Integer programming, lattices, and results in fixed dimension

Karen Aardal Friedrich Eisenbrand

June 2, 2004

Abstract

We review and describe several results regarding integer programming problems in fixed dimension. First, we describe various lattice basis reduction algorithms that are used as auxiliary algorithms when solving integer feasibility and optimization problems. Next, we review three algorithms for solving the integer feasibility problem. These algorithms are based on the idea of branching on lattice hyperplanes, and their running time is polynomial in fixed dimension. We also briefly describe an algorithm, based on a different principle, to count integer points in an integer polytope. We then turn the attention to integer optimization. Again, we describe three algorithms: binary search, a linear algorithm for a fixed number of constraints, and a randomized algorithm for a varying number of constraints. The topic of the next part of our chapter is how to use lattice basis reduction in problem reformulation. Finally, we review cutting plane results when the dimension is fixed.

1 Introduction

Integer programming problems have offered, and are still offering, many challenging theoretical and computational questions. We consider two integer programming problems. Given is a set of rational linear inequalities $\boldsymbol{Ax} \leq \boldsymbol{d}$. The first problem is the *integer feasibility problem*: Does there exist an integer vector $\boldsymbol{x}$ satisfying $\boldsymbol{Ax} \leq \boldsymbol{d}$? The second problem is the *integer optimization problem*: Determine an integer vector $\boldsymbol{x}$ that satisfies $\boldsymbol{Ax} \leq \boldsymbol{d}$, and also maximizes or minimizes a given linear function $\boldsymbol{c}^T\boldsymbol{x}$.

The feasibility problem was proved to be NP-complete in 1976, but an interesting complexity question remained: Is the feasibility problem solvable in polynomial time if the the number of variables, i.e., the number of components of $\boldsymbol{x}$, is fixed? The predominantly used algorithm, branch-and-bound, is not a polynomial time algorithm in fixed dimension, but in 1983 H.W. Lenstra, Jr. developed an algorithm with a polynomial running time if the dimension is fixed. His algorithm is based on results from number theory; in particular on properties of lattices and lattice bases. Since then we have seen several results built on knowledge about lattices, and also many other results for integer programming problems in fixed dimension.

In our chapter we will illustrate some of these results. Since lattices and lattice bases play an important role we will present three algorithms for finding “good” lattice bases in Section 3. In this section we also review algorithms to compute a shortest vector of a lattice. In Section 4 we focus on the integer feasibility problem and describe three algorithms built on the fundamental result that if a polytope does not contain an integer vector, then there exists a nonzero integer direction in which the polytope is intersected by at most $f(n)$ so-called lattice hyperplanes, where $f(n)$ is a function depending on the dimension n only. The integer optimization problem is treated in Section 5. Again three algorithms are described; first binary search, second a more involved algorithm that solves the problem in linear time when the number of constraints is fixed, and finally a randomized algorithm which reduces the dependence of the complexity on the number of constraints. In Section 6 we take another view of solving integer feasibility problems. Here we try to construct a lattice in which we can prove that solutions to the considered problems are short vectors in that lattice. Solutions, if they exist, can then be found by considering bases of the lattice in which the basis vectors are short. Finally, in Section 7 we review various results regarding cutting planes if, again, the dimension is fixed. Even though little explicit use is made of lattices in this section, the results tie in well with the results discussed in Sections 4–6, and address several complexity questions that are naturally raised in the context of integer programming in fixed dimension.

2 Notation and basic definitions

To make our chapter more accessible we present some basic notation and definitions in the following two subsections.

2.1 Numbers, vectors, matrices, and polyhedra

The set of real (integer, rational) numbers is denoted by $\mathbb{R}$ ($\mathbb{Z}$, $\mathbb{Q}$). If we require nonnegativity we use the notation $\mathbb{R}_{\geq 0}$, $\mathbb{Z}_{\geq 0}$, and $\mathbb{Q}_{\geq 0}$ respectively. The set of natural numbers is denoted by $\mathbb{N}$ and if we consider positive natural numbers we use the notation $\mathbb{N}_{>0}$. When we write $\boldsymbol{x}_j$ we mean the j-th vector in a sequence of vectors. The i-th component of a vector $\boldsymbol{x}$ will be denoted by x_i, and the i-th component of the vector $\boldsymbol{x}_j$ is written x^i_j. The *Euclidean length* of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is denoted by $\|\boldsymbol{x}\|$ and is computed as $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$, where $\boldsymbol{x}^T$ is the transpose of the vector $\boldsymbol{x}$. An $m \times n$ matrix $\boldsymbol{A}$ has columns $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$, and element (i, j) of $\boldsymbol{A}$ is denoted by a_{ij}. We use $(\boldsymbol{c})^{(m\times n)}$ to denote an $m \times n$ matrix in which all elements are equal to c. The $n \times n$ identity matrix is denoted by $\boldsymbol{I}^{(n)}$, and when it is clear form the context the superscripts of $(\boldsymbol{c})^{(m\times n)}$ and $\boldsymbol{I}^{(n)}$ are dropped. Given an $m \times n$ matrix $\boldsymbol{A}$, the inequality

$$\sqrt{\det(\boldsymbol{A}^T\boldsymbol{A})} \leq \|\boldsymbol{a}_1\| \cdots \|\boldsymbol{a}_n\| \tag{1}$$

is known as the *Hadamard inequality.* An integer nonsingular matrix $\boldsymbol{U}$ is *unimodular* if $\det(\boldsymbol{U}) = \pm 1$. A matrix of full row rank is said to be in *Hermite Normal Form*, (HNF), if it has the form $(\boldsymbol{C}, (\boldsymbol{0})^{(m\times(n-m))})$, where $\boldsymbol{C}$ is a lower triangular nonnegative $m\times m$ matrix in which the unique row maxima can be found along the diagonal. A rational $m\times n$ matrix $\boldsymbol{A}$ of full row rank has a unique Hermite normal form, $\mathrm{HNF}(\boldsymbol{A}) = (\boldsymbol{C}, (\boldsymbol{0})^{(m\times(n-m))}) = \boldsymbol{AU}$, where $\boldsymbol{U}$ is unimodular.

We use the notation $\lfloor x \rfloor$ and $\lceil x \rceil$ for the round down and round up of the number x. We define $\lceil x \rfloor := \lceil x - \frac{1}{2} \rceil$. The *size* of an integer z is the number $\mathrm{size}(z) = 1 + \lceil \log_2(|z| + 1) \rceil$. Likewise, the size of a matrix $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$ is the number of bits needed to encode $\boldsymbol{A}$, i.e., $\mathrm{size}(\boldsymbol{A}) = mn + \sum_{i,j} \mathrm{size}(a_{ij})$, see [99, p. 29].

A *polyhedron* P is a set of vectors of the form $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} \leq \boldsymbol{d}\}$, for some matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and some vector $\boldsymbol{d} \in \mathbb{R}^m$. We write $P = P(\boldsymbol{A}, \boldsymbol{d})$. If P is given as $P(\boldsymbol{A}, \boldsymbol{d})$, then $\mathrm{size}(P) = \mathrm{size}(\boldsymbol{A}) + \mathrm{size}(\boldsymbol{d})$. The polyhedron $P = P(\boldsymbol{A}, \boldsymbol{d})$ is *rational* if both $\boldsymbol{A}$ and $\boldsymbol{d}$ can be chosen to be rational. If P is bounded, then P is called a *polytope.* The *integer hull* P_I of a polyhedron P is the convex hull of the integer vectors in P. If P is rational, then P_I is a rational polyhedron again. The *dimension* of P is the dimension of the affine hull of P.

A *rational halfspace* is a set of the form $H = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{c}^T\boldsymbol{x} \leq \delta\}$, for some non-zero vector $\boldsymbol{c} \in \mathbb{Q}^n$ and some $\delta \in \mathbb{Q}$. The halfspace H is then denoted by $(\boldsymbol{c}^T\boldsymbol{x} \leq \delta)$. The corresponding *hyperplane*, denoted by

$(\boldsymbol{c}^T\boldsymbol{x} = \delta)$, is the set $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{c}^T\boldsymbol{x} = \delta\}$. A rational half space always has a representation in which the components of $\boldsymbol{c}$ are relatively prime integers. That is, we can chose $\boldsymbol{c} \in \mathbb{Z}^n$ with $\gcd(c_1, \dots, c_n) = 1$.

An inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ is called *valid* for a polyhedron P, if $(\boldsymbol{c}^T\boldsymbol{x} \leq \delta) \supseteq P$. A *face* of P is a set of the form $F = (\boldsymbol{c}^T\boldsymbol{x} = \delta) \cap P$, where $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ is valid for P. The inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ is a *face-defining* inequality for F. Clearly F is a polyhedron. If $P \supset F \supset \emptyset$, then F is called *proper*. A maximal (inclusion wise) proper face of P is called a *facet* of P, i.e., a proper face F is a facet if and only if $\dim(F) = \dim(P) - 1$. If the face-defining inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ defines a facet of P, then $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ is a *facet-defining* inequality. A proper face of P of dimension 0 is called a *vertex* of P. A vertex $\boldsymbol{v}$ of $P(\boldsymbol{A}, \boldsymbol{d})$ is uniquely determined by a subsystem $\boldsymbol{A}^{\boldsymbol{v}}\boldsymbol{x} \leq \boldsymbol{d}^{\boldsymbol{v}}$ of $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$, where $\boldsymbol{A}^{\boldsymbol{v}}$ is nonsingular and $\boldsymbol{v} = (\boldsymbol{A}^{\boldsymbol{v}})^{-1}\boldsymbol{d}^{\boldsymbol{v}}$. If P is full-dimensional, then P has a unique (up to scalar multiplication) minimal set of inequalities defining P, which correspond to the facets of P. A polytope P can be described as the convex hull of its vertices. A *d-simplex* is a polytope, which is the convex hull of $d + 1$ affinely independent points.

Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron. The *facet complexity* of P is the smallest number φ satisfying

- $\varphi \geq n$, and
- there exists a system $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ of rational linear inequalities defining P such that each inequality in $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ has size at most φ.

The *vertex complexity* of P is the smallest number ν, such that there exist rational vectors $\boldsymbol{q}_1, \dots, \boldsymbol{q}_k, \boldsymbol{c}_1, \dots, \boldsymbol{c}_t$, each of size at most ν, with

$$P = \mathrm{conv}(\{\boldsymbol{q}_1, \dots, \boldsymbol{q}_k\}) + \mathrm{cone}(\{\boldsymbol{c}_1, \dots, \boldsymbol{c}_t\}).$$

Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron of facet complexity φ and vertex complexity ν. Then (see Schrijver [99])

$$\nu \leq 4n^2\varphi \text{ and } \varphi \leq 4n^2\nu. \tag{2}$$

We refer to Nemhauser and Wolsey [85] and Schrijver [99] for further basics on the topics treated in this subsection.

2.2 Lattices and lattice bases

Let $\boldsymbol{b}_1, \dots, \boldsymbol{b}_l$ be linearly independent vectors in $\mathbb{R}^n$. The set

$$L = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = \sum_{j=1}^{l} \lambda_j \boldsymbol{b}_j, \lambda_j \in \mathbb{Z},\ 1 \leq j \leq l\} \tag{3}$$

is called a *lattice*. The set of vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l\}$ is called a *lattice basis*. The vectors of a lattice L form an additive group, i.e., $\mathbf{0} \in L$, and if $\boldsymbol{x}$ belongs to L, so does $-\boldsymbol{x}$, and if $\boldsymbol{x}, \boldsymbol{y} \in L$, then $\boldsymbol{x} \pm \boldsymbol{y} \in L$. Moreover, the group L is *discrete*, i.e., there exits a real number $r > 0$ such that the n-dimensional ball with radius r, centered at the origin, does not contain any other element from L except the origin.

The *rank* of L, rk L, is equal to the dimension of the Euclidean vector space generated by a basis of L. The rank of the lattice L in expression (3) is l, and we have $l \leq n$. If $l = n$ we call the lattice *full-dimensional*. Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots \boldsymbol{b}_l)$. If we want to emphasize that we are referring to a lattice L that is generated by the basis $\boldsymbol{B}$, then we use the notation $L(\boldsymbol{B})$. Two matrices $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathbb{R}^{n \times l}$ are bases of the same lattice $L \subseteq \mathbb{R}^n$, if and only if $\boldsymbol{B}_1 = \boldsymbol{B}_2\boldsymbol{U}$ for some $l \times l$ unimodular matrix $\boldsymbol{U}$. The *shortest nonzero vector* in the lattice L is denoted by $\mathrm{SV}(L)$ or $\mathrm{SV}(L(\boldsymbol{B}))$.

We will frequently make use of *Gram-Schmidt orthogonalization*. The Gram-Schmidt process derives orthogonal vectors $\boldsymbol{b}_j^*$, $1 \leq j \leq l$, from linearly independent vectors $\boldsymbol{b}_j$, $1 \leq j \leq l$. The vectors $\boldsymbol{b}_j^*$, $1 \leq j \leq l$, and the real numbers μ_{jk}, $1 \leq k < j \leq l$, are determined from $\boldsymbol{b}_j$, $1 \leq j \leq l$, by the recursion

$$\begin{aligned}
\boldsymbol{b}_1^* &= \boldsymbol{b}_1 \\
\boldsymbol{b}_j^* &= \boldsymbol{b}_j - \sum_{k=1}^{j-1} \mu_{jk} \boldsymbol{b}_k^*, \quad 2 \leq j \leq l,
\end{aligned}$$

where

$$\mu_{jk} = \frac{\boldsymbol{b}_j^T \boldsymbol{b}_k^*}{||\boldsymbol{b}_k^*||^2}, \quad 1 \leq k < j \leq l.$$

The Gram-Schmidt process yields thus a factorization of the matrix $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ as

$$(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*) \cdot \boldsymbol{R}, \tag{4}$$

where $\boldsymbol{R}$ is the matrix

$$\boldsymbol{R} = \begin{pmatrix} 1 & \mu_{21} & \cdots & \mu_{n1} \\ 0 & 1 & \cdots & \mu_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \tag{5}$$

The vector $\boldsymbol{b}_j^*$ is the projection of $\boldsymbol{b}_j$ on the orthogonal complement of $\sum_{k=1}^{j-1} \mathbb{R}\boldsymbol{b}_k = \{\sum_{k=1}^{j-1} m_k \boldsymbol{b}_k : \ m_k \in \mathbb{R}, \ 1 \leq k \leq j-1\}$, i.e., $\boldsymbol{b}_j^*$ is the component of $\boldsymbol{b}_j$ orthogonal to the real subspace spanned by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{j-1}$. Thus, any pair $\boldsymbol{b}_i^*$, $\boldsymbol{b}_k^*$ of the Gram-Schmidt vectors are mutually orthogonal. The multiplier μ_{jk} gives the length, relative to $\boldsymbol{b}_k^*$, of the component of the vector $\boldsymbol{b}_j$ in direction $\boldsymbol{b}_k^*$. The multiplier μ_{jk} is equal to zero if and only if $\boldsymbol{b}_j$ is orthogonal to $\boldsymbol{b}_k^*$. Notice that the Gram-Schmidt vectors corresponding to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l$ do not in general belong to the lattice generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l$, but they do span the same real vector space as $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l$.

Let W be the vector space spanned by the lattice L, and let B_W be an orthonormal basis for W. The *determinant of the lattice* L, $d(L)$, is defined as the absolute value of the determinant of any nonsingular linear transformation $W \to W$ that maps B_W onto a basis of L. This is the l-dimensional volume of the parallelepiped spanned by the vectors of any basis of L. Below we give three different formulae for computing $d(L)$. Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l)$ be a basis for the lattice $L \subset \mathbb{R}^n$, with $l \leq n$, and let $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_l^*$ be the vectors obtained from applying the Gram-Schmidt orthogonalization procedure to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l$.

$$
\begin{aligned}
d(L) &= ||\boldsymbol{b}_1^*|| \cdot ||\boldsymbol{b}_2^*|| \cdot \ \cdots \ \cdot ||\boldsymbol{b}_l^*||, \\
d(L) &= \sqrt{\det(\boldsymbol{B}^T \boldsymbol{B})}\ , \\
d(L) &= \lim_{r \to \infty} \frac{|\{\boldsymbol{x} \in L : ||\boldsymbol{x}|| < r\}|}{\operatorname{vol}(B_l(r))},
\end{aligned}
\tag{6}
$$

where $\operatorname{vol}(B_l(r))$ is the volume of the l-dimensional ball with radius r. If L is full-dimensional, then $d(L(\boldsymbol{B}))$ can be interpreted as the volume of the parallelepiped $\sum_{j=1}^{n} [0,1)\boldsymbol{b}_j$. In this case the determinant of the lattice can be computed straightforwardly as $d(L(\boldsymbol{B})) = |\det(\boldsymbol{B})|$. The determinant of $\mathbb{Z}^n$ is equal to one. It is clear from Expression (6) that the determinant of a lattice depends only on the lattice and not on the choice of basis, see also Section 3. We will often use Hadamard's inequality (1) to bound the determinant of the lattice, i.e.,

$$
d(L(\boldsymbol{B})) = \sqrt{\det(\boldsymbol{B}^T \boldsymbol{B})} \leq ||\boldsymbol{b}_1|| \cdot \cdots \cdot ||\boldsymbol{b}_l||, \tag{7}
$$

where equality holds if and only if the basis $\boldsymbol{B}$ is orthogonal.

A convex set $K \in \mathbb{R}^n$ is *symmetric about the origin* if $\boldsymbol{x} \in K$ implies that $-\boldsymbol{x} \in K$. We will refer to the following theorem by Minkowski later in the chapter.

Theorem 1 (Minkowski's convex body theorem [83]). *Let K be a compact convex set in $\mathbb{R}^n$ of volume vol(K) that is symmetric about the* *origin. Let m be an integer an let L be a lattice of determinant $d(L)$. Suppose that vol$(K) \geq m2^n d(L)$. Then K contains at least m pairs of points $\pm \boldsymbol{x}_j,\ 1 \leq j \leq m$ that are distinct from each other and from the origin.*

Let L be a full-dimensional lattice in $\mathbb{R}^n$. Its *dual lattice* L^* is defined as

$$L^* = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^T\boldsymbol{y} \in \mathbb{Z} \text{ for all } \boldsymbol{y} \in L\}\,.$$

For a lattice L and its dual we have $d(L) = d(L^*)^{-1}$.

For more details about lattices, see e.g. Cassels [22], Grötschel, Lovász, and Schrijver [55], and Schrijver [99].

3 Lattice basis reduction

In several of the sections in this chapter we will use representations of lattices using bases that consist of vectors that are short and nearly orthogonal. In Section 3.1 we motivate why short lattice vectors are interesting objects, and we describe the basic principle of obtaining a new basis from a known basis of a given lattice. In Section 3.2 we describe Lovász' basis reduction algorithm, and some variants. The first vector in a Lovász-reduced basis is an approximation of the shortest non-zero lattice vector. In Section 3.3 we introduce Korkine-Zolotareff-reducedness and present Kannan's algorithm for computing the shortest non-zero lattice vector. We also discuss the complexity status of the shortest and closest lattice vector problem. In Section 3.4 we describe the generalized basis reduction algorithm by Lovász and Scarf, which uses a polyhedral norm instead of the Euclidean norm as in Lovász' algorithm. Finally, in Section 3.5 we discuss fast basis reduction algorithms in the bit model.

3.1 Reduced bases, an informal introduction

A lattice of rank at least two has infinitely many bases. Some of these bases are more useful than others, and in the applications we consider in this chapter we use bases whose elements are "nearly orthogonal". Such bases are called *reduced*. There are several definitions of reducedness, and some of them will be discussed in the following sections. Having a reduced basis makes it possible to obtain important bounds on both algorithmic running times and quality of solutions when lattice representations are used in integer programming and related areas. The study of reduced bases appears as early as in work by Gauß [49], Hermite [59], Minkowski [82], and Korkine and Zolotareff [72].

In many applications it becomes essential to determine the shortest nonzero vector in a lattice. In the following we motivate why an "almost orthogonal basis" helps us to find this vector. Suppose that $L \subseteq \mathbb{R}^n$ is generated by the basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ and assume that the vectors $\boldsymbol{b}_j$ are pairwise orthogonal. Consider a nonzero element $\boldsymbol{v} = \sum_{j=1}^n \lambda_j \boldsymbol{b}_j$ of the lattice, where $\lambda_j \in \mathbb{Z}$ for $j = 1, \ldots, n$. One has

$$\begin{aligned} \|\boldsymbol{v}\|^2 &= \left(\sum_{j=1}^n \lambda_j \boldsymbol{b}_j\right)^T \left(\sum_{j=1}^n \lambda_j \boldsymbol{b}_j\right) \\ &= \sum_{j=1}^n \lambda_j^2 \|\boldsymbol{b}_j\|^2 \\ &\geq \min\{\|\boldsymbol{b}_j\|^2 \mid j = 1, \ldots, n\}, \end{aligned}$$

where the last inequality follows from the fact that the λ_j are integers and not all of them are zero. Therefore the shortest vector of L is the shortest vector of the basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.

How do we determine the shortest vector of L if the basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is not orthogonal but "almost orthogonal"? The Gram-Schmidt orthogonalization procedure, see Section 2.2, computes pairwise orthogonal vectors $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ and an upper triangular matrix $R \in \mathbb{R}^{n \times n}$ whose diagonal entries are all one such that

$$(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l) = (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_l^*) \cdot \boldsymbol{R}$$

holds. Furthermore one has $\|\boldsymbol{b}_j\| \geq \|\boldsymbol{b}_j^*\|$ for $j = 1, \ldots, n$. This implies the Hadadmard inequality (7): $d(L) = \|\boldsymbol{b}_1^*\| \cdots \|\boldsymbol{b}_n^*\| \leq \|\boldsymbol{b}_1\| \cdots \|\boldsymbol{b}_n\|$, where equality holds if and only if the $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ are pairwise orthogonal. The number $c = \|\boldsymbol{b}_1\| \cdots \|\boldsymbol{b}_n\| / d(L)$ is called the *orthogonality defect* of the lattice basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. By "almost orthogonal" we mean that the orthogonality defect of a reduced basis is bounded by a constant that depends on the dimension n of the lattice only.

