$ of the right-hand sides b , for which the system $Ax \leq b$ has a (fractional) solution. For each inequality $ab \leq \beta$, defining the polyhedron Q' , we can solve the following mixed-integer program

$$ab > \beta, \quad (b, y) \in Q, \quad y \in \mathbb{Z}^p,$$

and if any of these problems has a feasible solution (y, b) , then b is a vector in Q/\mathbb{Z}^p , for which the system $Ax \leq b$ has no integral solution. Hence, we can terminate and output “no” (with b being a certificate).

We can assume now that for all $b \in Q/\mathbb{Z}^p$ the system $Ax \leq b$ has a fractional solution. By applying the algorithm of Theorem 4.1, we construct a partition of Q' into the sets S_1, \dots, S_t , 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, $i = 1, \dots, t$. Furthermore, for each i , the algorithm constructs unimodular transformations U_{ij} and affine transformations T_{ij} , $j = 1, \dots, k_i$, such that P_b , with $b \in S_i$, contains an integral point if and only if $U_{ij}[T_{ij}b] \in P_b$ for some j . Again, k_i is fixed for a fixed n , $i = 1, \dots, t$.

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

$$(b, z) \in S'_i$$

has a solution for some integer $z \in \mathbb{Z}^{l_i}$. This can be expressed in terms of linear constraints, as S'_i is a partially open polyhedron. Let $x_j = U_{ij}[T_{ij}b]$. The points x_j can be described by linear inequalities as

$$T_{ij}b \leq z_j < T_{ij}b + \mathbf{1}, \quad x_j = U_{ij}z_j,$$

where $\mathbf{1}$ is the all-one vector. Then P_b does not contain an integral point if and only if $x_j \notin P_b$ for all $j = 1, \dots, k_i$. In this case, each x_j violates at least one constraint in the system $Ax \leq b$. We consider all possible tuples I of k_i constraints from $Ax \leq b$. Obviously, there are only m^{k_i} such tuples, that is, polynomially many in the input size. For each such tuple, we solve the mixed-integer program

$$\begin{aligned} (b, y) \in Q, (b, z) \in S'_i, \\ T_{ij}b \leq z_j < T_{ij}b + \mathbf{1}, \quad j = 1, \dots, k_i, \\ x_j = U_{ij}z_j, \quad j = 1, \dots, k_i, \\ a_{i,j}x_j > b_{i,j}, \quad j = 1, \dots, k_i, \\ y \in \mathbb{Z}^p, z \in \mathbb{Z}^{l_i}, z_j \in \mathbb{Z}^n, \quad j = 1, \dots, k_i, \end{aligned}$$

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

If there is a feasible solution b to one of these mixed-integer programs, then the answer to the original problem is “no” (with b being a certificate). If all these mixed-integer programs are infeasible, the answer is “yes”. \square

Remark. We would like to point out that Theorem 4.2 can also be proved differently. Bell (1977) and Scarf (1977) showed that if a system of linear inequalities $Ax \leq b$ has no integral solution, then

there is already a subsystem of at most 2^n inequalities that is infeasible in integer variables; see also (Schrijver, 1986, Theorem 16.5). Applied to (1) this means that there exists a $b \in Q/\mathbb{Z}^p$ such that the system $Ax \leq b$ has no integral solution, if and only there exist a $b \in Q/\mathbb{Z}^p$ and a subsystem $A'x \leq b'$ of $Ax \leq b$ with at most 2^n inequalities, which is infeasible in integer variables. Here b' is the projection of b into the according space. Since n is a constant, we can try out all $\binom{m}{2^n}$ different subsystems of $Ax \leq b$ and apply to each of these parametric polyhedra Kannan's algorithm to decide $\forall \exists$ -statements.

However, a similar argument does not yield our extension (Theorem 4.1) of Kannan's partitioning theorem itself, which associates to each b a fixed set of candidate integer solutions, depending on the partially open polyhedron of the partitioning, in which b is contained.