How does the orthogonality defect c come into play if one is interested in the shortest vector of a lattice? Again, consider a vector $\boldsymbol{v} = \sum_{j=1}^n \lambda_j \boldsymbol{b}_j$ of the lattice L generated by the basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ with orthogonality defect c. We now argue that if $\boldsymbol{v}$ is a shortest vector, then $|\lambda_j| \leq c$ for all j. This means that, with a reduced basis at hand, one only has to enumerate all $(2\,c + 1)^n$ vectors $(\lambda_1, \ldots, \lambda_n)$ with $|\lambda_j| \leq c$, compute the corresponding vector $\boldsymbol{v} = \sum_{j=1}^n \lambda_j \boldsymbol{b}_j$, and choose the shortest among them.

So suppose that one of the λ_j has absolute value strictly larger than c. Since the orthogonality defect is invariant under permutation of the basis vectors, we can assume that $j = n$. Consider the Gram-Schmidt orthogonalization $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. Since $\|\boldsymbol{b}_j^*\| \leq \|\boldsymbol{b}_j\|$ and since $\|\boldsymbol{b}_1\| \cdots \|\boldsymbol{b}_n\| \leq c\|\boldsymbol{b}_1^*\| \cdots \|\boldsymbol{b}_n^*\|$ one has $\|\boldsymbol{b}_n\| \leq c\|\boldsymbol{b}_n^*\|$ and thus

$$\begin{aligned}\|\boldsymbol{v}\| &= \|\lambda_n \boldsymbol{b}_n + \sum_{j=1}^{n-1} \lambda_j \boldsymbol{b}_j\| \\ &= \|\lambda_n \boldsymbol{b}_n^* + \boldsymbol{u}\|,\end{aligned}$$

where $\boldsymbol{u}$ is a vector in the subspace generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n-1}$. Since $\boldsymbol{u}$ and $\boldsymbol{b}_n^*$ are orthogonal we obtain

$$\begin{aligned}\|\boldsymbol{v}\| &= |\lambda_n| \, \|\boldsymbol{b}_n^*\| + \|\boldsymbol{u}\| \\ &> \|\boldsymbol{b}_n\|,\end{aligned}$$

which shows that $\boldsymbol{v}$ is not a shortest vector. Thus, a shortest vector of L can be computed from a basis with orthogonality defect c in $\mathrm{O}(c^{2n+1})$ steps.

In the following sections we present various reduction algorithms, and we begin with Lovász' algorithm that produces a basis with orthogonality defect bounded by $2^{n(n-1)/4}$. Lovász' algorithm runs in polynomial time in varying dimension. This implies that a shortest vector in a lattice can be computed from a Lovász-reduced basis by enumerating $(2 \cdot 2^{n(n-1)/4} + 1)^n = 2^{\mathrm{O}(n^3)}$ candidates, and thus in polynomial time if the dimension is fixed.

Before discussing specific basis reduction algorithms we describe the basic operations that are used to go from one lattice basis to another.

The following operations on a matrix are called *elementary column operations*:

- exchanging two columns,
- multiplying a column by -1,
- adding an integer multiple of one column to another column.

It is well known that a unimodular matrix can be derived from the identity matrix by elementary column operations.

To go from one basis to another is conceptually easy; given a basis $\boldsymbol{B}$ we just multiply $\boldsymbol{B}$ by a unimodular matrix, or equivalently, we perform a series of elementary column operations on $\boldsymbol{B}$, to obtain a new basis. The key question is of course how to do this efficiently such that the new basis is reduced according to the definition of reducedness we are using. In the following subsections we will describe some basis reduction algorithms, and highlight results relevant to integer programming.

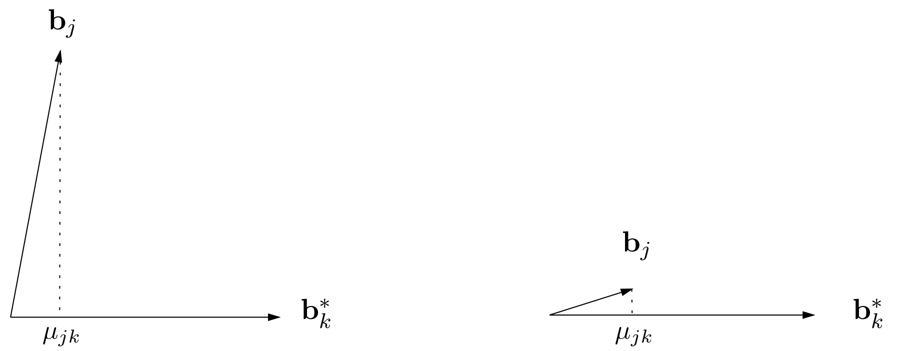
3.2 Lovász' basis reduction algorithm

In Lovász' [75] basis reduction algorithm the length of the vectors are measured using the Euclidean length, and the Gram-Schmidt vectors corresponding to the current basis are used as a reference for checking whether the basis vectors are nearly orthogonal. Let $L \subset \mathbb{R}^n$ be a lattice, and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l$, $l \le n$, be the current basis vectors for L. The vectors $\boldsymbol{b}_j^*$, $1 \le j \le l$, and the numbers μ_{jk}, $1 \le k < j \le l$ result from the Gram-Schmidt process as described in Section 2.2. A basis $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_l$ is called *reduced* in the sense of Lovász if

$$|\mu_{jk}| \le \frac{1}{2} \quad \text{for } 1 \le k < j \le l, \tag{8}$$

$$||\boldsymbol{b}_j^* + \mu_{j,j-1}\boldsymbol{b}_{j-1}^*||^2 \ge \frac{3}{4}||\boldsymbol{b}_{j-1}^*||^2 \quad \text{for } 1 < j \le l. \tag{9}$$

The constant $\frac{3}{4}$ in inequality (9) is arbitrarily chosen and can be replaced by any fixed real number $\frac{1}{4} < y < 1$. In a practical implementation one chooses a constant close to one. Below we explain why vectors satisfying Conditions (8) and (9) are relatively short and nearly orthogonal.



(a) $\mu_{jk} < 1/2$, since $\mathbf{b}_j$ is almost orthogonal to $\mathbf{b}_k^*$.

(b) $\mu_{jk} < 1/2$ since $\mathbf{b}_j$ is short compared to $\mathbf{b}_k^*$. If $k = j-1$, Condition (7) is violated.

Figure 1: Cases for which Condition (8) are satisfied.

Condition (8) is satisfied in two cases. The first case, see Figure 1(a), is if $\boldsymbol{b}_j$ is almost orthogonal to $\boldsymbol{b}_k^*$. Then, clearly, if we project $\boldsymbol{b}_j$ on $\boldsymbol{b}_k^*$, the absolute value of the length of this projection is going to be short relative to the length of $\boldsymbol{b}_k^*$. The second possibility for (8) to be satisfied, see Figure

1(b), is if $\boldsymbol{b}_j$ is short relative to $\boldsymbol{b}_k^*$. Even if $\boldsymbol{b}_j$ and $\boldsymbol{b}_k^*$ are not close to being orthogonal, the length of the projection of $\boldsymbol{b}_j$ on $\boldsymbol{b}_k^*$ will still be small relative to the length of $\boldsymbol{b}_k^*$. If we would accept this case we would also accept a basis in which $||\boldsymbol{b}_1|| >> ||\boldsymbol{b}_2|| >> \cdots >> ||\boldsymbol{b}_l||$, and where the vectors are far from being orthogonal. To prevent this, Condition (9) is enforced. Here we relate to the interpretation of the Gram-Schmidt vectors above, and notice that the vectors $\boldsymbol{b}_j^* + \mu_{j,j-1}\boldsymbol{b}_{j-1}^*$ and $\boldsymbol{b}_{j-1}^*$ are the projections of $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j-1}$ on the orthogonal complement of $\sum_{k=1}^{j-2} \mathbb{R}\boldsymbol{b}_k$. Consider the case where $k = j-1$, i.e., suppose that $\boldsymbol{b}_j$ is short compared to $\boldsymbol{b}_{j-1}^*$, which implies that $\boldsymbol{b}_j^*$ is short compared to $\boldsymbol{b}_{j-1}^*$ as $||\boldsymbol{b}_j^*|| \le ||\boldsymbol{b}_j||$. Suppose we *interchange* $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j-1}$. Then the new $\boldsymbol{b}_{j-1}^*$ will be the vector $\boldsymbol{b}_j^* + \mu_{j,j-1}\boldsymbol{b}_{j-1}^*$, which will be short compared to the old $\boldsymbol{b}_{j-1}^*$, i.e., Condition (9) will be violated.

Given a basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ one can apply a sequence of elementary column operations to obtain a basis satisfying (8) in the following way. Recall (see (4)) that the Gram-Schmidt process yields a factorization of the matrix $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ as $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*) \cdot \boldsymbol{R}$, where $\boldsymbol{R}$ is upper triangular, with all diagonal entries being equal to one. By subtracting integer multiples of column $\boldsymbol{r}_i$ from the columns $\boldsymbol{r}_{i+1}, \ldots, \boldsymbol{r}_n$, one can achieve that the elements $R(i,j)$ for $i < j$ are at most $1/2$ in absolute value. By doing so for $i = n-1, \ldots, 1$, in that order, one obtains a matrix $\boldsymbol{R}'$, which is upper triangular, with all diagonal elements equal to one, and all the elements above the diagonal being at most $1/2$ in absolute value. This yields a new basis $(\boldsymbol{b}_1', \ldots, \boldsymbol{b}_n') = (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*) \cdot \boldsymbol{R}'$, which satisfies (8). The replacement of the basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ by $(\boldsymbol{b}_1', \ldots, \boldsymbol{b}_n')$ is called *size reduction.* Notice that the Gram-Schmidt orthogonalization of $(\boldsymbol{b}_1', \ldots, \boldsymbol{b}_n')$ is given by $(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*) \cdot \boldsymbol{R}'$.

If Condition (9) is violated for a certain index j, then the vectors $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j-1}$ are *interchanged* to prevent us from accepting a basis with long non-orthogonal vectors as described in the previous paragraph. Lovász' basis reduction algorithm now performs size reductions and interchangings until the basis satisfies (8) and (9).

Algorithm 1 (Lovász' algorithm).

1. While Conditions (8) and (9) are not satisfied
   (a) Perform size reduction on the basis
   (b) If j is an index which violates (9), then interchange basis elements $j-1$ and j.

The key to the termination argument of Lovász' algorithm is the following potential function $\Phi(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ of a lattice basis $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$, $\boldsymbol{b}_j \in$

$\mathbb{Z}^n,\ 1 \leq j \leq n,$

$$\Phi(B) = \|\boldsymbol{b}_1^*\|^{2n}\|\boldsymbol{b}_2^*\|^{2(n-1)} \cdots \|\boldsymbol{b}_1^*\|^2.$$

The potential of an integer lattice basis is always an integer. Furthermore, an *interchange* step in Lovász' algorithm decreases the potential by a factor of 3/4 or a smaller number. Thus, if $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are two subsequent bases after an interchange step in Lovász' algorithm, then

$$\Phi(\boldsymbol{B}_2) \leq \frac{3}{4}\Phi(\boldsymbol{B}_1).$$

The potential of the input basis $\boldsymbol{B}$ can be bounded by $\Phi(\boldsymbol{B}) \leq (\|\boldsymbol{b}_1\| \cdots \|\boldsymbol{b}_n\|)^{2n}$. Therefore, the number of iterations of Lovász' algorithm is bounded by $\mathrm{O}(n(\log \|\boldsymbol{b}_1\| + \ldots + \|\boldsymbol{b}_n\|))$. In order to conclude that Lovasz' algorithm runs in polynomial time, one has further to show that the binary encoding lengths of the rational numbers representing the basis and the Gram-Schmidt orthogonalization remain polynomial in the input. For this, we refer to [75], where the following running time bound is given.

Theorem 2 ([75]). *Let $L \subseteq \mathbb{Z}^n$ be a lattice with basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, and let $\beta \in \mathbb{R}$, $\beta \geq 2$, be such that $||\boldsymbol{b}_j||^2 \leq \beta$ for $1 \leq j \leq n$. Then the number of arithmetic operations needed by the basis reduction algorithm as described in [75] is $O(n^4 \log \beta)$, and the integers on which these operations are performed each have binary length $O(n \log \beta)$.*

In terms of bit operations, Theorem 2 implies that Lovász' basis reduction algorithm has a running time of $O(n^6 (\log \beta)^3)$ using classical algorithms for addition and multiplication.

Example 1. Here we give an example of an initial and a reduced basis for a given lattice. Let L be the lattice generated by the vectors

$$\boldsymbol{b}_1 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \quad \boldsymbol{b}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The Gram-Schmidt vectors are $\boldsymbol{b}_1^* = \boldsymbol{b}_1$ and $\boldsymbol{b}_2^* = \boldsymbol{b}_2 - \mu_{21}\boldsymbol{b}_1^* = (1,1)^T - \frac{5}{17}\boldsymbol{b}_1^* = \frac{1}{17}(-3,12)^T$, see Figure 2a. Condition (8) is satisfied since $\boldsymbol{b}_2$ is short relative to $\boldsymbol{b}_1^*$. However, Condition (9) is violated, so we exchange $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$, giving

$$\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \boldsymbol{b}_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}.$$

We now have $\boldsymbol{b}_1^* = \boldsymbol{b}_1,\ \mu_{21} = \frac{5}{2}$ and $\boldsymbol{b}_2^* = \frac{1}{2}(3,-3)^T$, see Figure 2b.

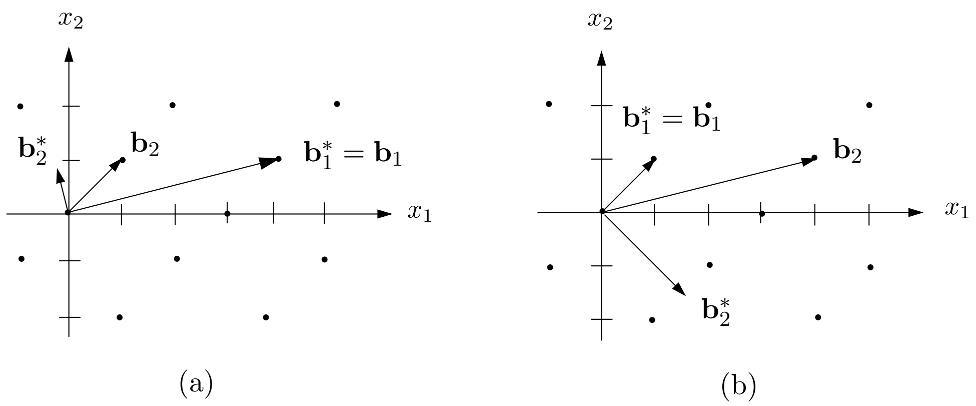


Figure 2:

Condition (8) is now violated, so we replace $\boldsymbol{b}_2$ by $\boldsymbol{b}_2 - 2\boldsymbol{b}_1 = (2, -1)^T$. Conditions (8) and (9) are satisfied for the resulting basis

$$\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \boldsymbol{b}_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix},$$

and hence this basis is reduced, see Figure 3.

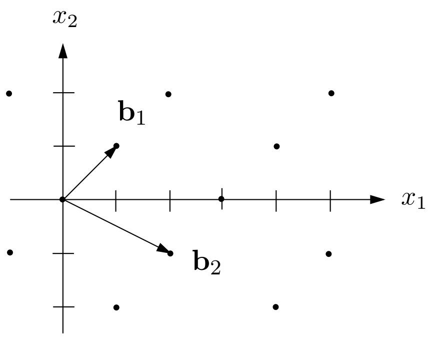


Figure 3: The reduced basis.

□

Next we will present some useful bounds on reduced basis vectors.

Proposition 1 ([75]). *Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a reduced basis for the lattice $L \subset \mathbb{R}^n$. Then,*

$$d(L) \leq \Pi_{j=1}^n ||\boldsymbol{b}_j|| \leq c_1 \cdot d(L), \tag{10}$$

where $c_1 = 2^{n(n-1)/4}$.

The first inequality in (10) is Hadamard's inequality (7) that holds for any basis of L. Recall that we refer to the ratio $\Pi_{j=1}^n ||\boldsymbol{b}_j||/d(L)$ as the orthogonality defect. Hermite [58] proved that each lattice $L \subset \mathbb{R}^n$ has a basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ such that $\Pi_{j=1}^n ||\boldsymbol{b}_j||/d(L) \leq c(n)$, where $c(n)$ is a constant depending only on n. The upper bound in (10) implies that the orthogonality defect of a Lovász-reduced basis is bounded from above by c_1. Better constants than c_1 are possible, but the question is then whether the basis can be obtained in polynomial time.

A consequence of Proposition 1 is that if we consider a basis that satisfies (10), and if $\boldsymbol{b}_n$ is the longest of the basis vectors, then the distance of $\boldsymbol{b}_n$ to the hyperplane generated by the basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n-1}$ is not too small as stated in the following corollary.

Corollary 1 ([76]). *Assume that* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ *is a basis such that (10) holds, and that, after possible reordering,* $||\boldsymbol{b}_n|| = \max_{1 \leq j \leq n}\{||\boldsymbol{b}_j||\}$. *Let* $H = \sum_{j=1}^{n-1} \mathbb{R}\boldsymbol{b}_j$ *and let* h *be the distance of basis vector* $\boldsymbol{b}_n$ *to* H. *Then*

$$c_1^{-1} \cdot ||\boldsymbol{b}_n|| \leq h \leq ||\boldsymbol{b}_n||, \tag{11}$$

where $c_1 = 2^{n(n-1)/4}$.

Proof: Let $L' = \sum_{j=1}^{n-1} \mathbb{Z}\boldsymbol{b}_j$. We have

$$d(L) = h \cdot d(L'). \tag{12}$$

Expressions (10) and (12) give

$$\Pi_{j=1}^n ||\boldsymbol{b}_j|| \leq c_1 \cdot d(L) = c_1 \cdot h \cdot d(L') \leq c_1 \cdot h \cdot \Pi_{j=1}^{n-1} ||\boldsymbol{b}_j||, \tag{13}$$

where the first inequality follows from the second inequality of (10), and where the last inequality follows from the first inequality of (10). From (13) we obtain $h \geq c_1^{-1}||\boldsymbol{b}_n||$. From the definition of h we have $h \leq ||\boldsymbol{b}_n||$, and this bound holds with equality if and only if the vector $\boldsymbol{b}_n$ is orthogonal to H. $\square$

The lower bound on h given in Corollary 1 plays a crucial role in the algorithm of H. W. Lenstra, Jr., which is described in Section 4.1.

Proposition 2 ([75]). *Let* $L \subset \mathbb{R}^n$ *be a lattice with reduced basis* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^n$. *Let* $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t \in L$ *be linearly independent. Then we have*

$$||\boldsymbol{b}_1||^2 \leq 2^{n-1}||\boldsymbol{x}||^2 \quad \textit{for all } \boldsymbol{x} \in L,\ \boldsymbol{x} \neq \boldsymbol{0}, \tag{14}$$

$$||\boldsymbol{b}_j||^2 \leq 2^{n-1} \max\{||\boldsymbol{x}_1||^2, ||\boldsymbol{x}_2||^2, \ldots, ||\boldsymbol{x}_t||^2\} \quad \textit{for } 1 \leq j \leq t. \tag{15}$$

Inequality (14) implies that the first reduced basis vector $\boldsymbol{b}_1$ is an approximation of the shortest nonzero vector in L.

Just as the first basis vector is an approximation of the shortest vector of the lattice (14), the other basis vectors are approximations of the *successive minima* of the lattice. The j-th successive minimum of $|| \ ||$ on L is the smallest positive value ν_j such that there exists j linearly independent elements of the lattice L in the ball of radius ν_j centered at the origin.

Proposition 3 ([75]). *Let $\nu_1, \ldots, \nu_l$ denote the successive minima of $|| \ ||$ on L, and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l$ be a reduced basis for L. Then*

$$2^{(1-j)/2}\nu_j \leq ||\boldsymbol{b}_j|| \leq 2^{(l-1)/2}\nu_j \quad \textit{for } 1 \leq j \leq l.$$

In recent years several new variants of Lovász' basis reduction algorithm have been developed and a number of variants for implementation have been suggested. We mention a few below, and recommend the paper by Schnorr and Euchner [93] for a more detailed overview. Schnorr [91] extended Lovász' algorithm to a family of polynomial time algorithms that, given $\epsilon > 0$, finds a non-zero vector in an n-dimensional lattice that is no longer than $(1 + \epsilon)^n$ times the length of the shortest vector in the lattice. The degree of the polynomial that bounds the running time of the family of algorithms increases as ϵ goes to zero. Seysen [101] developed an algorithm in which the intermediate integers that are produced are no larger than the input integers. Seysen's algorithm performs well particularly on lower-dimensional lattices. Schnorr and Euchner [93] discuss the possibility of computing the Gram-Schmidt vectors using floating point arithmetic while keeping the basis vectors in exact arithmetic in order to improve the practical performance of the algorithm. The drawback of this approach is that the basis reduction algorithm might become unstable. They propose a floating point version with good stability, but cannot prove that the algorithm always terminates. Their computational study indicates that their version is stable on instances of dimension up to 125 having input numbers of bit length as large as 300. Our experience is that one can use basis reduction for problems of larger dimensions if the input numbers are smaller, but once the dimension reaches about 300-400, basis reduction will be slow. Another version considered by Schnorr and Euchner is basis reduction *with deep insertions*. Here, they allow for a vector b_k to be swapped with a vector with lower index than $k - 1$. Schnorr [91], [92] also developed a variant of Lovász' algorithm in which not only two vectors are interchanged during the reduction process, but where blocks $\boldsymbol{b}_j, \boldsymbol{b}_{j+1}, \ldots, \boldsymbol{b}_{j+\beta-1}$ of β consecutive vectors are transformed so as to minimize the j-th Gram Schmidt vector $\boldsymbol{b}_j^*$.

This so called *block reduction* produces shorter basis vectors but needs more computing time. The shortest vector $\boldsymbol{b}_j^*$ in a block of size β is determined by complete enumeration of all short lattice vectors. Schnorr and Hörner [94] develop and analyze a rule for pruning this enumeration process.

For the reader interested in using a version of Lovász' basis reduction algorithm there are some useful libraries available on the Internet. Two of them are LiDIA - a C++ Library for Computational Number Theory [77] and NTL - a Library for doing Number Theory, developed by V. Shoup [102].

3.3 Korkine-Zolotareff reduction and fast algorithms for the shortest vector problem

As we have mentioned in Section 3.1, one can compute a shortest vector of a lattice that is represented by a Lovász-reduced basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ in $2^{\mathrm{O}(n^3)}$ steps via enumerating the candidates $\sum_{j=1}^n \lambda_j \, \boldsymbol{b}_j$, where $|\lambda_j| \leq 2^{n(n-1)/4}$ and choosing the shortest nonzero vector from this set.

Kannan [64, 66] provided an algorithm for the shortest vector problem, whose dependence on the dimension is $2^{\mathrm{O}(n \log n)}$. Helfrich [57] improved Kannan's algorithm. Recently, Ajtai, Kumar and Sivakumar [8] presented a randomized algorithm for the shortest vector problem, with an expected dependence of $2^{\mathrm{O}(n)}$. In the following, we briefly review the main idea of Kannan's algorithm and the improvement by Helfrich, see also [65]. Recall the Gram-Schmidt orthogonalization $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ of a lattice basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ from Section 2.2.

A lattice basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is *Korkine-Zolotareff reduced*, or *K-Z reduced* for short, if the following conditions hold.

1. The vector $\boldsymbol{b}_1$ is a shortest vector of the lattice generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.
2. The numbers μ_{jk} in the Gram-Schmidt orthogonalization of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ satisfy $|\mu_{jk}| \leq 1/2$, cf. Section 3.2, Expression (8).
3. If $\boldsymbol{b}_2', \ldots, \boldsymbol{b}_n'$ denote the projection of $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ onto the orthogonal complement of the space generated by $\boldsymbol{b}_1$, then $\boldsymbol{b}_2', \ldots, \boldsymbol{b}_n'$ is Korkine-Zolotareff reduced.

A two-dimensional lattice basis that is K-Z reduced is also called *Gauß* reduced, see [49]. The algorithm of Kannan computes a Korkine-Zolotareff reduced basis in dimension n by first computing a partially Korkine-Zolotareff reduced lattice basis, from which a shortest vector is among $2^{\mathrm{O}(n \log n)}$ candidates. The basis is partially Korkine-Zolotareff reduced with the help of an algorithm for Korkine-Zolotareff reduction in dimension $n-1$.

With a shortest vector at hand, one can then compute a fully K-Z reduced basis by K-Z reducing the projection along the orthogonal complement of this shortest vector. A lattice basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is *partially Korkine-Zolotareff reduced* or *partially K-Z reduced* for short, if it satisfies the following properties.

1. If $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n$ denotes the projection of $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ onto the orthogonal complement of the space generated by $\boldsymbol{b}_1$, then $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n$ is Korkine-Zolotareff reduced.

2. The numbers μ_{jk} in the Gram-Schmidt orthogonalization of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ satisfy $|\mu_{jk}| \leq 1/2$.

3. $\|\boldsymbol{b}'_2\| \geq 1/2 \, \|\boldsymbol{b}_1\|$.

Notice that, once Conditions 1 and 3 hold, Condition 2 can be satisfied, as explained in Section 3.2, via a size reduction step. Size reduction does not destroy Conditions 1 and 3. Condition 1 can be satisfied by applying Kannan's algorithm for full K-Z reduction to $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n$, and applying the transformation to the original vectors $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ If then Condition 3 is not satisfied, then Helfrich [57] has proposed to replace $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ with the *Gauß-reduction* of this pair, or equivalently its K-Z reduction. Clearly, if $\boldsymbol{b}_1, \boldsymbol{b}_2$ is Gauß-reduced, which means that $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$ and the angle enclosed by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ is at least $60°$ and at most $120°$, then Condition 3 holds.

The following algorithm computes a partially K-Z reduced basis from a given input basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. It uses as a subroutine an algorithm to K-Z reduce the lattice basis $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n$.

Algorithm 2 (Partial K-Z reduction).

1. Apply Lovàsz' basis reduction algorithm to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.

2. K-Z reduce $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n$ and apply the corresponding transformation to $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$.

3. Perform size reduction on $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.

4. If $\|\boldsymbol{b}'_2\| < 1/2 \, \|\boldsymbol{b}_1\|$, then replace $\boldsymbol{b}_1, \boldsymbol{b}_2$ by its Gauß reduction and go to Step 2.

We show in a moment that we can extract a shortest vector from a partially K-Z reduced basis in $2^{\mathrm{O}(n \log n)}$ steps, but before, we analyze the running time of the algorithm.

Theorem 3 ([57]). *Step 4 of Algorithm 2 is executed at most* $\log n + 6$ *times.*

Proof. Let $\boldsymbol{v}$ be a shortest vector and let $\boldsymbol{b}_1, \dots, \boldsymbol{b}_n$ be the lattice basis immediately before Step 4 of Algorithm 2 and let $\boldsymbol{b}_2', \dots, \boldsymbol{b}_n'$ denote the projection of $\boldsymbol{b}_2, \dots, \boldsymbol{b}_n$ onto the orthogonal complement of $\boldsymbol{b}_1$.

If Step 4 is executed, then $\boldsymbol{v}$ is not equal to $\boldsymbol{b}_1$. Then clearly, the projection of $\boldsymbol{v}$ onto the orthogonal complement of $\boldsymbol{b}_1$ is nonzero. Since $\boldsymbol{b}_2', \dots, \boldsymbol{b}_n'$ is K-Z reduced it follows that $\|\boldsymbol{v}\| \geq \|\boldsymbol{b}_2'\|$ holds. Denote the Gauß reduction of $\boldsymbol{b}_1, \boldsymbol{b}_2$ by $\widetilde{\boldsymbol{b}}_1, \widetilde{\boldsymbol{b}}_2$. The determinant of $L(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is equal to $\|\boldsymbol{b}_1\| \, \|\boldsymbol{b}_2'\|$. After the Gauß reduction in Step 4, we have therefore

$$\begin{aligned} \|\widetilde{\boldsymbol{b}}_1\| &\leq 2\sqrt{\|\boldsymbol{b}_1\| \, \|\boldsymbol{b}_2'\|} && (16) \\ &\leq 2\sqrt{\|\boldsymbol{b}_1\| \, \|\boldsymbol{v}\|}. && (17) \end{aligned}$$

Dividing this inequality by $\|\boldsymbol{v}\|$ gives

$$\frac{\|\widetilde{\boldsymbol{b}}_1\|}{\|\boldsymbol{v}\|} \leq 2\sqrt{\frac{\|\boldsymbol{b}_1\|}{\|\boldsymbol{v}\|}}.$$

Thus, if $\boldsymbol{b}_1^{(i)}$ denotes the first basis vector after the i-th execution of Step 4, one has

$$\frac{\|\boldsymbol{b}_1^{(i)}\|}{\|\boldsymbol{v}\|} \leq 4\left(\frac{\|\boldsymbol{b}_1^{(0)}\|}{\|\boldsymbol{v}\|}\right)^{(1/2)^i}. \tag{18}$$

Since we start with a Lovàsz reduced basis, we know that $\|\boldsymbol{b}_1^{(0)}\|/\|\boldsymbol{v}\| \leq 2^{(n-1)/2}$ holds, and consequently that $\|\boldsymbol{b}_1^{(\log n)}\|/\|\boldsymbol{v}\| \leq 8$. Each further Gauß reduction decreases the length of the first basis vector by at least $3/4$. Therefore the number of runs through Step 4 is bounded by $\log n + 6$. $\square$

We now argue, that with such a partially K-Z reduced basis $\boldsymbol{b}_1, \dots, \boldsymbol{b}_n$ at hand, one only needs to check $\mathrm{O}(n)^n$ candidates for the shortest vector. Let $\boldsymbol{v} = \sum_{j=1}^n \lambda_j \boldsymbol{b}_j$ be a shortest vector. After rewriting each $\boldsymbol{b}_j$ in terms of the Gram-Schmidt orthogonalization one obtains

$$\begin{aligned}
\boldsymbol{v} &= \sum_{j=1}^{n}\sum_{k=1}^{j}(\lambda_j \mu_{jk} \boldsymbol{b}_k^*) \\
&= \sum_{k=1}^{n}(\sum_{j=k}^{n}\lambda_j \mu_{jk})\boldsymbol{b}_k^*.
\end{aligned}$$

The length of $\boldsymbol{v}$ satisfies

$$\|\boldsymbol{v}\| = \sum_{k=1}^{n}|\sum_{j=k}^{n}(\lambda_j \mu_{jk})|\|\boldsymbol{b}_k^*\|. \tag{19}$$

Consider the coefficient $c_n = |\lambda_n \mu_{nn}| = |\lambda_n|$ of $\|\boldsymbol{b}_n^*\|$ in (19). We can bound this absolute value by $|\lambda_n| \leq \|\boldsymbol{v}\|/\|\boldsymbol{b}_n^*\| \leq \|\boldsymbol{b}_1\|/\|\boldsymbol{b}_n^*\|$. This leaves us $1+2\|\boldsymbol{b}_1\|/\|\boldsymbol{b}_n^*\|$ possibilities for λ_n. Suppose now that we picked $\lambda_n, \ldots, \lambda_{j+1}$ and inspect the coefficient c_j of $\|\boldsymbol{b}_j^*\|$ in (19), which is

$$\begin{aligned}
c_j &= |\sum_{k=j}^{n}(\lambda_k \mu_{kj})| \\
&= |\lambda_j + \sum_{k=j+1}^{n}(\lambda_k \mu_{kj})|.
\end{aligned}$$

Since the inequality $c_j \leq \|\boldsymbol{b}_1\|/\|\boldsymbol{b}_j^*\|$ must hold, this leaves only $1+2\,\|\boldsymbol{b}_1\|/\|\boldsymbol{b}_j^*\|$ possibilities to pick λ_j. Thus by choosing the coefficients $\lambda_n, \ldots, \lambda_1$ in this order, one has at most $\prod_{j=1}^{n}(1+2\,\|\boldsymbol{b}_1\|/\|\boldsymbol{b}_j^*\|)$ candidates.

Suppose $\|\boldsymbol{b}_j^*\| > \|\boldsymbol{b}_1\|$ for some j. Then $\boldsymbol{b}_j$ can never have a nonzero coefficient λ_j in a shortest vector representation $\boldsymbol{v} = \sum_{j=1}^{n}\lambda_j \boldsymbol{b}_j$. Because in that case, $\boldsymbol{v}$ has a nonzero component in its projection to the orthogonal complement of $\boldsymbol{b}_1\mathbb{R} + \ldots + \boldsymbol{b}_{i-1}\mathbb{R}$ and since $\boldsymbol{b}_2', \ldots, \boldsymbol{b}_n'$ is K-Z reduced, this implies that $||\boldsymbol{v}|| \geq \|\boldsymbol{b}_j^*\| > \|\boldsymbol{b}_1\|$, which is impossible. Thus we can assume that $\|\boldsymbol{b}_j^*\| \leq \|\boldsymbol{b}_1\|$ holds for all $j = 1, \ldots, n$. Otherwise, $\boldsymbol{b}_j$ can be discarded. Therefore the number of candidates N for the tuples $(\lambda_1, \ldots, \lambda_n)$ satisfies

$$\begin{aligned}
N &\leq \prod_{j=1}^{n}(1+2\,\|\boldsymbol{b}_1\|/\|\boldsymbol{b}_j^*\|) \\
&\leq \prod_{j=1}^{n}(3\,\|\boldsymbol{b}_1\|/\|\boldsymbol{b}_j^*\|) \\
&= 3^n\,\|\boldsymbol{b}_1\|^n/d(L).
\end{aligned}$$

Next we give an upper bound for $\|\boldsymbol{b}_1\|$. If $\boldsymbol{b}_1$ is a shortest vector, then Minkowski's theorem, (Theorem 1 in Section 2.2) guarantees that $\|\boldsymbol{b}_1\| \leq \sqrt{n}\; d(L)^{1/n}$ holds. If $\boldsymbol{b}_1$ is not a shortest vector, then the shortest vector $\boldsymbol{v}$ has a nonzero projection onto the orthogonal complement of $\boldsymbol{b}_1 \mathbb{R}$. Since $\boldsymbol{b}_2', \dots, \boldsymbol{b}_n'$ is K-Z reduced, this implies that $\|\boldsymbol{v}\| \geq \|\boldsymbol{b}_2'\| \geq 1/2\,\|\boldsymbol{b}_1\|$, since the basis is partially K-Z reduced. In any case we have $\|\boldsymbol{b}_1\| \leq 2\sqrt{n}\; d(L)^{1/n}$ and thus that $N \leq 6^n\, n^{n/2}$.

Now it is clear how to compute a K-Z reduced basis and thus a shortest vector. With an algorithm for K-Z reduction in dimension $n-1$, one uses Algorithm 2 to partially K-Z reduce the basis and then one checks all possible candidates for a shortest vector. Then one performs K-Z reduction on the basis for the projection onto the orthogonal complement of the shortest vector. Kannan [66] has shown that this procedure for K-Z reduction requires $\mathrm{O}(n)^n\,\varphi$ operations, where φ is the binary encoding length of the initial basis and where the operands during the execution of the algorithm have at most $\mathrm{O}(n^2\varphi)$ bits.

Theorem 4 ([66]). *Let $\boldsymbol{b}_1, \dots, \boldsymbol{b}_n$ be a lattice basis of binary encoding length φ. There exists an algorithm which computes a K-Z reduced basis of $L(\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ with $\mathrm{O}(n)^n\,\varphi$ arithmetic operations on rationals of size $\mathrm{O}(n^2\varphi)$.*

Further notes. Van Emde Boas [45] proved that the shortest vector problem with respect to the l_∞ norm is NP-hard, and he conjectured that it is NP-hard with respect to the Euclidean norm. In the same paper he proved that the closest vector problem is NP-hard for any norm. Recently substantial progress has been made in gaining more information about the complexity status of the two problems. Ajtai [7] proved that the shortest vector problem is NP-hard for randomized problem reductions. This means that the reduction makes use of results of a probabilistic algorithm. These results are true with probability arbitrarily close to one. Ajtai also showed that approximating the length of a shortest vector in a given lattice within a factor $1 + 1/2^{n^c}$ is NP-hard for some constant c. The non-approximability factor was improved to $(1 + 1/n^\epsilon)$ by Cai and Nerurkar [21]. Micciancio [81] improved this factor substantially by showing that it is NP-hard to approximate the shortest vector in a given lattice within any constant factor less that $\sqrt{2}$ for randomized problem reductions, and that the same result holds for deterministic problem reductions (the "normal" type of reductions used in an NP-hardness proof) under the condition that a certain number theoretic conjecture holds. Micciancio's results hold for any l_p norm. Goldreich and

Goldwasser [51] proved that it is not NP-hard to approximate the shortest vector, or the closest vector, within a factor $\sqrt{n}$ unless the polynomial-time hierarchy collapses. Goldreich et al. [52] show that, given oracle access to a subroutine that returns approximate closest vectors in a given lattice, one can find in polynomial time approximate shortest vectors in the same lattice with the same approximation factor. This implies that the shortest vector problem is not harder than the closest vector problem. From the other side, Kannan [65] showed that any algorithm producing an approximate shortest vector with approximation factor $f(n)$, where $f(n)$ is a nondecreasing function, can be used to produce an approximate closest vector to whithin $n^{3/2}f(n)^2$. For a recent overview on complexity results related to lattice problems, see for instance Cai [20], and Nguyen and Stern [87].

Kannan [66] also developed an exact algorithm for the closest vector problem, see also Helfrich [57] and Blömer [14].

3.4 The generalized basis reduction algorithm

In the generalized basis reduction algorithm a norm related to a full-dimensional compact convex set C is used, instead of the Euclidean norm as in Lovász' algorithm. A compact convex set $C \subseteq \mathbb{R}^n$ that is symmetric about the origin gives rise to a norm $F(\boldsymbol{c}) = \inf\{t \geq 0 \mid \boldsymbol{c}/t \in C\}$. Lovász and Scarf [79] call the function F the *distance function* with respect to C. As in Lovász' basis reduction algorithm, the generalized basis reduction algorithm finds short basis vectors with respect to the chosen norm. Moreover, the first basis vector is an approximation of the shortest nonzero lattice vector.

Given the convex set C we define a dual set $C^* = \{\boldsymbol{y} \mid \boldsymbol{y}^T\boldsymbol{c} \leq 1 \text{ for all } \boldsymbol{c} \in C\}$. We also define a distance function associated with a projection of C. Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a basis for $\mathbb{Z}^n$, and let C_j be the projection of C onto the orthogonal complement of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{j-1}$. We have that $\boldsymbol{c} = \beta_j\boldsymbol{b}_j + \cdots + \beta_n\boldsymbol{b}_n \in C_j$ if and only if there exist $\alpha_1, \ldots, \alpha_{j-1}$ such that $\boldsymbol{c} + \alpha_1\boldsymbol{b}_1 + \cdots + \alpha_{j-1}\boldsymbol{b}_{j-1} \in C$. The distance function associated with C_j is defined as:

$$F_j(\boldsymbol{c}) = \min_{\alpha_1,\ldots,\alpha_{j-1}} F(\boldsymbol{c} + \alpha_1\boldsymbol{b}_1 + \cdots + \alpha_{j-1}\boldsymbol{b}_{j-1}). \tag{20}$$

Using duality, one can show that $F_j(\boldsymbol{c})$ is also the optimal value of the maximization problem:

$$F_j(\boldsymbol{c}) = \max\{\boldsymbol{c}^T\boldsymbol{z} \mid \boldsymbol{z} \in C^*,\ \boldsymbol{b}_1^T\boldsymbol{z} = 0, \ldots, \boldsymbol{b}_{j-1}^T\boldsymbol{z} = 0\}. \tag{21}$$

In Expression (21), note that only vectors $\boldsymbol{z}$ that are orthogonal to the basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{j-1}$ are considered. This is similar to the role played by the

Gram-Schmidt basis in Lovász' basis reduction algorithm. Also, notice that if C is a polytope, then (21) is a linear program. The distance function F has the following properties:

- F can be computed in polynomial time,
- F is convex,
- $F(-\boldsymbol{x}) = F(\boldsymbol{x})$,
- $F(t\boldsymbol{x}) = tF(\boldsymbol{x})$ for $t > 0$.

Lovász and Scarf use the following definition of a reduced basis. A basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is called *reduced* in the sense of Lovász and Scarf if

$$F_j(\boldsymbol{b}_{j+1} + \mu \boldsymbol{b}_j) \geq F_j(\boldsymbol{b}_{j+1}) \quad \text{for } 1 \leq j \leq n-1 \text{ and all integers } \mu, \tag{22}$$

$$F_j(\boldsymbol{b}_{j+1}) \geq (1-\epsilon) F_j(\boldsymbol{b}_j) \quad \text{for } 1 \leq j \leq n-1, \tag{23}$$

where ϵ satisfies $0 < \epsilon < \frac{1}{2}$. A basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, not necessarily reduced, is called *proper* if

$$F_k(\boldsymbol{b}_j + \mu \boldsymbol{b}_k) \geq F_k(\boldsymbol{b}_j) \quad \text{for } 1 \leq k < j \leq n. \tag{24}$$

The algorithm is called *generalized* basis reduction since it generalizes Lovász' basis reduction algorithm in the following sense. If the convex set C is an ellipsoid, then a proper reduced basis is precisely a Lovász-reduced basis.

An important question is how to check whether Condition (22) is satisfied for all integers μ. Here we make use of the dual relationship between Formulations (20) and (21). We have the following equality: $\min_{\alpha \in \mathbb{R}} F_j(\boldsymbol{b}_{j+1} + \alpha \boldsymbol{b}_j) = F_{j+1}(\boldsymbol{b}_{j+1})$. Let α^* denote the optimal α in the minimization. The function F_j is convex, and hence the integer μ that minimizes $F_j(\boldsymbol{b}_{j+1} + \mu \boldsymbol{b}_j)$ is either $\lfloor \alpha^* \rfloor$ or $\lceil \alpha^* \rceil$. If the convex set C is a rational polytope, then $\alpha^* \in \mathbb{Q}$ is the optimal dual variable corresponding to the constraint $\boldsymbol{b}_j^T \boldsymbol{z} = 0$ in the optimization problem $F_{j+1}(\boldsymbol{b}_{j+1})$, cf. (21), which implies that the integer μ that minimizes $F_j(\boldsymbol{b}_{j+1} + \mu \boldsymbol{b}_j)$ can be determined by solving two additional linear programs, unless α^* is integral.