5 Integer programming gaps

Now, we describe how Theorem 4.2 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

$$\max \{ cx : Ax \leq b, x \in \mathbb{Z}^n \}, \quad (13)$$

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

$$\max \{ cx : Ax \leq b \}. \quad (14)$$

Consider the following system of inequalities:

$$\begin{aligned} cx &\geq \beta, \\ Ax &\leq b. \end{aligned}$$

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 (14) 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 is 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 (14), the integer program (13) must have a solution of value at least $\beta - \gamma$. Equivalently, the system

$$\begin{aligned} cx &\geq \beta - \gamma, \\ Ax &\leq b, \end{aligned} \quad (15)$$

must have an integral solution. If there exists $(b, \beta) \in Q$ such that (15) has no integral 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 $Ax \leq b$ has a solution in integer variables.

Now, this is exactly the question for the algorithm of Theorem 4.2: Is there a $(\beta, b) \in Q'$ such that the system (15) has no integral solution, but there exists $y \in \mathbb{Z}^n$ such that $Ay \leq b$? Here $Q' = Q - \gamma(1, 0)$ is the appropriate translate of the set Q . If the algorithm answers “no” with the certificate b , then the integer program (13), with the right-hand side b , has no solution of value greater than

$\beta - \gamma$ while being feasible. On the other hand, $(\beta, b) \in Q$, thus the corresponding linear solution has optimum value of at least β . We can conclude that the maximum integer programming gap is greater than γ . This gives us the following theorem.

Theorem 5.1. *There is 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 (13) 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 of γ , hence the maximum integer programming gap.

Acknowledgments

We thank the anonymous referees for their careful reading and their very valuable suggestions, which helped us to improve the presentation of the results in this paper. Thanks also to Matthias Köppe for many discussions on generating functions.

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.
- Bárány, I., Howe, R., and Lovász, L. (1992). On integer points in polyhedra: A lower bound. *Combinatorica*, 12(2):135–142.
- Barvinok, A. I. (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 Pommersheim, J. E. An algorithmic theory of lattice points in polyhedra. In *New perspectives in algebraic combinatorics (Berkeley, CA, 1996–97)*, pages 91–147. Cambridge Univ. Press, Cambridge, 1999.
- Barvinok, A. I. and Woods, K. M. (2003). Short rational generating functions for lattice point problems. *Journal of the American Mathematical Society*, 16(4):957–979.
- Bell, D. E. (1977). A theorem concerning the integer lattice. *Studies in Appl. Math.*, 56(2):187–188.
- 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.
- Hayes, A. C. and Larman, D. G. (1983). The vertices of the knapsack polytope. *Discrete Applied Mathematics*, 6(2):135–138.

- 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, $\forall\exists$ 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, Providence, RI. American Mathematical Society.
- Kannan, R. (1992). Lattice translates of a polytope and the Frobenius problem. *Combinatorica*, 12(2):161–177.
- Kannan, R. and Bachem, A. (1979). Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507.
- Kannan, R. and Lovász, L. (1988). Covering minima and lattice-point-free convex bodies. *Annals of Mathematics*, 128:577–602.
- Kantor, J.-M. (1999). On the width of lattice-free simplices. *Compositio Mathematica*, 118(3):235–241.
- 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.
- Köppe, M. and Verdoolaege, S. Computing parametric rational generating functions with a primal barvinok algorithm. arXiv.org, arXiv:0705.3651v2, 2007.
- Lenstra, Jr., H. W. (1983). Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548.
- Scarf, H. E. (1977). An observation on the structure of production sets with indivisibilities. *Proc. Nat. Acad. Sci. U.S.A.*, 74(9):3637–3641.
- Shevchenko, V. N. (1981). O chisle kraynih toчек v tselochislennom programmirovanii [Russian; On the number of extreme points in integer programming]. *Kibernetika*, (2):133–134.
- Schrijver, A. (1986). *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Chichester, SXW.
- Sebő, A. (1999). An introduction to empty lattice simplices. In Cornuéjols, G., Burkard, R. E., and Woeginger, G. J., editors, *Integer Programming and Combinatorial Optimization, 7th International IPCO Conference, Graz, Austria, June 9–11, 1999, Proceedings*, volume 1610 of *Lecture Notes in Computer Science*, pages 400–414. Springer.
- Stockmeyer, L. J. (1976). The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22.

Verdoolaege, S., Seghir, R., Beyls, K., Loechner, V. and Bruynooghe, M. Counting integer points in parametric polytopes using Barvinok's rational functions. *Algorithmica*, 48(1):37–66, 2007. ISSN 0178-4617.

Wrathall, C. (1976). Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33.