Condition (24) is analogous to Condition (8) of Lovász' basis reduction algorithm, and is violated if adding an integer multiple of $\boldsymbol{b}_k$ to $\boldsymbol{b}_j$ yields a distance function value $F_k(\boldsymbol{b}_j + \mu \boldsymbol{b}_k)$ that is smaller than $F_k(\boldsymbol{b}_j)$. In the generalized basis reduction algorithm we only check whether the condition is satisfied for $k = j-1$ (cf. Condition (22)), and we use the value of μ that minimizes $F_j(\boldsymbol{b}_{j+1} + \mu \boldsymbol{b}_j)$ as mentioned above. If Condition (22) is violated, we do a *size reduction*, i.e., we replace $\boldsymbol{b}_{j+1}$ by $\boldsymbol{b}_{j+1} + \mu \boldsymbol{b}_j$.

Condition (23) corresponds to Condition (9) in Lovász' algorithm, and ensures that the basis vectors are in the order of increasing distance function value, aside from the factor $(1-\epsilon)$. Recall that we want the first basis vector to be an approximation of the shortest lattice vector. If Condition (23) is violated we interchange vectors $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j+1}$.

The algorithm works as follows. Let C be a compact convex set, and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be an initial basis for $\mathbb{Z}^n$. Typically $\boldsymbol{b}_j = \boldsymbol{e}_j$, where $\boldsymbol{e}_j$ is the j-th unit vector in $\mathbb{R}^n$. Let j be the first index for which Conditions (22) or (23) are not satisfied. If (22) is violated, we replace $\boldsymbol{b}_{j+1}$ by $\boldsymbol{b}_{j+1} + \mu \boldsymbol{b}_j$ with the appropriate value of μ. If Condition (23) is satisfied after the replacement, we let $j := j + 1$. If Condition (23) is violated, we interchange $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j+1}$, and let $j := j - 1$ if $j \geq 2$. If $j = 1$, we remain at this level. The operations that the algorithm performs on the basis vectors are elementary column operations as in Lovász' algorithm. The vectors that we obtain as output from the generalized basis reduction algorithm can therefore be written as the product of the initial basis matrix and a unimodular matrix, which implies that the output vectors form a basis for the lattice $\mathbb{Z}^n$. The question is how efficient the algorithm is.

Theorem 5 ([79]). *Let ϵ be chosen as in (23), let $\gamma = 2 + 1/\log(1/(1-\epsilon))$, and let $B(R)$ be a ball with radius R containing C. Moreover, let $U = \max_{1 \leq j \leq n}\{F_j(\boldsymbol{b}_j)\}$, where $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is the initial basis, and let $V = 1/(R(nRU)^{n-1})$.*

The generalized basis reduction algorithm runs in polynomial time for fixed n. The maximum number of interchanges performed during the execution of the algorithm is

$$\left(\frac{\gamma^n - 1}{\gamma - 1}\right)\left(\frac{\log(U/V)}{\log(1/(1-\epsilon))}\right).$$

It is important to notice that, so far, the generalized basis reduction algorithm has been proved to run in polynomial time for *fixed* n only, whereas Lovász' basis reduction algorithm runs in polynomial time for arbitrary n (cf. Theorem 2).

We now give a few properties of a Lovász-Scarf reduced basis. If one can obtain a basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ such that $F_1(\boldsymbol{b}_1) \leq F_2(\boldsymbol{b}_2) \leq \cdots \leq F_n(\boldsymbol{b}_n)$, then one can prove that $\boldsymbol{b}_1$ is the shortest integer vector with respect to the distance function. The generalized basis reduction algorithm does not produce a basis with the above property, but it gives a basis that satisfies the following weaker condition.

Theorem 6 ([79]). *Let $0 < \epsilon < \frac{1}{2}$, and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a Lovász-Scarf reduced basis. Then*

$$F_{j+1}(\boldsymbol{b}_{j+1}) \geq (\frac{1}{2} - \epsilon) F_j(\boldsymbol{b}_j) \quad \textit{for } 1 \leq j \leq n-1.$$

We can use this theorem to obtain a result analogous to (14) of Proposition 2.

Proposition 4 ([79]). *Let $0 < \epsilon < \frac{1}{2}$, and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a Lovász-Scarf reduced basis. Then*

$$F(\boldsymbol{b}_1) \leq (\frac{1}{2} - \epsilon)^{1-n} F(\boldsymbol{x}) \quad \textit{for all } \boldsymbol{x} \in \mathbb{Z}^n, \boldsymbol{x} \neq \mathbf{0}.$$

We can also relate the distance function $F_j(\boldsymbol{b}_j)$ to the *j-th successive minimum* of F on the lattice $\mathbb{Z}^n$ (cf. Proposition 3). $\nu_1, \ldots, \nu_n$ are the *successive minima* of F on $\mathbb{Z}^n$ if there are vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{Z}^n$ with $\nu_j = F(\boldsymbol{x}_j)$, such that for each $1 \leq j \leq n$, $\boldsymbol{x}_j$ is the shortest lattice vector (with respect to F) that is linearly independent of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{j-1}$.

Proposition 5 ([79]). *Let $\nu_1, \ldots, \nu_n$ denote the successive minima of F on the lattice $\mathbb{Z}^n$, let $0 < \epsilon < \frac{1}{2}$, and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a Lovász-Scarf reduced basis. Then*

$$(\frac{1}{2} - \epsilon)^{j-1} \nu_j \leq F_j(\boldsymbol{b}_j) \leq (\frac{1}{2} - \epsilon)^{j-n} \nu_j \quad \textit{for } 1 \leq j \leq n.$$

The first reduced basis vector is an approximation of the shortest lattice vector (Proposition 4). In fact the generalized basis reduction algorithm can be used to find the shortest vector in the lattice in polynomial time for fixed n. This algorithm is used as a subroutine of Lovász and Scarf's algorithm for solving the integer programming problem "Is $X \cap \mathbb{Z}^n \neq \emptyset$?" described in Section 4.3. To find the shortest lattice vector we proceed as follows. If the basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is Lovász-Scarf reduced, we can obtain a bound on the coordinates of lattice vectors $\boldsymbol{c}$ that satisfy $F_1(\boldsymbol{c}) \leq F_1(\boldsymbol{b}_1)$. We express the vector $\boldsymbol{c}$ as an integer linear combination of the basis vectors, i.e., $\boldsymbol{c} = \lambda_1 \boldsymbol{b}_1 + \cdots + \lambda_n \boldsymbol{b}_n$, where $\lambda_j \in \mathbb{Z}$. We have

$$F_1(\boldsymbol{b}_1) \geq F_1(\boldsymbol{c}) \geq F_n(\boldsymbol{c}) = F_n(\lambda_n \boldsymbol{b}_n) = |\lambda_n| F_n(\boldsymbol{b}_n), \tag{25}$$

where the second inequality holds since $F_n(\boldsymbol{c})$ is more constrained than $F_1(\boldsymbol{c})$ (cf. (21)), the first equality holds due to the constraints $\boldsymbol{b}_i^T \boldsymbol{z} = 0,\ 1 \leq i \leq$

$n-1$, and the second equality holds as $F(t\boldsymbol{x}) = tF(\boldsymbol{x})$ for $t > 0$. We can now use (25) to obtain the following bound on $|\lambda_n|$:

$$|\lambda_n| \leq \frac{F_1(\boldsymbol{b}_1)}{F_n(\boldsymbol{b}_n)} \leq \frac{1}{(\frac{1}{2}-\epsilon)^{n-1}},$$

where the last inequality is obtained by applying Theorem 6 iteratively. Notice that the bound on λ_n is a constant for fixed n. In a similar fashion we can obtain a bound on λ_j for $n-1 \geq j \geq 1$. Suppose that we have chosen multipliers $\lambda_n, \ldots, \lambda_{j+1}$ and that we want to determine a bound on λ_j. Let γ^* be the value of γ that minimizes $F_j(\lambda_n \boldsymbol{b}_n + \cdots + \lambda_{j+1} b_{j+1} + \gamma \boldsymbol{b}_j)$. If this minimum is greater than $F_1(\boldsymbol{b}_1)$, then there does not exist a vector $\boldsymbol{c}$, with $\lambda_n, \ldots, \lambda_{j+1}$ fixed such that $F_1(\boldsymbol{c}) \leq F_1(\boldsymbol{b}_1)$, since in that case $F_1(\boldsymbol{b}_1) < F_j(\lambda_n \boldsymbol{b}_n + \cdots + \lambda_{j+1}\boldsymbol{b}_{j+1} + \gamma^* \boldsymbol{b}_j) \leq F_j(\lambda_n \boldsymbol{b}_n + \cdots + \lambda_j \boldsymbol{b}_j) = F_j(\boldsymbol{c}) \leq F_1(\boldsymbol{c})$, which yields a contradiction. If the minimum is less than or equal to $F_1(\boldsymbol{b}_1)$, then we can obtain the bound:

$$|\lambda_j - \gamma^*| \leq 2\frac{F_1(\boldsymbol{b}_1)}{F_j(\boldsymbol{b}_j)} \leq \frac{2}{(\frac{1}{2}-\epsilon)^{j-1}}.$$

Hence, we obtain a search tree that has at most n levels, and, given the bounds on the multipliers λ_j, each level consists of a constant number of nodes if n is fixed.

The generalized basis reduction algorithm was implemented by Cook, Rutherford, Scarf, and Shallcross [29], and by Wang [104]. Cook et al. used generalized basis reduction to derive a heuristic version of the integer programming algorithm by Lovász and Scarf (see Section 4.3) to solve difficult integer network design instances. Wang [104] solved both linear and nonlinear integer programming problems using the generalized basis reduction algorithm as a subroutine. For an small example on how to use the generalized basis reduction algorithm, we refer to Section 4.3, Example 2.

3.5 Fast algorithms in the bit model

The running times of the algorithms for lattice basis reduction depend on the number of bits that are necessary to represent the numbers of the input basis. The complexity model that reflects the fact that arithmetic operations on large numbers do not come for free is the *bit-complexity* model. Addition and subtraction of φ-bit integers takes $O(\varphi)$ time. The current state of the art method for multiplication [97] shows that the bit complexity $M(\varphi)$ of multiplication and division is $O(\varphi \log \varphi \log\log \varphi)$, see [6, p. 279].

The use of this complexity model is best illustrated with algorithms to compute the *greatest common divisor* of two integers. The *Euclidean algorithm* for computing the greatest common divisor $\gcd(a_0, a_1)$ of two integers $a_0, a_1 > 0$ computes the remainder sequence $a_0, a_1, \ldots a_{k-1}, a_k \in \mathbb{N}_{>0}$, where $a_i,\ i \geq 2$ is given by $a_{i-2} = a_{i-1}q_{i-1} + a_i$, with $q_i \in \mathbb{N},\ 0 < a_i < a_{i-1}$, and where a_k divides a_{k-1} exactly. If $a_0 = F_n$ and $a_1 = F_{n-1}$, were F_i denotes the i-th Fibonacci number, then the remainder sequence, generated by the Euclidean algorithm, is the sequence of Fibonacci numbers $F_n, F_{n-1}, \ldots, F_0$. Since the size of the n-th Fibonacci number is $\Theta(n)$, it follows that the Euclidean algorithm requires $\Omega(\varphi^2)$ bit-operations on an input of size φ. It can be shown, that the Euclidean algorithm runs in time $\Theta(\varphi^2)$ even if one uses the naive algorithms for basic arithmetic operations, see [71]. However, a gcd can be computed in $\mathrm{O}(M(\varphi)\log\varphi)$ bit operations with the algorithm of Schönhage [95].

The greatest common divisor of two integers a and b is the absolute value of the shortest vector of the 1-dimensional lattice $a\,\mathbb{Z} + b\,\mathbb{Z}$. Thus shortest vector computation and lattice basis reduction form a natural generalization of greatest common divisor computation. In this section we treat the dimension n as a constant and consider the bit-complexity of the shortest vector problem and lattice basis reduction in fixed dimension.

Schönhage [96] and Yap [105] proved that a 2-dimensional lattice basis can be K-Z reduced (or Gauß reduced) with $\mathrm{O}(M(\varphi)\log\varphi)$ bit-operations. In fact, 2-dimensional K-Z reduction can be solely based on Schönhage's [95] classical algorithm on the fast computation of continued fractions and the original reduction algorithm of Gauß [49], see [39].

Theorem 7 ([96, 105]). *Let $\boldsymbol{B} \in \mathbb{Z}^{2\times 2}$ be a two dimensional lattice basis with $\mathrm{size}(\boldsymbol{B}) = \varphi$. Then $\boldsymbol{B}$ can be K-Z reduced with $\mathrm{O}(M(\varphi)\log\varphi)$ bit-operations.*

Eisenbrand and Rote [43] showed that a lattice basis $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{Z}^{n\times n}$ of binary encoding length φ can be reduced in $\mathrm{O}(M(\varphi)\log^{n-1}\varphi)$ bit-operations. In this section we describe how this result can be obtained with the algorithm for partial K-Z reduction, presented in Section 3.3. For the three-dimensional case, van Sprang [103] and Semaev [100] provided an algorithm which requires $\mathrm{O}(\varphi^2)$ bit-operations, using the naive quadratic algorithms for multiplication and division.

Theorem 8. *Let $\boldsymbol{B} \in \mathbb{Z}^{n\times n}$ be a lattice basis with $\mathrm{size}(\boldsymbol{B}) = \varphi$. Then $\boldsymbol{B}$ can be K-Z reduced with $\mathrm{O}(M(\varphi)(\log\varphi)^{n-1})$ bit-operations.*

To prove this theorem, recall Algorithm 2 for partial K-Z reduction. We modify this algorithm as follows.

- Instead of computing a Lovász reduced basis in Step 1, compute the Hermite normal form of $\boldsymbol{B}$
- The stopping condition in Step 4 is modified, such that we go to Step 2 as long as $\|\boldsymbol{b}_1\| > 8\sqrt{n}\ d(L)^{1/n}$.

We assume that a $(n-1)$-dimensional rational lattice basis $\boldsymbol{B'} \in \mathbb{Z}^{(n-1)\times(n-1)}$ of size φ can be K-Z reduced with $\mathrm{O}(M(\varphi)(\log\varphi)^{n-2})$ bit operations.

We now analyze this modified algorithm. Recall that the HNF can be computed with a constant number of extended-gcd computations and a constant number of arithmetic operations, thus with $\mathrm{O}(M(\varphi)\log\varphi)$ bit-operations. If $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n$ is in Hermite normal form, then $\boldsymbol{b}_1$ is a vector which has zeroes in its $n-1$ first components, and a factor of the determinant in its last component. Thus, by swapping $\boldsymbol{b}_1$ and $\boldsymbol{b}_n$ one has a basis, whose first vector $\boldsymbol{b}_1$ satisfies $\|\boldsymbol{b}_1\| \le d(L)$.

Minkowski's theorem (Theorem 1 in Section 2.2) implies that the length of the shortest vector v of L is bounded by $\|v\| \le \sqrt{n}\ d(L)^{1/n}$. Thus in the proof of Theorem 3 we can replace inequality (17) by the inequality

$$\|\widetilde{\boldsymbol{b}}_1\| \le 2\sqrt{\|\boldsymbol{b}_1\|\sqrt{n}\ d(L)^{1/n}}.$$

Following the proof, we replace inequality (18) by

$$\frac{\|\boldsymbol{b}_1^{(i)}\|}{\sqrt{n}\ d(L)^{1/n}} \le 4\left(\frac{\|\boldsymbol{b}_1^{(0)}\|}{\sqrt{n}\ d(L)^{1/n}}\right)^{(1/2)^i}. \tag{26}$$

This means that after $\mathrm{O}(\log\log(d(L))$ iterations of the outer loop of the modified Algorithm 2, one has $\|\boldsymbol{b}_1\| \le 8\sqrt{n}\ d(L)^{1/n}$. It follows that the number of runs through the outer loop is bounded by a $\mathrm{O}(\log\varphi)$. Thus using the assumption that an $(n-1)$-dimensional lattice basis can be K-Z reduced in $\mathrm{O}(M(\varphi)(\log\varphi)^{n-2})$, we see that the modified Algorithm 2 runs with $\mathrm{O}(M(\varphi)(\log\varphi)^{n-1})$ bit-operations.

How quickly can the shortest vector be determined from the returned basis? Following the discussion preceding Theorem 4 we obtain the upper bound $N \le 3^n(8\cdot 8\cdot\sqrt{n}\cdot d(L)^{1/n})^n/d(L) = 24^n n^{n/2}$, which is a constant in fixed dimension. This proves Theorem 8.

It is currently not known whether a shortest vector can be computed in $\mathrm{O}(M(\varphi)\ \log\varphi)$ bit-operations.

4 Algorithms for the integer feasibility problem in fixed dimension

Let $\boldsymbol{A}$ be a rational $m \times n$-matrix and let $\boldsymbol{d}$ be a rational m-vector. Let $X = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$. We consider the integer feasibility problem in the following form:

$$\text{Does there exist an integer vector } \boldsymbol{x} \in X? \tag{27}$$

Karp [69] showed that the zero-one integer feasibility problem is NP-complete, and Borosh and Treybig [17] proved that the integer feasibility problem (27) belongs to NP. Combining these results implies that (27) is NP-complete. The NP-completeness of the zero-one version is a fairly straightforward consequence of the proof by Cook [26] that the satisfiability problem is NP-complete. An important open question was still: Can the integer feasibility problem be solved in polynomial time in bounded dimension? If the dimension $n = 1$, the affirmative answer is trivial. Some special cases of $n = 2$ were proven to be polynomially solvable by Hirschberg and Wong [60], and by Kannan [63]. Scarf [90] showed that (27), for the general case $n = 2$, is polynomially solvable. Both Hirschberg and Wong, and Scarf conjectured that the integer feasibility problem could be solved in polynomial time if the dimension is fixed. The proof of this conjecture was given by H. W. Lenstra, Jr. [76].

Let K be a full-dimensional closed convex set in $\mathbb{R}^n$ given by integer input. The *width of K along the nonzero integer vector $\boldsymbol{v}$* is defined as

$$w_{\boldsymbol{v}}(K) = \max\{\boldsymbol{v}^T\boldsymbol{x} : \boldsymbol{x} \in K\} - \min\{\boldsymbol{v}^T\boldsymbol{x} : \boldsymbol{x} \in K\}. \tag{28}$$

The *width* of K, $w(K)$, is the minimum of its widths along nonzero integer vectors $\boldsymbol{v} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$. Notice that this is different from the definition of the geometric width of a polytope (see p 6 in [54]). Khinchine [70] proved that if K does not contain a lattice point, then there exists a nonzero integer vector $\boldsymbol{c}$ such that $w_{\boldsymbol{c}}(K)$ is bounded from above by a constant depending only on the dimension.

Theorem 9 (Khinchine's flatness theorem [70]). *There exists a constant $f(n)$ depending only on the dimension n, such that each convex body $K \subseteq \mathbb{R}^n$ containing no integer points has width at most $f(n)$.*

Currently the best asymptotic bounds on $f(n)$ are given in [9]. Tight bounds seem to be unknown already in dimension 3.

To appreciate Khinchine's results we first have to interpret what the width of K in direction $\boldsymbol{v}$ means. To do that it is easier to look at the *integer width of* K *in the nonzero integer direction* $\boldsymbol{v}$, $w^I_{\boldsymbol{v}}(K) = \lfloor \max\{\boldsymbol{v}^T\boldsymbol{x} : \boldsymbol{x} \in K\}\rfloor - \lceil \min\{\boldsymbol{v}^T\boldsymbol{x} : \boldsymbol{x} \in K\}\rceil + 1$. The integer width of K in the direction $\boldsymbol{v}$ is the number of lattice hyperplanes intersecting K in direction $\boldsymbol{v}$. The width $w_{\boldsymbol{v}}(K)$ is an approximation of the integer width, so Khinchine's results says that if K is lattice point free, then there exists an integer vector $\boldsymbol{c}$ such that the number of lattice hyperplanes intersecting K in direction $\boldsymbol{c}$ is small. The direction $\boldsymbol{c}$ is often referred to as a "thin" direction, and we say that K is "thin" or "flat" in direction $\boldsymbol{c}$.

The algorithms we are going to describe in this section do not directly use Khinchine's flatness theorem, but they do use ideas that are related. First, we are going to find a point $\boldsymbol{x}$, not necessarily integer, that lies approximately in the center of the polytope X. Given the point $\boldsymbol{x}$ we can quickly find a lattice point $\boldsymbol{y}$ reasonably close to $\boldsymbol{x}$. Either $\boldsymbol{y}$ is also in X, in which case our feasibility problem is solved, or it is outside of X. If $\boldsymbol{y} \notin X$, then we know X cannot be too big since $\boldsymbol{x}$ and $\boldsymbol{y}$ are close. In particular, we can show that if we use a reduced basis and branch in the direction of the longest basis vector, then the number of lattice hyperplanes intersecting X is going to be bounded by a constant depending only on n. Then, for each of these hyperplanes we consider the polytope formed by the intersection of X with that polytope. This is a polytope in dimension less than or equal to $n-1$. For the new polytope we repeat the process. We can illustrate the algorithm by a search tree that has at most n levels, and a number of nodes at each level that is bounded by a constant depending only on the dimension on that level.

In the following three subsections we describe algorithms, based on the above idea, for solving the integer feasibility problem (27) in polynomial time for fixed dimension. Lenstra's algorithm is presented in Section 4.1. In Section 4.2 we present a version of Lenstra's algorithm that follows from Lovász' theorem on thin directions. Both of these algorithms use Lovász' basis reduction algorithm. In Section 4.3 we describe the algorithm of Lovász and Scarf [79], which is based on the generalized basis reduction algorithm. Finally, in Section 4.4 we give an outline of Barvinok's algorithm to count integer points in integer polytopes. This algorithm does not use "width" as the main concept, but exponential sums and decomposition of cones. Barvinok's algorithm runs in polynomial time if the dimension is fixed, so his result generalizes Lenstra's result.

4.1 Lenstra's algorithm

If one uses branch-and-bound for solving problem (27) it is possible, even in dimension 2, to create an arbitrarily deep search tree for certain thin polytopes, see e.g. [5]. Lenstra [76] suggested to transform the polytope using a linear transformation τ such that the polytope τX becomes "round" according to a certain measure. Assume without loss of generality that the polytope X is full-dimensional and bounded, and let $B(\boldsymbol{p}, z) = \{\boldsymbol{x} \in \mathbb{R}^n : ||\boldsymbol{x} - \boldsymbol{p}|| \leq z\}$ be the closed ball with center $\boldsymbol{p}$ and radius z. The transformation τ that we apply to the polytope is constructed such that $B(\boldsymbol{p}, r) \subset \tau X \subset B(\boldsymbol{p}, R)$ for some $\boldsymbol{p} \in \tau X$, with r, R satisfying

$$\frac{R}{r} \leq c_2, \tag{29}$$

where c_2 is a constant that depends only on the dimension n. Relation (29) is the measure of "roundness" that Lenstra uses. For an illustration, see Figure 4. Once we have transformed the polytope, we need to apply the

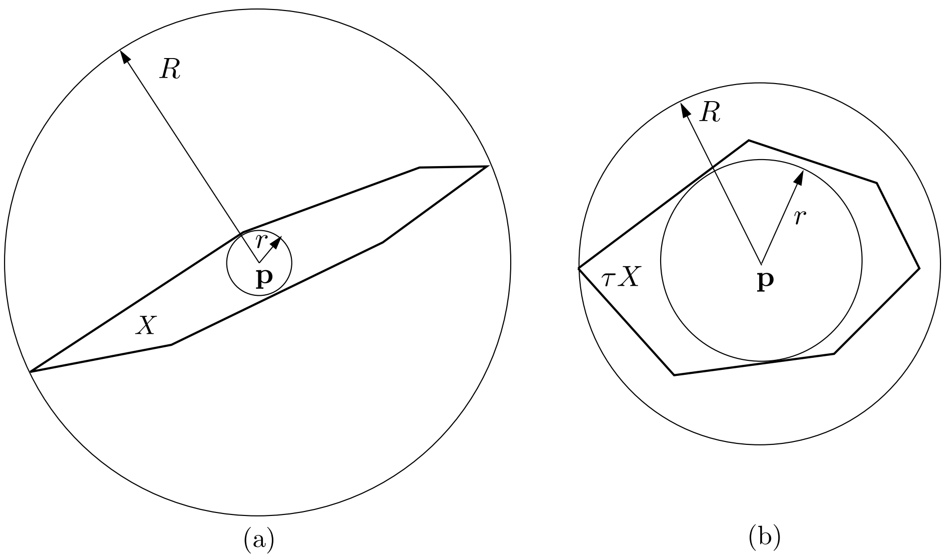


Figure 4: (a) The original polytope X is thin, and the ratio R/r is large. (b) The transformed polytope τX is "round", and R/r is relatively small.

same transformation to the lattice, which gives us the following feasibility problem that is equivalent to problem (27):

$$\text{Is } \tau\mathbb{Z}^n \cap \tau X \neq \emptyset? \tag{30}$$

The vectors $\tau\boldsymbol{e}_j$, $1 \le j \le n$, where $\boldsymbol{e}_j$ is the j-th unit vector in $\mathbb{R}^n$, form a basis for the lattice $\tau\mathbb{Z}^n$. If the polytope X is thin, then this will translate to the lattice basis vectors $\tau\boldsymbol{e}_j$, $1 \le j \le n$ in the sense that these vectors are long and non-orthogonal. This is where lattice basis reduction becomes useful. Once we have the transformed polytope τX, Lenstra uses the following lemma to find a lattice point quickly.

Lemma 1 ([76]). *Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be any basis for L. Then for all $\boldsymbol{x} \in \mathbb{R}^n$ there exists a vector $\boldsymbol{y} \in L$ such that*

$$||\boldsymbol{x} - \boldsymbol{y}||^2 \le \frac{1}{4}(||\boldsymbol{b}_1||^2 + \cdots + ||\boldsymbol{b}_n||^2).$$

The proof of this lemma suggests a fast construction of the vector $\boldsymbol{y} \in L$ given the vector $\boldsymbol{x}$.

Next, let $L = \tau\mathbb{Z}^n$, and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a basis for L such that (10) holds. Notice that (10) holds if the basis is reduced. Also, reorder the vectors such that $||\boldsymbol{b}_n|| = \max_{1 \le j \le n}\{||\boldsymbol{b}_j||\}$. Let $\boldsymbol{x} = \boldsymbol{p}$ where $\boldsymbol{p}$ is the center of the closed balls $B(\boldsymbol{p}, r)$ and $B(\boldsymbol{p}, R)$. Apply Lemma 1 to the given $\boldsymbol{x}$. This gives a lattice vector $\boldsymbol{y} \in \tau\mathbb{Z}^n$ such that

$$||\boldsymbol{p} - \boldsymbol{y}||^2 \le \frac{1}{4}(||\boldsymbol{b}_1||^2 + \cdots + ||\boldsymbol{b}_n||^2) \le \frac{1}{4} \cdot n \cdot ||\boldsymbol{b}_n||^2 \tag{31}$$

in polynomial time. We now distinguish two cases. Either $\boldsymbol{y} \in \tau X$ or $\boldsymbol{y} \notin \tau X$. In the first case we are done, so assume we are in the second case. Since $\boldsymbol{y} \notin \tau X$ we know that $\boldsymbol{y}$ is not inside the ball $B(\boldsymbol{p}, r)$ as $B(\boldsymbol{p}, r)$ is completely contained in τX. Hence we know that $||\boldsymbol{p} - \boldsymbol{y}|| > r$, or using (31), that

$$r < \frac{1}{2} \cdot \sqrt{n} \cdot ||\boldsymbol{b}_n||. \tag{32}$$

Below we will describe the tree search algorithm and argue why it is polynomial for fixed n. The distance between any two consecutive lattice hyperplanes, as defined in Corollary 1, is equal to h. We now create t subproblems by considering intersections between the polytope τX with t of these parallel hyperplanes. Each of the subproblems has dimension at least one lower than the parent problem and they are solved recursively. The procedure of splitting the problem into subproblems of lower dimension is called "branching", and each subproblem is represented by a node in the enumeration tree. In each node we repeat the whole process of transformation, basis reduction and, if necessary, branching. The enumeration tree created by this recursive process is of depth at most n, and the number of nodes at each level is bounded by a constant that depends only on the dimension. The value of t will be computed below.

Let H, h and L' be defined as in Corollary 1 of Section 3.2, and its proof. We can write L as

$$L = L' + \mathbb{Z}\boldsymbol{b}_n \subset H + \mathbb{Z}\boldsymbol{b}_n = \cup_{k\in\mathbb{Z}}(H + k\boldsymbol{b}_n). \tag{33}$$

So the lattice L is contained in countably many parallel hyperplanes. For an example we refer to Figure 5. The distance between two consecutive

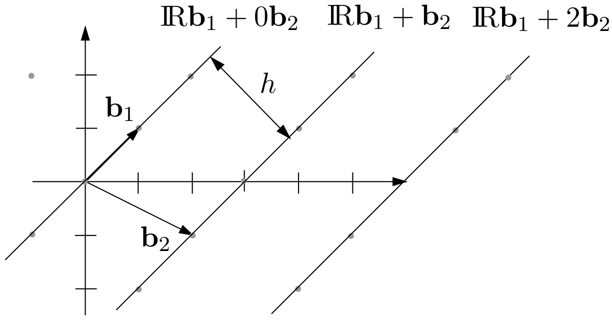


Figure 5:

hyperplanes is h, and Corollary 1 says that h is bounded from below by $c_1^{-1}||\boldsymbol{b}_n||$, which implies that not too many hyperplanes intersect τX. To determine precisely how many hyperplanes intersect τX, we approximate τX by the ball $B(\boldsymbol{p}, R)$. If t is the number of hyperplanes intersecting $B(\boldsymbol{p}, R)$ we have

$$t - 1 \leq \frac{2R}{h}.$$

Using the relationship (29) between the radii R and r we have $2R \leq 2rc_2 < c_2\sqrt{n}||\boldsymbol{b}_n||$, where the last inequality follows from (32). Since $h \geq c_1^{-1}||\boldsymbol{b}_n||$, we get the following bound on the number of hyperplanes that we need to consider:

$$t - 1 \leq \frac{2R}{h} < c_1 c_2\sqrt{n},$$

which depends on the dimension only. The values of the constants c_1 and c_2 that are used by Lenstra are: $c_1 = 2^{n(n-1)/4}$ and $c_2 = 2n^{3/2}$. Lenstra discusses ways of improving these values. To determine the values of k in expression (33), we express $\boldsymbol{p}$ as a linear combination of the basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. Recall that $\boldsymbol{p}$ is the center of the ball $B(\boldsymbol{p}, R)$ that was used to approximate τX.

So far we have not mentioned how to determine the transformation τ and hence the balls $B(\boldsymbol{p}, r)$ and $B(\boldsymbol{p}, R)$. We give the general idea here without going into detail. First, determine an n-simplex contained in X. This can be done in polynomial time by repeated calls to the ellipsoid algorithm. The resulting simplex is described by its extreme points $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_n$. By again applying the ellipsoid algorithm repeatedly we can decide whether there exists an extreme point $\boldsymbol{x}$ of X such that if we replace $\boldsymbol{v}_j$ by $\boldsymbol{x}$ we obtain a new simplex whose volume is at least a factor of $\frac{3}{2}$ larger than the current simplex. We stop the procedure if we cannot find such a new simplex. The factor $\frac{3}{2}$ can be modified, but the choice will affect the value of the constant c_2, see [76] for further details. We now map the extreme points of the simplex to the unit vectors of $\mathbb{R}^{n+1}$ so as to obtain a regular n-simplex, and we denote this transformation by τ. Lenstra [76] shows that τ has the property that if we let $\boldsymbol{p} = 1/(n+1) \sum_{j=0}^{n} \boldsymbol{e}_j$, where $\boldsymbol{e}_j$ is the j-th unit vector of $\mathbb{R}^{n+1}$ (i.e., $\boldsymbol{p}$ is the center of the regular simplex), then there exist closed balls $B(\boldsymbol{p}, r)$ and $B(\boldsymbol{p}, R)$ such that $B(\boldsymbol{p}, r) \subset \tau X \subset B(\boldsymbol{p}, R)$ for some $\boldsymbol{p} \in \tau X$, with r, R satisfying $R/r \leq c_2$.

Kannan [66] developed a variant of Lenstra's algorithm. The algorithm follows Lenstra's algorithm up to the point where he has applied a linear transformation to the polytope X and obtained a polytope τX such that $B(\boldsymbol{p}, r) \subset \tau X \subset B(\boldsymbol{p}, R)$ for some $\boldsymbol{p} \in \tau X$. Here Kannan proceeds as follows. He applies a reduction algorithm to a basis of the lattice $\tau \mathbb{Z}^n$ that produces a "reduced" basis defined differently to a Lovász' reduced basis. In particular, in Kannan's reduced basis the first basis vector is the shortest nonzero lattice vector. As in Lenstra's algorithm two cases are considered. Either τX is relatively large which implies that τX contains a lattice vector, or τX is small, which means that not too many lattice hyperplanes can intersect τX. Each such intersection gives rise to a subproblem of at least one dimension lower. Kannan's reduced basis makes it possible to improve the bound on the number of hyperplanes that has to be considered to $O(n^{5/2})$. Lenstra's algorithm has been implemented by Gao and Zhang [47], and a heuristic version of the algorithm has been developed and implemented by Aardal et al. [1], and Aardal and Lenstra [4]

4.2 Lovász' theorem on thin directions

Let $E(\boldsymbol{z}, \boldsymbol{D}) = \{\boldsymbol{x} \in \mathbb{R}^n \mid (\boldsymbol{x}-\boldsymbol{z})^T \boldsymbol{D}^{-1}(\boldsymbol{x}-\boldsymbol{z}) \leq 1\}$. $E(\boldsymbol{z}, \boldsymbol{D})$ is the *ellipsoid* in $\mathbb{R}^n$ associated with the vector $\boldsymbol{z} \in \mathbb{R}^n$ and the positive definite $n \times n$ matrix $\boldsymbol{D}$. The vector $\boldsymbol{z}$ is the *center* of the ellipsoid. Goffin [50] showed that for any full-dimensional rational polytope X it is possible, in polynomial time, to find a vector $\boldsymbol{p} \in \mathbb{Q}^n$ and a positive definite $n \times n$ matrix $\boldsymbol{D}$ such that

$$E(\boldsymbol{p}, \frac{1}{(n+1)^2}\boldsymbol{D}) \subseteq X \subseteq E(\boldsymbol{p}, \boldsymbol{D})\,. \tag{34}$$

Grötschel, Lovàsz and Schrijver [54] showed a similar result for the case where the polytope is not given explicitly, but by a separation algorithm.

The norm $//\ //$ defined by the matrix $\boldsymbol{D}^{-1}$ is given by $//\boldsymbol{x}// = \sqrt{\boldsymbol{x}\boldsymbol{D}^{-1}\boldsymbol{x}}$. Lovász used basis reduction with the norm $//\ //$, and the result by Goffin to obtain the following theorem.

Theorem 10 (see [99]). *Let $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ be a system of m rational inequalities in n variables, let $X = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$, and let $w_{\boldsymbol{c}}(X)$ be defined as in Expression (28). There exists a polynomial algorithm that finds either an integer vector $\boldsymbol{y} \in X$, or a vector $\boldsymbol{c} \in \mathbb{Z}^n \setminus \{\boldsymbol{0}\}$ such that*

$$w_{\boldsymbol{c}}(X) \leq n(n+1)2^{n(n-1)/4}$$

We will sketch the proof of the theorem for the case that X is full-dimensional and bounded. For the not full-dimensional case, and the case where P is unbounded we refer to the presentation by Schrijver [99]. Notice that the algorithm of Theorem 10 is polynomial for arbitrary n.

Proof of the full-dimensional bounded case: Assume that $dim(X) = n$. Here we will not make a transformation to a lattice $\tau\mathbb{Z}^n$, but remain in the lattice $\mathbb{Z}^n$. First, find two ellipsoids $E(\boldsymbol{p}, \frac{1}{(n+1)^2}\boldsymbol{D})$ and $E(\boldsymbol{p}, \boldsymbol{D})$, such that (34) holds, by the algorithm of Goffin. Next, we apply basis reduction, using the norm $//\ //$ defined by $\boldsymbol{D}^{-1}$, to the unit vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ to obtain a reduced basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ for the lattice $\mathbb{Z}^n$ that satisfies (cf. the second inequality of (10))

$$\Pi_{j=1}^n //\boldsymbol{b}_j// \leq 2^{n(n-1)/4}\sqrt{\det(\boldsymbol{D}^{-1})}. \tag{35}$$

Next, reorder the basis vectors such that $//\boldsymbol{b}_n// = \max_{1 \leq j \leq n}\{//\boldsymbol{b}_j//\}$. After reordering, inequality (35) still holds. Write $\boldsymbol{p} = \sum_{j=1}^n \alpha_j \boldsymbol{b}_j$, and let $\boldsymbol{y} =$

$\sum_{j=1}^{n}\lceil\alpha_j\rfloor\boldsymbol{b}_j$. Notice that $\boldsymbol{y}\in\mathbb{Z}^n$. If $\boldsymbol{y}\in X$ we are done, and if not we know that $\boldsymbol{y}\notin E(\boldsymbol{p},(1/(n+1)^2)\boldsymbol{D})$, so

$$\frac{1}{(n+1)^2}<(\boldsymbol{y}-\boldsymbol{p})^T\boldsymbol{D}^{-1}(\boldsymbol{y}-\boldsymbol{p})=//\boldsymbol{y}-\boldsymbol{p}//^2=//\sum_{j=1}^{n}(\alpha_j-\lceil\alpha_j\rfloor)\boldsymbol{b}_j//^2\,.$$

From this expression we obtain

$$\frac{1}{(n+1)}<\sum_{j=1}^{n}(\alpha_j-\lceil\alpha_j\rfloor)//\boldsymbol{b}_j//\leq\frac{n}{2}//\boldsymbol{b}_n//,$$

so

$$//\boldsymbol{b}_n//>\frac{2}{n(n+1)}. \tag{36}$$

Choose a direction $\boldsymbol{c}$ such that the components of $\boldsymbol{c}$ are relatively prime integers, and such that $\boldsymbol{c}$ is orthogonal to the subspace generated by the basis vectors $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{n-1}$. One can show, see Schrijver [99], pp 257–258, that if we consider a vector $\boldsymbol{x}$ such that $\boldsymbol{x}^T\boldsymbol{D}^{-1}\boldsymbol{x}\leq 1$, then

$$|\boldsymbol{c}^T\boldsymbol{x}|\leq\sqrt{\det(\boldsymbol{D})}//\boldsymbol{b}_1//\cdot\cdots\cdot//\boldsymbol{b}_{n-1}//\leq 2^{n(n-1)/4}//\boldsymbol{b}_n//^{-1}<\frac{n(n+1)}{2}2^{n(n-1)/4}, \tag{37}$$

where the second inequality follows from inequality (35), and the last inequality follows from (36). If $\boldsymbol{z}\in E(\boldsymbol{p},\boldsymbol{D})$, then

$$|\boldsymbol{c}^T(\boldsymbol{z}-\boldsymbol{p})|\leq\frac{n(n+1)}{2}2^{n(n-1)/4},$$

which implies

$$\begin{aligned} w_{\boldsymbol{c}}(X) &= \max\{\boldsymbol{c}^T\boldsymbol{x}\mid\boldsymbol{x}\in X\}-\min\{\boldsymbol{c}^T\boldsymbol{x}\mid\boldsymbol{x}\in X\} \qquad (38)\\ &\leq \max\{\boldsymbol{c}^T\boldsymbol{x}\mid\boldsymbol{x}\in E(\boldsymbol{p},\boldsymbol{D})\}-\min\{\boldsymbol{c}^T\boldsymbol{x}\mid\boldsymbol{x}\in E(\boldsymbol{p},\boldsymbol{D})\}\\ &\leq n(n+1)2^{n(n-1)/4}, \end{aligned}$$

which gives the desired result. □

Lenstra's result that the integer feasibility problem can be solved in polynomial time for fixed n follows from Theorem 10. If we apply the algorithm implied by Theorem 10, we either find an integer point $\boldsymbol{y}\in X$ or a thin direction $\boldsymbol{c}$, i.e., a direction $\boldsymbol{c}$ such that equation (38) holds. Assume that the direction $\boldsymbol{c}$ is the outcome of the algorithm. Let $\mu=\lceil\min\{\boldsymbol{c}^T\boldsymbol{x}\mid\boldsymbol{x}\in X\}\rceil$.

All points in $X \cap \mathbb{Z}^n$ are contained in the parallel hyperplanes $\boldsymbol{c}^T\boldsymbol{x} = t$ where $t = \mu, \ldots, \mu + n(n+1)2^{n(n-1)/4}$, so if n is fixed, then the number of hyperplanes is constant, and each of them gives rise to a subproblem of dimension less than or equal to $n-1$. For each of these lower-dimensional problems we repeat the algorithm of Theorem 10. The search tree has at most n levels and the number of nodes at each level is bounded by a constant depending only on the dimension.

Remark. The ingredients of Theorem 10 are actually present in Lenstra's paper [76]. In the preprinted version, however, the two auxiliary algorithms used by Lenstra; the algorithm to make the set X appear round, and the basis reduction algorithm, were polynomial for fixed n only, which was enough to prove his result that the integer programming feasibility problem can be solved in polynomial time in fixed dimension. Later, Lovász' basis reduction algorithm [75] was developed, and Lovász also pointed out that the "rounding" of X can be done in polynomial time for varying n due to the ellipsoid algorithm. Lenstra uses both of these algorithms in the published version of the paper.

4.3 The Lovász-Scarf algorithm

The integer feasibility algorithm of Lovász and Scarf [79] determines, in polynomial time for fixed n, either a certificate for feasibility, or a thin direction of X. If a thin direction is found, then one needs to branch, i.e., divide the problem into lower-dimensional subproblems, in order to determine whether or not a feasible vector exists, but then the number of branches is bounded by a constant for fixed n. If the algorithm indicates that X contains an integer vector, then one needs to determine a so-called Korkine-Zolotareff basis in order to construct a feasible vector. The Lovász-Scarf algorithm avoids the approximations by balls as in Lenstra's algorithm, or by ellipsoids as in the algorithm implied by Lovász' result. Again, we assume that $X = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$ is bounded, rational, and full-dimensional.

Let $(X - X) = \{(\boldsymbol{x} - \boldsymbol{y}) \mid \boldsymbol{x} \in X,\ \boldsymbol{y} \in X)\}$ be the difference set corresponding to X. Recall that $(X-X)^*$ denotes the dual set corresponding to $(X - X)$, and notice that $(X - X)^*$ is symmetric about the origin. The distance functions associated with $(X-X)^*$ are:

$$\begin{aligned} F_j(\boldsymbol{c}) &= \min_{\alpha_1,\ldots,\alpha_{j-1}\in\mathbb{Q}} F(\boldsymbol{c}+\alpha_1\boldsymbol{b}_1+\cdots+\alpha_{j-1}\boldsymbol{b}_{j-1}) \\ &= \max\{\boldsymbol{c}^T(\boldsymbol{x}-\boldsymbol{y}) \mid \boldsymbol{x}\in X,\ \boldsymbol{y}\in X,\ \boldsymbol{b}_1^T(\boldsymbol{x}-\boldsymbol{y})=0,\ldots,\boldsymbol{b}_{j-1}^T(\boldsymbol{x}-\boldsymbol{y})=0\}, \end{aligned}$$

(cf. expressions (20) and (21)). Here, we notice that $F(\boldsymbol{c}) = F_1(\boldsymbol{c})$ is the width of X in the direction $\boldsymbol{c}$, $w_{\boldsymbol{c}}(X)$ (see Expression (28) in the introduction to Section 4). From the above we see that a lattice vector $\boldsymbol{c}$ that minimizes the width of the polytope X is a *shortest lattice vector* for the polytope $(X-X)^*$.

To outline the algorithm by Lovász and Scarf we need the results given in Theorem 11 and 12 below, and the definition of a *generalized Korkine-Zolotareff basis.* Let $\boldsymbol{b}_j,\ 1\le j\le n$ be defined recursively as follows. Given $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{j-1}$, the vector $\boldsymbol{b}_j$ minimizes $F_j(\boldsymbol{x})$ over all lattice vectors that are linearly independent of $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{j-1}$. A generalized Korkine-Zolotareff (KZ) basis is defined to be any proper basis $\boldsymbol{b}'_1,\ldots,\boldsymbol{b}'_n$ associated with $\boldsymbol{b}_j$, $1\le j\le n$ (see Expression (24) for the definition of a proper basis). The notion of a generalized KZ basis was introduced by Kannan and Lovász [67], [68]. Kannan and Lovász [67] gave an algorithm for computing a generalized KZ basis in polynomial time for fixed n. Notice that $\boldsymbol{b}'_1$ in a generalized KZ basis is the shortest non-zero lattice vector.

Theorem 11 ([68]). *Let $F(\boldsymbol{c})$ be the length of the shortest non-zero lattice vector $\boldsymbol{c}$ with respect to the set $(X-X)^*$, and let $\rho_{\mathrm{KZ}} = \sum_{j=1}^n F_j(\boldsymbol{b}'_j)$, where $\boldsymbol{b}'_j$, $1\le j\le n$ is a generalized Korkine-Zolotareff basis. There exists a universal constant c_0 such that*

$$F(\boldsymbol{c})\rho_{\mathrm{KZ}} \le c_0\cdot n\cdot(n+1)/2.$$

To derive their result, Kannan and Lovász used a lower bound on the product of the volume of a convex set $C\subset\mathbb{R}^n$ that is symmetric about the origin, and the volume of its dual C^*. The bound, due to Bourgain and Milman [18], is equal to $\frac{c_{\mathrm{BM}}^n}{n^n}$, where c_{BM} is a constant depending only on n. In Theorem 11 we have $c_0 = \frac{4}{c_{\mathrm{BM}}}$, see also the remark below.

Theorem 12 ([68]). *Let $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n$ be any basis for $\mathbb{Z}^n$, and let X be a bounded convex set that is symmetric about the origin. If $\rho = \sum_{j=1}^n F_j(\boldsymbol{b}_j)\le 1$, then X contains an integer vector.*

The first step of the Lovász-Scarf algorithm is to compute the shortest vector $\boldsymbol{c}$ with respect to $(X-X)^*$ using the algorithm described in Section

3.4. If $F(\boldsymbol{c}) \geq c_0 \cdot n \cdot (n+1)/2$, then $\rho_{\mathrm{KZ}} \leq 1$, which by Theorem 12 implies that X contains an integer vector. If $F(\boldsymbol{c}) < c_0 \cdot n \cdot (n+1)/2$, then we need to branch. Due to the definition of $F(\boldsymbol{c})$ we know in this case that $w_{\boldsymbol{c}}(X) < c_0 \cdot n \cdot (n+1)/2$, which implies that the polytope X in the direction $\boldsymbol{c}$ is "thin". As in the previous subsection we create one subproblem for every hyperplane $\boldsymbol{c}^T\boldsymbol{x} = \mu, \ldots, \boldsymbol{c}^T\boldsymbol{x} = \mu + c_0 \cdot n \cdot (n+1)/2$, where $\mu = \lceil \min\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in X\} \rceil$. Once we have fixed a hyperplane $\boldsymbol{c}^T\boldsymbol{x} = t$, we have obtained a problem in dimension less than or equal to $n-1$, and we repeat the process. This procedure creates a search tree that is at most n deep, and that has a constant number of branches at each level when n is fixed. The algorithm called in each branch is, however, polynomial for fixed dimension only. First, the generalized basis reduction algorithm runs in polynomial time for fixed dimension, and second, computing the shortest vector $\boldsymbol{c}$ is done in polynomial time for fixed dimension. An alternative would be to use the first reduced basis vector with respect to $(X-X)^*$, instead of the shortest vector $\boldsymbol{c}$. According to Proposition 4, $F(\boldsymbol{b}_1) \leq (\frac{1}{2} - \epsilon)^{1-n}F(\boldsymbol{c})$. In this version of the algorithm we would first check whether $F(\boldsymbol{b}_1) \geq c_0 \cdot n \cdot (n+1)/(2(\frac{1}{2}-\epsilon)^{1-n})$. If yes, then X contains an integer vector, and if no, we need to branch, and we create at most $c_0 \cdot n \cdot (n+1)/(2(\frac{1}{2}-\epsilon)^{n-1})$ hyperplanes.

If the algorithm terminates with the result that X contains an integer vector, then Lovász and Scarf describe how such a vector can be constructed by using the Korkine-Zolotareff basis (see [79], proof of Theorem 10).

Lagarias, Lenstra, and Schnorr [73] derive bounds on the Euclidean length of Korkine-Zolotareff reduced basis vectors of a lattice and its dual lattice. The bounds are given in terms of the successive minima of L and the dual lattice L^*. Later, Kannan and Lovász [67], [68] introduced the generalized Korkine-Zolotareff basis, as defined above, and derived bounds of the same type as in the paper by Lagarias et al. These bounds were used to study covering minima of a convex set with respect to a lattice, such as the covering radius, and the lattice width. An important result by Kannan and Lovász is that the product of the first successive minima of the lattices L and L^* is bounded from above by $c_0 \cdot n$. This improves on a similar result of Lagarias et al. and implies Theorem 11 above. There are many interesting results on properties of various lattice constants. Many of them are described in the survey by Kannan [65], and will not be discussed further here.

Example 2. The following example demonstrates a few iterations with the generalized basis reduction algorithm. Consider the polytope $X = \{\boldsymbol{x} \in$

$\mathbb{R}^2_{\geq 0} \mid x_1 + 7x_2 \geq 7,\ 2x_1 + 7x_2 \leq 14,\ -5x_1 + 4x_2 \leq 4\}$. Let $j = 1$ and $\epsilon = \frac{1}{4}$. Assume we want to use the generalized basis reduction algorithm to find a direction in which the width of X is small. Recall that a lattice vector $\boldsymbol{c}$ that minimizes the width of X is a shortest lattice vector with respect to the set $(X - X)^*$. The first reduced basis vector is an approximation of the shortest vector for $(X - X)^*$ and hence an approximation of the thinnest direction for X. The distance functions associated with $(X - X)^*$ are

$$F_j(\boldsymbol{c}) = \max\{\boldsymbol{c}^T(\boldsymbol{x} - \boldsymbol{y}) \mid \boldsymbol{x} \in X,\ \boldsymbol{y} \in X,\ \boldsymbol{b}_i^T(\boldsymbol{x} - \boldsymbol{y}) = 0, 1 \leq i \leq j - 1\}.$$

The initial basis is

$$\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

We obtain $F_1(\boldsymbol{b}_1) = 7.0,\ F_1(\boldsymbol{b}_2) = 1.8,\ \mu = 0$, and $F_1(\boldsymbol{b}_2 + 0\boldsymbol{b}_1) = 1.8$, see Figure 6. Here we see that the number of lattice hyperplanes intersecting X in direction $\boldsymbol{b}_1$ is 8. The hyperplanes are $x_1 = 0,\ x_1 = 1, \ldots,\ x_1 = 7$. The number of hyperplanes intersecting X in direction $\boldsymbol{b}_2$ is 2: $x_2 = 0,\ x_2 = 1$.

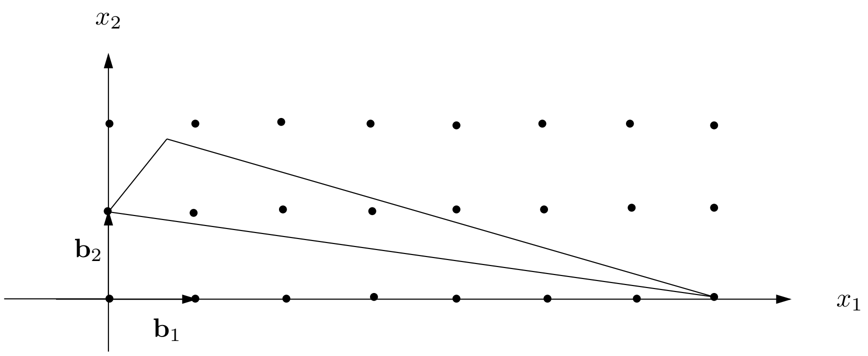


Figure 6: The unit vectors form the initial basis.

Checking Conditions (22) and (23) shows that Condition (22) is satisfied as $F_1(\boldsymbol{b}_2 + 0\boldsymbol{b}_1) \geq F_1(\boldsymbol{b}_2)$, but that Condition (23) is violated as $F_1(\boldsymbol{b}_2) \not\geq (3/4)F_1(\boldsymbol{b}_1)$, so we interchange $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ and remain at $j = 1$.

Now we have $j = 1$ and

$$\boldsymbol{b}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \boldsymbol{b}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

$F_1(\boldsymbol{b}_1) = 1.8,\ F_1(\boldsymbol{b}_2) = 7.0,\ \mu = 4$, and $F_1(\boldsymbol{b}_2 + 4\boldsymbol{b}_1) = 3.9$

Condition (22) is violated as $F_1(\boldsymbol{b}_2 + 4\boldsymbol{b}_1) \not\geq F_1(\boldsymbol{b}_2)$, so we replace $\boldsymbol{b}_2$ by $\boldsymbol{b}_2 + 4\boldsymbol{b}_1 = (1,4)^T$. Given the new basis vector $\boldsymbol{b}_2$ we check Condition (23) and we conclude that this condition is satisfied. Hence the basis

$$\boldsymbol{b}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \boldsymbol{b}_2 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

is Lovász-Scarf reduced, see Figure 7. In the root node of our search tree we would create two branches corresponding to the lattice hyperplanes $x_2 = 0$ and $x_2 = 1$.

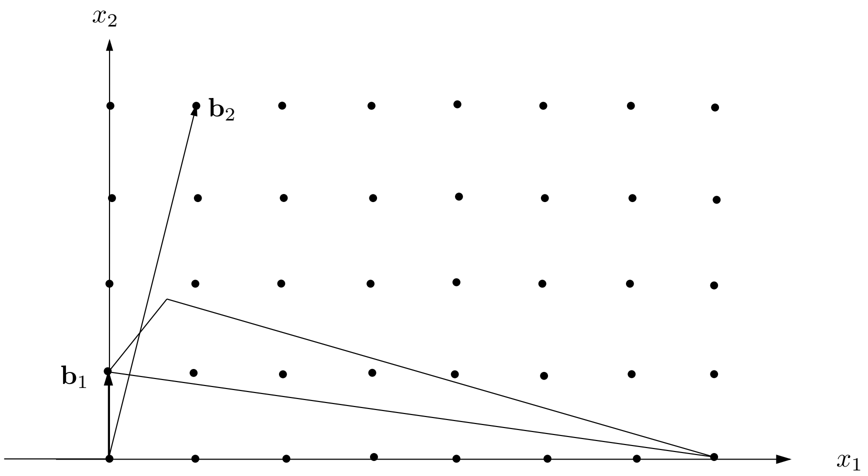


Figure 7: The reduced basis yields thin directions for the polytope.

□

4.4 Counting integer points in polytopes

Barvinok [12] showed that there exists a polynomial time algorithm for *counting* the number of integer points in a polytope if the dimension is fixed. Barvinok's result therefore generalizes the result of Lenstra [76]. Before Barvinok developed his counting algorithm, polynomial algorithms were only known for dimensions $n = 1, 2, 3, 4$. The cases $n = 1, 2$ are relatively simple, and for the challenging cases $n = 3, 4$, algorithms were developed by Dyer [37]. On the approximation side, Cook, Hartmann, Kannan, and McDiarmid [28] developed an algorithm that for a given rational number $\epsilon > 0$ counts the number of points in a polytope with a relative error less than ϵ in time polynomial in the input size and $1/\epsilon$.

Barvinok based his algorithm on an identity by Brion for exponential sums over polytopes. Later, Dyer and Kannan [38] developed a simplification of Barvinok's algorithm in which the step of the algorithm that uses the property that the exponential sum can be continued to define a meromorphic function over $\mathbb{C}^n$ (cf. Proposition 1) is unnecessary. In addition, Dyer and Kannan observed that Lenstra's algorithm is no longer needed as a subroutine of Barvinok's algorithm. See also the paper by Barvinok and Pommersheim [11] for a more elementary description of the algorithm. De Loera et al. [36] introduced further practical improvements over Dyer and Kannan's version, and implemented their version of the algorithm, which uses Lovász' basis reduction algorithm. De Loera et al. report on the first computational results from using an algorithm to count the number of lattice points in a polytope. These results are encouraging.

To describe Barvinok's algorithm in detail would require the introduction of quite a lot of new material, which would take us outside the scope of this chapter. The results is so important though that we still want to give a high-level presentation here.

Barvinok's algorithm counts integer points in an integer simplex; given $k+1$ integer vectors such that their convex hull is a k-dimensional simplex Δ, compute the number of integer points in Δ. Dyer [37] had previously shown that the problem of counting integer points in a polytope can be reduced to counting integer points in polynomially many integer simplices. See also Cook et al. [28], who proved that if P_I is the integer hull of the rational polyhedron $P \subset \mathbb{R}^n$ given by m inequalities whose size is at most φ, then for fixed n an upper bound on the number of vertices of P_I is $\mathrm{O}(m^n\varphi^{n-1})$.

The main tools of Barvinok's algorithm are *decompositions of rational cones in so-called primitive cones*, and *exponential sums over polytopes*. The decomposition of cones will be treated very briefly. For details we refer to Section 5 of Barvinok's paper. For an exponential sum over a polytope P we write

$$\sum_{\boldsymbol{x}\in(P\cap\mathbb{Z}^n)} \exp\{\boldsymbol{c}^T\boldsymbol{x}\}\,, \tag{39}$$

where P is a polytope in $\mathbb{R}^n$, and $\boldsymbol{c}$ is an n-dimensional real vector.

Before giving an outline of the algorithm we need to introduce new notation. A convex cone $K \in \mathbb{R}^n$ is *rational* if it is the conic hull of finitely many integer generators, i.e., $K = \mathrm{cone}\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k\},\ \boldsymbol{u}_i \in \mathbb{Z}^n,\ 1 \le i \le k$. A cone K is *simple* if it can be generated by linearly independent vectors. A simple rational cone K is *primitive* if $K = \mathrm{cone}\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k\}$, where $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$ form a basis of the lattice $\mathbb{Z}^n \cap \mathrm{lin}(K)$, where $\mathrm{lin}(K)$ is the linear hull of K. A *meromorphic function* $f(\boldsymbol{z})$ is a single-valued function that can be expressed as $f(\boldsymbol{z}) = g(\boldsymbol{z})/h(\boldsymbol{z})$, where $g(\boldsymbol{z})$ and $h(\boldsymbol{z})$ are functions that are analytic at all finite points of the complex plane $\mathbb{C}$. We can associate a meromorphic function with each rational cone.

Proposition 6. *Let K be a simple rational cone. Let $\boldsymbol{c} \in \mathbb{R}^n$ be a vector such that the inner product $(\boldsymbol{c}^T\cdot)$ decreases along the extreme rays of K. Then the series*

$$\sum_{\boldsymbol{x}\in(K\cap\mathbb{Z}^n)} \exp\{\boldsymbol{c}^T\boldsymbol{x}\}$$

converges and defines a meromorphic function in $\boldsymbol{c} \in \mathbb{C}^n$. This function is denoted by $\sigma(K;\boldsymbol{c})$. If $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k \in \mathbb{Z}^n$ are linearly independent generators of K, then for all $\boldsymbol{c} \in \mathbb{C}^n$ the following holds,

$$\sigma(K;\boldsymbol{c}) = p_K(\exp\{c_1\},\ldots,\exp\{c_n\}) \cdot \Pi_{i=1}^{k}\frac{1}{1-\exp\{\boldsymbol{c}^T\boldsymbol{u}_i\}}, \tag{40}$$

where p_K is a Laurent polynomial in n variables.

We observe that the set of singular points of $\sigma(K;\boldsymbol{c})$ is the set of hyperplanes $H_i = \{\boldsymbol{c} \in \mathbb{R}^n \mid \ \boldsymbol{c}^T\boldsymbol{u}_i = 0\}$, $1 \le i \le k$. The question now is how we can obtain an explicit expression for the number of points in a polytope from the result above. The key to such an expression is the following theorem by Brion.

Theorem 13 ([19]). *Let $P \subset \mathbb{R}^n$ be a rational polytope, and let V be the set of vertices of P. For each vertex $\boldsymbol{v} \in V$, the* supporting cone *$K_{\boldsymbol{v}}$ of P at $\boldsymbol{v}$ is defined as $K_{\boldsymbol{v}} = \{\boldsymbol{u} \in \mathbb{R}^n \mid \ \boldsymbol{v}+\delta\boldsymbol{u} \in P$ for all sufficiently small $\delta > 0\}$. Then*

$$\sum_{\boldsymbol{x}\in(P\cap\mathbb{Z}^n)} \exp\{\boldsymbol{c}^T\boldsymbol{x}\} = \sum_{\boldsymbol{v}\in V} \exp\{\boldsymbol{c}^T\boldsymbol{v}\}\cdot\sigma(K_{\boldsymbol{v}};\boldsymbol{c}) \tag{41}$$

for all $\boldsymbol{c} \in \mathbb{R}^n$ that are not singular points for any of the functions $\sigma(K_{\boldsymbol{v}};\boldsymbol{c})$.

Considering the left-hand side of expression (41), it seems tempting to use $\boldsymbol{c} = \boldsymbol{0}$ in expression (41) for $P = \Delta$, since this will contribute 1 to the sum from every integer point, but this is not possible since $\boldsymbol{0}$ is a singular point for the functions $\sigma(K_{\boldsymbol{v}};\boldsymbol{c})$. Instead we take a vector $\boldsymbol{c}$ that is regular for all of the functions $\sigma(K_{\boldsymbol{v}};\boldsymbol{c})$, $\boldsymbol{v} \in V$, and a parameter t, and we compute the constant term of the Taylor expansion of the function $\sum_{\boldsymbol{x}\in\Delta\cap\mathbb{Z}^n}\exp\{t\cdot(\boldsymbol{c}^T\boldsymbol{x})\}$ in the neighborhood of the point $t = 0$. Equivalently, due to Theorem 13 we can instead compute the constant terms of the Laurent expansions of the functions $\exp\{t \cdot (\boldsymbol{c}^T\boldsymbol{v})\} \cdot \sigma(K_{\boldsymbol{v}}; t \cdot \boldsymbol{c})$ for all vertices $\boldsymbol{v}$ of Δ. These constant terms are denoted by $R(K_{\boldsymbol{v}}, \boldsymbol{v}, \boldsymbol{c})$. In general there does not exist an explicit formula for $R(K_{\boldsymbol{v}}, \boldsymbol{v}, \boldsymbol{c})$, but if $K_{\boldsymbol{v}}$ is primitive, then such an explicit expression does exist, and is based on the fact that the function $\sigma(K; \boldsymbol{c})$ in expression (40) looks particularly simple if K is a primitive cone, namely, the polynomial p_K is equal to one.

Proposition 7. *Assume that $K \subset \mathbb{R}^n$ is a primitive cone with primitive generators $\{\boldsymbol{u}_1, \ldots \boldsymbol{u}_k\}$. Then*

$$\sigma(K; \boldsymbol{c}) = \Pi_{i=1}^{k} \frac{1}{1 - \exp\{\boldsymbol{c}^T\boldsymbol{u}_i\}} \,.$$

A simple rational cone can be expressed as an integer linear combination of primitive cones in polynomial time if the dimension n is fixed (see also Section 5 in [12]) as is stated in the following important theorem by Barvinok.

Theorem 14 ([12]). *Let us fix $n \in \mathbb{N}$. Then there exists a polynomial algorithm that for any given rational cone K constructs a family $K_i \subset \mathbb{R}^n$, $i \in I$ of rational primitive cones and computes integer numbers ϵ_i, $i \in I$ such that*

$$K = \sum_{i \in I} \epsilon_i K_i \text{ and } \sigma(K; c) = \sum_{i \in I} \epsilon_i \sigma(K_i; c) \tag{42}$$

for all $c \in \mathbb{R}^n$ that are regular points for the functions $\sigma(K; c)$, $\sigma(K_i; c)$, $i \in I$.

Notice that the numbers ϵ_i, $i \in I$, in expression (42) are either equal to $+1$ or -1.

Barvinok's decomposition of rational cones leads to a polynomial algorithm for fixed n for computing the constant term $R(K, \boldsymbol{v}, \boldsymbol{c})$ for an arbitrary rational cone K and an arbitrary vector $\boldsymbol{v}$. Lenstra's algorithm is used as a subroutine in the decomposition. As mentioned earlier, Lenstra's algorithm is not necessary in the algorithm presented by Dyer and Kannan.

The only component of the overall algorithm that we are missing is how to construct a generic vector $\boldsymbol{c}$ that is not a singular point for $\sigma(K_{\boldsymbol{v}}; \boldsymbol{c})$. This can be done in polynomial time as is stated in the following lemma.

Lemma 2 ([12]). *There exists a polynomial time algorithm that for any given $n \in \mathbb{N}$, for any given $m \in \mathbb{N}$, and for any rational vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m \in \mathbb{Q}^n$ constructs a rational vector $\boldsymbol{c}$ such that $\boldsymbol{c}^T\boldsymbol{u}_i \neq 0$ for $1 \leq i \leq m$.*

To summarize, a sketch of Barvinok's algorithm is as follows. First, for each vertex $\boldsymbol{v}$ of the simplex Δ, compute the integer generators of the supporting cone $K_{\boldsymbol{v}}$. Each cone $K_{\boldsymbol{v}}$ is then expressed as an integer linear combination of primitive cones K_i, i.e., $K_{\boldsymbol{v}} = \sum_{i \in I_v} \lambda_i K_i$ for integer λ_i. By using Lemma 2 we can now construct a vector $\boldsymbol{c}$ that is not orthogonal to any of the generators of the cones K_i, $i \in \cup_{\boldsymbol{v}} I_{\boldsymbol{v}}$, which means that $\boldsymbol{c}$ is not a singular point for the functions $\sigma(K_i; \boldsymbol{c})$. Next, for all $\boldsymbol{v}$ and $I_{\boldsymbol{v}}$ compute the constant term $R(K_i, \boldsymbol{v}, \boldsymbol{c})$ of the function $\exp\{t \cdot (\boldsymbol{c}^T \boldsymbol{v})\} \cdot \sigma(K_i; t \cdot \boldsymbol{c})$ as $t \to 0$. Let $\#(\Delta \cap \mathbb{Z}^n)$ denote the number of integer points in the simplex Δ. Through Brion's expression (41) we have now obtained

$$\#(\Delta \cap \mathbb{Z}^n) = \sum_{\boldsymbol{v} \in V} \sum_{i \in I_v} \lambda_i \cdot R(K_i, \boldsymbol{v}, \boldsymbol{c}) \,.$$

5 Algorithms for the integer optimization problem in fixed dimension

So far we have only dealt with the integer feasibility problem in fixed dimension n. We now come to algorithms that solve the *integer optimization problem* in fixed dimension. Here one is given an integer matrix $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ and integer vectors $\boldsymbol{d} \in \mathbb{Z}^m$ and $\boldsymbol{c} \in \mathbb{Z}^n$, where the dimension n is fixed. The task is to find an integer vector $\boldsymbol{x}^* \in \mathbb{Z}^n$ that satisfies $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{d}$, and that maximizes $\boldsymbol{c}^T \boldsymbol{x}$. Thus the integer feasibility problem is a subproblem of the integer optimization problem. Let φ be the maximum size of $\boldsymbol{c}$ and a constraint $\boldsymbol{a}_i \boldsymbol{x} \le d_i$ of $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{d}$. The running time of the methods described here will be estimated in terms of the number of constraints m and the number φ.

The integer optimization problem can be reduced to the integer feasibility problem (27) via binary search, see, e.g. [54, 99]. This approach yields a running time of $\mathrm{O}(m\,\varphi + \varphi^2)$, and is described in Section 5.1.

There have been many efficient algorithms for the 2-dimensional integer optimization problem. Feit [46], and Zamanskij and Cherkasskij [106] provided an algorithm for the 2-dimensional integer optimization problem that runs in $\mathrm{O}(m \ \log m + m\,\varphi)$ steps. Other algorithms are by Kanamaru et al. [62] ($\mathrm{O}(m \ \log m + \varphi)$), and by Eisenbrand and Rote [42] ($\mathrm{O}(m + (\log m)\varphi)$). Eisenbrand and Laue [41] recently provided a linear time algorithm ($\mathrm{O}(m + \varphi)$).

A randomized algorithm for arbitrary fixed dimension was proposed by Clarkson [25], which we present in Section 5.3. His result can be stated in the more general framework of *LP-type problems*. Applied to integer programming, the result is as follows. An integer optimization problem that is defined by m constraints can be solved with expected number of $\mathrm{O}(m)$ basic operations and $\mathrm{O}(\log m)$ calls to another algorithm that solves an integer optimization problem with a fixed number of constraints, see also [48]. In the description of Clarkson's algorithm here, we ignore the dependence of the running time on the dimension. Clarkson's algorithm has played an important role in the search for faster algorithms in varying dimension for *linear programming* in the ram-model of complexity. For more on this fascinating topic, see [80] and [48].

We also sketch a recent result by Eisenbrand [40] in Section 5.2, which shows that an integer optimization problem of binary encoding size φ with a fixed number of constraints can be solved with $\mathrm{O}(\varphi)$ arithmetic operations on rationals of size $\mathrm{O}(\varphi)$. Thus with Clarkson's result one obtains an expected running time of $\mathrm{O}(m + (\log m)\varphi)$ arithmetic operations for the integer optimization problem.

First we will transform the integer optimization problem into a more convenient form. If $\boldsymbol{U} \in \mathbb{Z}^{n\times n}$ is a unimodular matrix, then by substituting $\boldsymbol{y} = \boldsymbol{U}^{-1}\boldsymbol{x}$, the integer optimization problem above is the problem to find a vector $\boldsymbol{y}^* \in \mathbb{Z}^n$ that satisfies $\boldsymbol{A}\,\boldsymbol{U}\,\boldsymbol{y}^* \leq \boldsymbol{d}$ and maximizes $\boldsymbol{c}^T\boldsymbol{U}\boldsymbol{y}$. With a sequence of extended-greatest common divisor operations, one can compute a unimodular $\boldsymbol{U} \in \mathbb{Z}^{n\times n}$ of binary encoding length $\mathrm{O}(\varphi)$ (n is fixed) such that $\boldsymbol{c}^T\boldsymbol{U} = (\gcd(c_1,\ldots,c_n), 0\ldots,0)$. Therefore we can assume that the objective vector $\boldsymbol{c}$ is the first unit vector.

The algorithms for the integer feasibility problem (27), which we discussed in Section 4, require $\mathrm{O}(m + \varphi)$ arithmetic operations to be solved. This is linear in the input encoding. Therefore we can assume that the system $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ is integer feasible.

Now, there exists an optimal $\boldsymbol{x}^* \in \mathbb{Z}^n$ whose binary encoding length is $\mathrm{O}(\varphi)$, see, e.g. Schrijver [99, p. 239]. This means that we can assume that the constraints $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ describe a polytope. This polytope can be translated with an integer vector into the positive orthant.

Notice that the above described transformation can be carried out with $\mathrm{O}(m + \varphi)$ basic operations. Furthermore the number of constraints of the transformed system is $\mathrm{O}(m)$ and the binary encoding length of each constraint remains $\mathrm{O}(\varphi)$. Thus given $\boldsymbol{A}, \boldsymbol{d}$ and $\boldsymbol{c}$, we can in $\mathrm{O}(m + \varphi)$ steps check whether the system $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ is integer feasible and carry out the above described transformation. We therefore define the integer optimization problem as being the following:

Given an integer matrix $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$ and an integer vector $\boldsymbol{d} \in \mathbb{Z}^m$ defining a polytope $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$ such that $P \subseteq \mathbb{R}^n_{\geq 0}$ and $P \cap \mathbb{Z}^n \neq \emptyset$:

$$\text{Find an integer vector } \boldsymbol{x}^* \in \mathbb{Z}^n \text{, with maximal first component, satisfying } \boldsymbol{A}\boldsymbol{x}^* \leq \boldsymbol{d}. \tag{43}$$

5.1 Binary Search

We first describe and analyze the binary search technique for the integer optimization problem. As we argued, we can assume that $P \subseteq [0, M]^n$, where $M \in \mathbb{N}$ and that M is part of the input. In the course of binary search, one keeps two integers $l, u \in \mathbb{N}$ such that $l \leq x_1^* \leq u$. We start with $l = 0$ and $u = M$. In the i-th iteration, one checks whether the system $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$, $x_1 \geq \lfloor (l+u)/2 \rfloor$ is integer feasible. If it is feasible, then one sets $l = \lfloor (l+u)/2 \rfloor$. If the system is integer infeasible, one sets $u = \lfloor (l+u)/2 \rfloor$. After $\mathrm{O}(\mathrm{size}(M))$ steps one has either $l = u$ or $l + 1 = u$ and the optimum can be found with at most two more calls to an integer feasibility algorithm.

The binary encoding length of M is at most $\mathrm{O}(\varphi)$, see, e.g. [99, p. 120]. Therefore the integer optimization problem can be solved with $\mathrm{O}(\varphi)$ queries to an integer feasibility algorithm.

Theorem 15. *An integer optimization problem* (43) *in fixed dimension defined by m constraints, each of binary encoding length at most φ, can be solved with* $\mathrm{O}(m\,\varphi + \varphi^2)$ *basic operations on rational numbers of size* $\mathrm{O}(\varphi)$.

5.2 A linear algorithm

In this section, we outline a recent algorithm by Eisenbrand [40] that solves an integer optimization problem with a fixed number of constraints in linear time. Thus, the complexity of integer feasibility with a fixed number of variables and a fixed number of constraints can be matched with the complexity of the Euclidean algorithm in the arithmetic model.

As in the algorithms in Sections 4.2 and 4.3 one makes use of the lattice width concept, see Expression (28) and Theorem 9 in the introduction of Section 4.

The first step of the algorithm is to reduce the integer optimization problem over a full-dimensional polytope to a disjunction of integer optimization problems over *two-layer simplices*. A two layer simplex is a full-dimensional simplex, whose vertices can be partitioned into two sets V and W, such that the first components of the elements in each of the sets V and W agree, i.e., for all $\boldsymbol{v}_1, \boldsymbol{v}_2 \in V$ one has $v_1^1 = v_2^1$ and for all $\boldsymbol{w}_1, \boldsymbol{w}_2 \in W$ one has $w_1^1 = w_2^1$.

How can one reduce the integer optimization problem over a polytope P to a sequence of integer optimization problems over two-layer simplices? Simply consider the hyperplanes $x_1 = v_1$ for each vertex $\boldsymbol{v}$ of P. If the number of constraints defining P is fixed, then these hyperplanes partition P into a constant number of polytopes, whose vertices can be grouped into two groups, according to the value of their first component. Thus we can assume that the vertices of P itself can be partitioned into two sets V and W, such that the first components of the elements in each of the sets V and W agree. Carathéodory's theorem, see Schrijver [99, p. 94], implies that P is covered by the simplices that are spanned by the vertices of P. These simplices are two-layer simplices. Therefore, the integer optimization problem in fixed dimension with a fixed number of constraints can be reduced in constant time to a constant number of integer optimization problems over a two-layer simplex.

The key idea is then to let the objective function slide into the two-layer simplex, until the width of the truncated simplex exceeds the flatness bound. In this way, one can be sure that the optimum of the integer optimization problem lies in the truncation, which is still flat. Thereby one has reduced the *integer optimization problem* in dimension n to a constant number of *integer optimization problems* in dimension $n-1$ and binary search can be avoided.

How do we determine a parameter π such that the truncated two-layer simplex $\Sigma \cap (x_1 \geq \pi)$ just exceeds the flatness bound? We explain the idea with the help of the 3-dimensional example in Figure 8. Here we have a two-layer simplex Σ in 3-space. The set V consists of the points $\mathbf{0}$ and $\boldsymbol{v}_1$ and W consists of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. The picture on the left describes a particular point in time, where the objective function slid into Σ. So we consider the truncation $\Sigma \cap (x_1 \geq \pi)$ for some $\pi \geq w_1^1$. This truncation is the convex hull of the points

$$\mathbf{0}, \boldsymbol{v}_1, \mu \boldsymbol{w}_1, \mu \boldsymbol{w}_2, (1-\mu)\boldsymbol{v}_1 + \mu \boldsymbol{w}_1, (1-\mu)\boldsymbol{v}_1 + \mu \boldsymbol{w}_2, \tag{44}$$

where $\mu = \pi / w_1^1$. Now consider the simplex $\Sigma_{V,\mu W}$, which is spanned by the points $\mathbf{0}, \boldsymbol{v}_1, \mu \boldsymbol{w}_1, \mu \boldsymbol{w}_2$. This simplex is depicted on the right in Figure 8. If this simplex is scaled by 2, then it contains the truncation $\Sigma \cap (x_1 \geq \pi)$. This is easy to see, since the scaled simplex contains the points $2(1-\mu)\,\boldsymbol{v}_1$, $2\,\mu\,\boldsymbol{w}_1$ and $2\,\mu\,\boldsymbol{w}_2$. So we have the condition $\Sigma_{V,\mu W} \subseteq \Sigma \cap (x_1 \geq \pi) \subseteq 2\,\Sigma_{V,\mu W}$. From this we can infer the important observation

$$w(\Sigma_{V,\mu W}) \leq w(\Sigma \cap (\boldsymbol{x}(1) \geq \pi)) \leq 2\,w(\Sigma_{V,\mu W}). \tag{45}$$

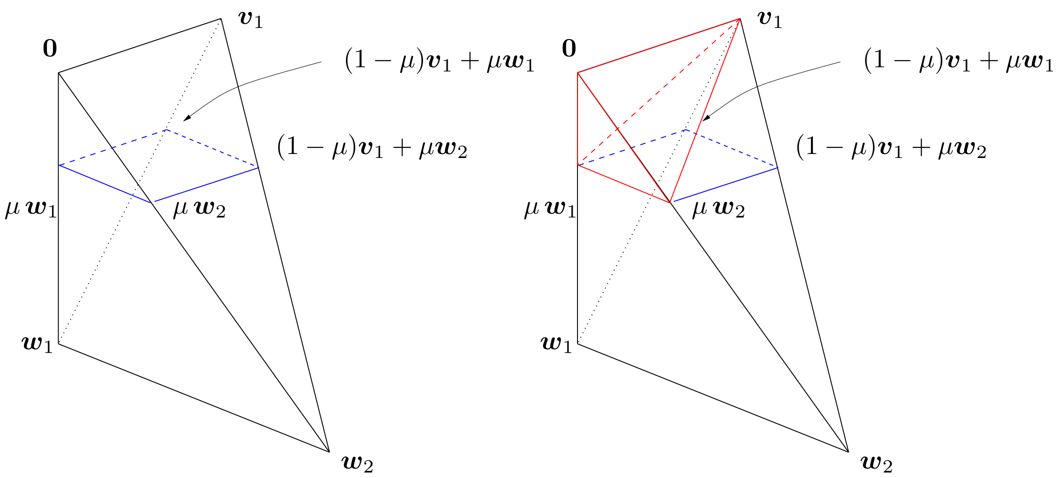


Figure 8: Solving the parametric lattice width problem.

This means that we essentially determine the correct π by determining a $\mu \geq 0$, such that the width of the simplex $\Sigma_{V,\mu W}$ just exceeds the flatness bound. The width of $\Sigma_{V,\mu W}$ is roughly (up to a constant factor) the length of the shortest vector of the lattice $L_\mu = L(\boldsymbol{A})$, where $\boldsymbol{A}$ is the matrix

$$A = \begin{pmatrix} \mu\boldsymbol{w}_1^T \\ \mu\boldsymbol{w}_2^T \\ \boldsymbol{v}_1 \end{pmatrix}.$$

Thus we have to find a parameter μ, such that the shortest vector of L_μ is sandwiched between $f(n)+1$ and $\gamma \cdot (f(n)+1)$ for some constant γ. This problem can be understood as a *parametric shortest vector* problem.

To describe this problem let us introduce some notation. We define for an $n \times n$-matrix $\boldsymbol{A} = (a_{ij})_{\forall i,j}$, the matrix $\boldsymbol{A}^{\mu,k} = (a_{ij})^{\mu,k}_{\forall i,j}$, as

$$a^{\mu,k}_{ij} = \begin{cases} \mu \cdot a_{ij}, & \text{if } i \leq k, \\ a_{ij}, & \text{otherwise.} \end{cases} \tag{46}$$

In other words, the matrix $\boldsymbol{A}^{\mu,k}$ results from $\boldsymbol{A}$ by scaling the first k rows with μ. The parametric shortest vector problem is now defined as follows.

> Given a nonsingular matrix $\boldsymbol{A} \in \mathbb{Z}^{n\times n}$ and some $U \in \mathbb{N}$, find a parameter $p \in \mathbb{N}$ such that $U \leq \mathrm{SV}(L(\boldsymbol{A}^{p,k})) \leq 2^{n+1/2} \cdot U$ or assert that $\mathrm{SV}(L) > U$.

It turns out that the parametric shortest vector problem can be solved in linear time when the dimension is fixed. From this, it follows that the integer optimization problem in fixed dimension with a fixed number of constraints can be solved in linear time.

Theorem 16 ([40]). *An integer program of binary encoding length φ in fixed dimension which is defined by a fixed number of constraints can be solved with* $\mathrm{O}(\varphi)$ *arithmetic operations on rational numbers of binary encoding length* $\mathrm{O}(\varphi)$*.*

5.3 Clarkson's random sampling algorithm

Clarkson [25] presented a *randomized algorithm* for problems of *linear programming type*. This algorithm solves an integer optimization problem that is defined by m constraints with an expected number of $\mathrm{O}(m)$ basic arithmetic operations and $\mathrm{O}(\log m)$ calls to an algorithm that solves an integer optimization problem defined by a fixed-size subset of the constraints. The expected running time of this method for an integer optimization problem defined by m constraints, each of size at most φ, can thus be bounded by $\mathrm{O}(m + (\log m)\varphi)$ arithmetic operations on rationals of size $\mathrm{O}(\varphi)$.

Let P be the polytope defined by $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}, 0 \leq x_j \leq M, \ 1 \leq j \leq n\}$. The integer vectors $\widetilde{\boldsymbol{x}} \in \mathbb{Z}^n \cap P$ satisfy $0 \leq \widetilde{x_j} \leq M$ for $1 \leq j \leq n$, where M is an integer of binary encoding length $\mathrm{O}(\varphi)$. A feasible integer point $\widetilde{\boldsymbol{x}}$ is optimal with respect to the objective vector $\boldsymbol{c} = ((M+1)^{n-1}, (M+1)^{n-2}, \ldots, (M+1)^0)^T$ if and only if it has maximal first component. Observe that the binary encoding length of this perturbed objective function vector $\boldsymbol{c}$ is is $\mathrm{O}(\varphi)$. Moreover, for each pair of distinct points $\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2 \in [0, M]^n \cap \mathbb{Z}^n$, $\widetilde{\boldsymbol{x}}_1 \neq \widetilde{\boldsymbol{x}}_2$, we have $\boldsymbol{c}^T\widetilde{\boldsymbol{x}}_1 \neq \boldsymbol{c}^T\widetilde{\boldsymbol{x}}_2$.

In the sequel we use the following notation. If H is a set of linear integer constraints, then the *integer optimum* defined by H is the *unique* integer point $\boldsymbol{x}^*(H) \in \mathbb{Z}^n \cap [0, M]^n$ which satisfies all constraints $h \in H$ and maximizes $\boldsymbol{c}^T\boldsymbol{x}$. Observe that, due to the perturbed objective function $\boldsymbol{c}^T\boldsymbol{x}$, the point $\boldsymbol{x}^*(H)$ is uniquely defined for any set of constraints H. The integer optimization problem now reads as follows:

$$\text{Given a set } H \text{ of integer constraints, find } \boldsymbol{x}^*(H). \tag{47}$$

A *basis* of a set of constraints H, is a minimal subset B of H such that $\boldsymbol{x}^*(B) = \boldsymbol{x}^*(H)$. The following is a consequence of a theorem of Bell [13] and Scarf [89], see Schrijver [99, p. 234].

Theorem 17. *Any set H of constraints in dimension n has a basis B of cardinality $|B| \leq 2^n - 1$.*

In the following we use the letter D for the number $2^n - 1$. Clarksons algorithm works for many *LP-type* problems, see Gärtner and Welzl [48] for more examples. The maximal cardinality of a basis is generally referred to as the *combinatorial dimension* of the LP-type problem.

Now we are ready to describe the algorithm. It comes in two layers that we call *Clarkson 1* and *Clarkson 2* respectively. The input of both algorithms is a set of constraints H and the output is $\boldsymbol{x}^*(H)$. The algorithm *Clarkson 1* keeps a constraint set G, which is initially empty and grows in the course of the algorithm. In one iteration, one draws a subset $R \subseteq H$ of cardinality $|R| = \lceil D\sqrt{m}\,\rceil$ at random and computes the optimum $\boldsymbol{x}^*(G \cup R)$ with the algorithm *Clarkson 2* described later. Now one identifies the constraints $V \subseteq H$ that are violated by $\boldsymbol{x}^*(G \cup R)$. We will prove below that the expected cardinality of V is $\sqrt{m}$. In Step (2c), the constraints V are added to the set G, if the cardinality of V does not exceed twice its expected cardinality. In this case, i.e., if $|V| \leq 2\sqrt{m}$, then an iteration of the `REPEAT`-loop is called *successful*.

Algorithm 3 (Clarkson 1).

1. $r \leftarrow \lceil D\sqrt{m}\,\rceil$, $G \leftarrow \emptyset$
2. `REPEAT`
   (a) Choose random $R \in \binom{H}{r}$
   (b) Compute $\boldsymbol{x}^* = \boldsymbol{x}^*(G \cup R)$ *with Clarkson 2*
   (c) $V \leftarrow \{h \in H \mid \boldsymbol{x}^* \text{ violates } h\}$
   (d) `IF` $|V| \leq 2\sqrt{m}$, `THEN` $G \leftarrow G \cup V$
3. `UNTIL` $V = \emptyset$
4. `RETURN` $\boldsymbol{x}^*$

How many expected iterations will *Clarkson 1* perform? To analyze this, let $B \subseteq H$ be a basis of H. Observe that, if the set V, which is computed in Step (2c), is nonempty, then there must be a constraint $b \in B$ that also belongs to V. Because, if no constraint in B is violated by $\boldsymbol{x}^*(G \cup R)$, then one has $\boldsymbol{x}^*(G \cup R) = \boldsymbol{x}^*(G \cup R \cup B) = \boldsymbol{x}^*(H)$ and V must be empty. Thus at each successful iteration, at least one new element of B enters the set G. We conclude that the number of successful iterations is bounded by D. The *Markov inequality*, see, e.g. Motwani and Raghavan [84] says that the probability that a random variable exceeds k-times its expected value is bounded by $1/k$. Therefore the expected number of iterations of the `REPEAT`-loop is bounded by $2\,D$. The additional arithmetic operations of each iteration is $\mathrm{O}(m)$ if n is fixed, and each iteration requires the solution of an integer optimization problem in fixed dimension with $\mathrm{O}(\sqrt{m})$ constraints.

Theorem 18 ([25]). *Given a set H of m integer linear constraints in fixed dimension, the algorithm* Clarkson 1 *computes $\boldsymbol{x}^*(H)$ with a constant number of expected calls to an algorithm which solves the integer optimization problem for a subset of* $\mathrm{O}(\sqrt{m})$ *constraints and an expected number of* $\mathrm{O}(m)$ *basic operations.*

We still need to prove that the expected cardinality of V in Step (2c) is at most $\sqrt{m}$. Following the exposition of Gärtner and Welzl [48], we do this in the slightly more general setting where H can be a multiset of constraints.

Lemma 3 ([25, 48]). *Let G be a set of integer linear constraints and let H be a multiset of m integer constraints in dimension n. Let $R \in \binom{H}{r}$ be a random subset of H of cardinality r. The expected cardinality of the set $V_R = \{h \in H \mid \boldsymbol{x}^*(G \cup R)$ violates $h\}$ is at most $D(m-r)/(r+1)$.*

This lemma establishes our desired bound on the cardinality of V in Step 2c, because there we have $r = \lceil D\sqrt{m}\,\rceil$ and thus

$$D(m-r)/(r+1) \leq Dm/r \leq \sqrt{m}. \tag{48}$$

Proof of Lemma 3. The expected cardinality of V_R is equal to the sum of all the cardinalities of V_R, where R is an r-element subset of H, divided by the number of ways that r elements can be drawn from H,

$$E(|V_R|) = \left(\sum_{R \in \binom{H}{r}} |V_R| \right) \Big/ \binom{m}{r}. \tag{49}$$

Let $\chi_G(Q,h)$, $Q \subseteq H$, $h \in H$ be the characteristic function for the event that $\boldsymbol{x}^*(G \cup Q)$ violates h. Thus

$$\chi_G(Q,h) = \begin{cases} 1 & \text{if } \boldsymbol{x}^*(G \cup Q) \text{ violates } h, \\ 0 & \text{otherwise.} \end{cases} \tag{50}$$

With this we can write

$$\binom{m}{r} E(|V_R|) = \sum_{R \in \binom{H}{r}} \sum_{h \in H \setminus R} \chi_G(R, h) \tag{51}$$

$$= \sum_{Q \in \binom{H}{r+1}} \sum_{h \in Q} \chi_G(Q - h, h) \tag{52}$$

$$\leq \sum_{Q \in \binom{H}{r+1}} D \tag{53}$$

$$= \binom{m}{r+1} D. \tag{54}$$

From (51) to (52) we used the fact that the ways in which we can choose a set R of cardinality r from H and then a constraint h from $H \setminus R$ are exactly the ways in which we can choose a set Q of cardinality $r+1$ from H and then one constraint h from Q. To justify the step from (52) to (53), consider a basis B_Q of $Q \cup G$. If h is not from the basis B_Q, then $\boldsymbol{x}^*(G \cup Q) = \boldsymbol{x}^*(G \cup (Q \setminus \{h\}))$. Therefore $\sum_{h \in Q} \chi_G(Q - h, h) \leq D$. □

The algorithm *Clarkson 2* proceeds from another direction. Instead of randomly sampling large sets of constraints and augmenting a set of constraints G, one at the time, a set R of cardinality $6\,D^2$ is drawn and the optimum $\boldsymbol{x}^*(R)$ is determined in each iteration with the algorithm outlined in Section 5.2. As in *Clarkson 1* one determines the constraints $V = \{h \in H \mid \boldsymbol{x}^*(R) \text{ violates } h\}$. If this set is nonempty, then there must be constraints of a basis B of H that are in V. One then doubles the probability of each constraint $h \in V$ to be drawn in the next round. This procedure is repeated until $V = \emptyset$. Instead of explicitly speaking about probabilities of a constraint $h \in H$, we follow again the exposition of Gärtner and Welzl [48], who assign a *multiplicity* $\mu(h) \in \mathbb{N}$ to each constraint of H. In this way one can think of H as being a multiset and apply Lemma 3 in the analysis. Let $Q \subseteq H$ be a subset of the constraints, then $\mu(Q)$ denotes the sum of the multiplicities of Q, $\mu(Q) = \Sigma_{h \in Q} \mu(h)$. In the beginning $\mu(h) = 1$ for each $h \in H$.

Algorithm 4 (Clarkson 2).

1. $r \leftarrow 6D^2$
2. `REPEAT:`
   (a) Choose random $R \in \binom{H}{r}$

(b) Compute $\boldsymbol{x}^* = \boldsymbol{x}^*(R)$

(c) $V \leftarrow \{h \in H \mid \boldsymbol{x}^* \text{ violates } h\}$

(d) `IF` $\mu(V) \leq \mu(H)/(3D)$ `THEN` for all $h \in V$ do $\mu_h \leftarrow 2\,\mu_h$

3. `UNTIL` $V = \emptyset$

4. `RETURN` $\boldsymbol{x}^*$

An iteration through the `REPEAT`-loop is called a *successful iteration*, if the condition in the `IF`-statement in Step (2d) is true. Using Lemma 3 the expected cardinality of V (as a multiset) is at most $\mu(H)/(6D)$. Again with the Markov inequality, the expected number of total iterations is at most twice the number of the successful iterations of the algorithm.

Let $B \subseteq H$ be a basis of H. In each successful iteration, the multiplicity of at least one element of B is doubled. Since $|B| \leq D$, the multiplicity of at least one element of B will be at least 2^k after $k\,D$ successful iterations. Therefore one has $2^k \leq \mu(B)$ after $k\,D$ successful iterations.

The number $\mu(B)$ is bounded by $\mu(H)$. In the beginning $\mu(H) = m$. After Step (2d) one has $\mu(H) := \mu(H) + \mu(V) \leq \mu(H)(1 + 1/(3\,D))$. Thus after $k\,D$ successful iterations one has $\mu(B) \leq m\,(1 + 1/(3D))^{k\,D}$. Using the inequality $e^t \geq (1+t)$ for $t \geq 0$, we obtain the following lemma on the number of successful iterations.

Lemma 4. *Let B be a basis of H and suppose that H has at least $6\,D^2$ elements. After $k\,D$ successful iterations of* Clarkson 2 *one has*

$$2^k \leq \mu(B) \leq m\,e^{k/3}.$$

This implies that the number of successful iterations is bounded by $\mathrm{O}(\log m)$. The expected number of iterations is therefore also $\mathrm{O}(\log m)$. In each iteration, one computes one integer optimization problem with a fixed number of constraints. If φ is the maximal binary encoding length of a constraint in H, this costs $\mathrm{O}(\varphi)$ basic operations with the linear algorithm of Section 5.2. Then one has to check each constraint in H, whether it is violated by $\boldsymbol{x}^*(R)$. This costs $\mathrm{O}(m)$ arithmetic operations. Altogether we obtain the following running time.

Lemma 5 ([25]). *Let H be a set of integer linear constraints and let φ be the maximal binary encoding length of a constraint $h \in H$. Then the integer optimization problem* (47) *can be solved with the randomized algorithm* Clarkson 2 *with an expected number of* $\mathrm{O}(m \log m + (\log m)\varphi)$ *basic operations.*

Now we estimate the running time of *Clarkson 1* where we plug in the running time bound for Step (2b). We obtain a expected constant number of calls to *Clarkson 2* on $\mathrm{O}(\sqrt{m})$ constraints and an additional cost of $\mathrm{O}(m)$ basic operations for the other steps. Thus we have a total amount of $\mathrm{O}(m + \sqrt{m}\log\sqrt{m} + (\log\sqrt{m})\varphi) = \mathrm{O}(m + (\log m)\varphi)$ basic operations.

Theorem 19 ([25]). *Let H be a set of integer linear constraints and let φ be the maximal binary encoding length of a constraint $h \in H$. Then the integer optimization problem* (43) *can be solved with a randomized algorithm with an expected number of* $\mathrm{O}(m + (\log m)\varphi)$ *basic operations.*

6 Using lattices to reformulate the problem

Here we will study some special types of integer feasibility problems that have been successfully solved by the following approach. Create a lattice L such that we can say that feasible solutions to our problem are short vectors in L. Once we have L, we write down an initial basis $\boldsymbol{B}$ for L, we then apply basis reduction to $\boldsymbol{B}$, which produces $\boldsymbol{B}'$. The columns of $\boldsymbol{B}'$ are relatively short and some might be feasible for our problem. If not, do a search for a feasible solution, or prove than none exists.

In Section 6.1 we present results for subset sum problems arising in knapsack cryptosystems. In cryptography, researchers have made extensive use of lattices and basis reduction algorithms to *break* cryptosystems; their computational experiments were among the first to establish the practical effectiveness of basis reduction algorithms. On the "constructive side" recent complexity results on lattice problems have also inspired researchers to *develop* cryptographic schemes based on the hardness of certain lattice problems. Even though cryptography is not within the central scope of this chapter, and even though knapsack cryptosystems have long been broken, we still wish to present the main result by Lagarias and Odlyzko [74], since it illustrates a nice application of lattice basis reduction, and since it has inspired the work on integer programming presented in Section 6.2. There, we will see how systems of linear diophantine equations with lower and upper bounds on the variables can be solved by similar techniques.

For comprehensive surveys on the topic of lattices in cryptography we refer to the surveys of Joux and Stern [61], and of Nguyen and Stern [86, 87].

6.1 Cryptosystems – solving subset sum problems

A sender wants to transmit a message to a receiver. The plaintext message of the sender consists of a 0-1 vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, and this message is encrypted by using integer weights $a_1, \ldots, a_n$ leading to an encrypted message $a_0 = \sum_{j=1}^{n} a_j x_j$. The coefficients a_j, $1 \le j \le n$, are known to the public, but there is a hidden structure in the relation between these coefficients, called a trapdoor, which only the receiver knows. If the trapdoor is known, then the subset sum problem:

$$\text{Determine a 0-1 vector } \boldsymbol{x} \text{ such that } \sum_{j=1}^{n} a_j x_j = a_0 \tag{55}$$

can be solved easily. For an eavesdropper who does not know the trapdoor, however, the subset sum problem should be hard to solve in order to obtain a secure transmission.

The *density* of a set of coefficients a_j, $1 \le j \le n$ is defined as

$$\delta(\boldsymbol{a}) = d(\{a_1, \ldots, a_n\}) = \frac{n}{\log_2(\max_{1 \le j \le n}\{a_j\})}.$$

The density, as defined above, is an approximation of the information rate at which bits are transmitted. The interesting case is $\delta(\boldsymbol{a}) \le 1$, since for $\delta(\boldsymbol{a}) > 1$ the subset sum problem (55) will in general have several solutions, which makes it unsuitable for generating encrypted messages. Lagarias and Odlyzko [74] proposed an algorithm based on basis reduction that often finds a solution to the subset sum problem (55) for instances having relatively low density. Earlier research had found methods based on recovering trapdoor information. If the information rate is high, i.e., $\delta(\boldsymbol{a})$ is high, then the trapdoor information is relatively hard to conceal. The result of Lagarias and Odlyzko therefore complements the earlier results by providing a method that is successful for low-density instances. In their algorithm Lagarias and Odlyzko consider a lattice in $\mathbb{Z}^{n+1}$ consisting of vectors of the following form:

$$L_{\boldsymbol{a},a_0} = \{(x_1, \ldots, x_n,\ (\boldsymbol{a}\boldsymbol{x} - a_0\xi))^T\} \tag{56}$$

where ξ is a variable associated with the right-hand side of $\boldsymbol{a}\boldsymbol{x} = a_0$. Notice that the lattice vectors that are interesting for the subset sum problem all have $\xi = 1$ and $\boldsymbol{a}\boldsymbol{x} - a_0\xi = 0$. It is easy to write down an initial basis $\boldsymbol{B}$ for $L_{\boldsymbol{a},a_0}$:

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{I}^{(n)} & \boldsymbol{0}^{(n\times 1)} \\ \boldsymbol{a} & -a_0 \end{pmatrix}. \tag{57}$$

To see that $\boldsymbol{B}$ is a basis for $L_{\boldsymbol{a},a_0}$, we note that taking integer linear combinations of the column vectors of $\boldsymbol{B}$ generates vectors of type (56). Let $\boldsymbol{x} \in \mathbb{Z}^n$ and $\xi \in \mathbb{Z}$. We obtain

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{a}\boldsymbol{x} - a_0\xi \end{pmatrix} = \boldsymbol{B} \begin{pmatrix} \boldsymbol{x} \\ \xi \end{pmatrix}.$$

The algorithm SV (Short Vector) by Lagarias and Odlyzko consists of the following steps.

1. Apply Lovász' basis reduction algorithm to the basis $\boldsymbol{B}$ (57), which yields a reduced basis $\widetilde{\boldsymbol{B}}$.
2. Check if any of the columns $\widetilde{\boldsymbol{b}}_k = (\widetilde{b}_k^1, \ldots, \widetilde{b}_k^{n+1})$ has all $\widetilde{b}_k^j = 0$ or γ for some fixed constant γ, for $1 \le j \le n$. If such a reduced basis vector is found, check if the vector $x_j = \widetilde{b}_k^j/\gamma$, $1 \le j \le n$, is a solution to $\sum_{j=1}^n a_j x_j = a_0$, and if yes, stop. Otherwise go to Step 3.
3. Repeat Steps 1 and 2 for the basis $\boldsymbol{B}$ with $a_0 = \sum_{j=1}^n a_j - a_0$, which corresponds to complementing all x_j-variables, i.e., considering $1 - x_j$ instead of x_j.

Algorithm SV runs in polynomial time as Lovász' basis reduction algorithm runs in polynomial time. It is not certain, however, that algorithm SV actually produces a solution to the subset sum problem. As Theorem 20 below shows, however, we can expect algorithm SV to work well on instances of (55) having low density. Consider a 0-1 vector $\boldsymbol{x}$, which we will consider as fixed. We assume that $\sum_{j=1}^n x_j \le \frac{n}{2}$. The reason for this assumption is that either $\sum_{j=1}^n x_j \le \frac{n}{2}$, or $\sum_{j=1}^n x'_j \le \frac{n}{2}$, where $x'_j = (1 - x_j)$, and since algorithm SV is run for both cases, one can perform the analysis for the vector that does satisfy the assumption. Let $\bar{\boldsymbol{x}} = (x_1, \ldots, x_n, 0)$. Let the sample space $\Lambda(A, \bar{\boldsymbol{x}})$ of lattices be defined to consist of all lattices $L_{\boldsymbol{a},a_0}$ generated by the basis (57) such that

$$1 \le a_j \le A, \quad \text{for } 1 \le j \le n, \tag{58}$$

and

$$a_0 = \sum_{j=1}^n a_j \bar{x}_j.$$

There is precisely one lattice in the sample space for each vector a satisfying (58). Therefore the sample space consists of A^n lattices.

Theorem 20 ([74]). *Let $\bar{\boldsymbol{x}}$ be a 0-1 vector for which $\sum_{j=1}^{n} \bar{x}_j \leq \frac{n}{2}$. If $A = 2^{\beta n}$ for any constant $\beta > 1.54725$, then the number of lattices $L_{\boldsymbol{a},a_0}$ in $\Lambda(A, \bar{\boldsymbol{x}})$ that contain a vector $\boldsymbol{v}$ such that $\boldsymbol{v} \neq k\bar{\boldsymbol{x}}$ for all $k \in \mathbb{Z}$, and such that $||\boldsymbol{v}||^2 \leq \frac{n}{2}$ is*

$$O(A^{n-c_1(\beta)}(\log A)^2), \tag{59}$$

where $c_1(\beta) = 1 - \frac{1.54725}{\beta} > 0$.

For $A = 2^{\beta n}$, the density of the subset sum problems associated with the lattices in the sample space can be proved to be equal to β^{-1}. This implies that Theorem 20 applies to lattices having density $\delta(\boldsymbol{a}) < (1.54725)^{-1} \approx 0.6464$. Expression (59) gives a bound on the number of lattices we need to subtract from the total number of lattices in the sample space, A^n, in order to obtain the number of lattices in $\Lambda(A, \bar{\boldsymbol{x}})$ for which $\bar{\boldsymbol{x}}$ is the *shortest* non-zero vector. Here we notice that the term (59) grows slower than the term A^n as n goes to infinity, and hence we can conclude that "almost all" lattices in the sample space $\Lambda(A, \bar{\boldsymbol{x}})$ have $\bar{\boldsymbol{x}}$ as the shortest vector. So, the subset sum problems (55) with density $\delta(\boldsymbol{a}) < 0.6464$ could be solved in polynomial time if we had an oracle that could compute the shortest vector in the lattice $L_{\boldsymbol{a},a_0}$. Lagarias and Odlyzko also prove that the algorithm SV actually finds a solution to "almost all" feasible subset sum problems (55) having density $\delta(\boldsymbol{a}) < (2 - \epsilon)(\log(\frac{4}{3}))^{-1}n^{-1}$ for any fixed $\epsilon > 0$.

Coster, Joux, LaMacchia, Odlyzko, Schnorr, and Stern [34] proposed two ways of improving Theorem 20. They showed that "almost all" subset sum problems (55) having density $\delta(\boldsymbol{a}) < 0.9408$ can be solved in polynomial time in presence of an oracle that finds the shortest vector in certain lattices. Both ways of improving the bound on the density involve some changes in the lattice considered by Lagarias and Odlyzko. The first lattice $L'_{\boldsymbol{a},a_0} \in \mathbb{Q}^{n+1}$ considered by Coster et al. is defined as

$$L'_{\boldsymbol{a},a_0} = \{(x_1 - \frac{1}{2}\xi, \ldots, x_n - \frac{1}{2}\xi,\ N(\boldsymbol{a}\boldsymbol{x} - a_0\xi))^T\},$$

where N is a natural number. The following basis $\bar{\boldsymbol{B}}$ spans $L'_{\boldsymbol{a},a_0}$:

$$\bar{\boldsymbol{B}} = \begin{pmatrix} \boldsymbol{I}^{(n)} & (-\boldsymbol{\frac{1}{2}})^{(n\times 1)} \\ N\boldsymbol{a} & -Na_0 \end{pmatrix}. \tag{60}$$

As in the analysis by Lagarias and Odlyzko, we consider a fixed vector $\boldsymbol{x} \in \{0,1\}^n$, and we let $\bar{\boldsymbol{x}} = (x_1, \ldots, x_n, 0)$. The vector $\bar{\boldsymbol{x}}$ does not belong to the lattice $L'_{\boldsymbol{a},a_0}$, but the vector $\boldsymbol{w} = (w_1, \ldots, w_n, 0)$, where $w_j = x_j -$

$\frac{1}{2}$, $1 \leq j \leq n$ does. So, if Lovász' basis reduction algorithm is applied to $\bar{\boldsymbol{B}}$ and if the reduced basis $\bar{\boldsymbol{B}}'$ contains a vector $(w_1, \ldots, w_n, 0)$ with $w_j = \{-\frac{1}{2}, \frac{1}{2}\}$, $1 \leq j \leq n$, then the vector $(w_j + \frac{1}{2})$, $1 \leq j \leq n$ solves the subset sum problem (55). By shifting the feasible region to be symmetric about the origin we now look for vectors of shorter Euclidean length. Coster et al. prove the following theorem that is analogous to Theorem 20.

Theorem 21 ([34]). *Let A be a natural number, and let $a_1, \ldots, a_n$ be random integers such that $1 \leq a_j \leq A$, for $1 \leq j \leq n$. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$, $x_j \in \{0, 1\}$, be fixed, and let $a_0 = \sum_{j=1}^{n} a_j x_j$. If the density $\delta(\boldsymbol{a}) < 0.9408$, then the subset sum problem (55) defined by $a_1, \ldots, a_n$ can "almost always" be solved in polynomial time by a single call to an oracle that finds the shortest vector in the lattice $L'_{\boldsymbol{a},a_0}$.*

Coster et al. prove Theorem 21 by showing that the probability that the lattice $L'_{\boldsymbol{a},a_0}$ contains a vector $\boldsymbol{v} = (v_1, \ldots, v_{n+1})$ satisfying

$$\boldsymbol{v} \neq k\boldsymbol{w} \ \text{ for all } k \in \mathbb{Z}, \text{ and } ||\boldsymbol{v}||^2 \leq ||\boldsymbol{w}||^2$$

is bounded by

$$n(4n\sqrt{n} + 1)\frac{2^{c_0 n}}{A} \tag{61}$$

for $c_0 = 1.0628$. Using the lattice $L'_{\boldsymbol{a},a_0}$, note that $||\boldsymbol{w}||^2 \leq \frac{n}{4}$. The number N in basis (60) is used in the following sense. Any vector in the lattice L' is an integer linear combination of the basis vectors. Hence, the $(n+1)$-st element of a such a lattice vector is an integer multiple of N. If N is chosen large enough, then a lattice vector can be "short" only if the $(n+1)$-st element is equal to zero. Since it is known that the length of $\boldsymbol{w}$ is bounded by $\frac{1}{2}\sqrt{n}$, then it suffices to choose $N > \frac{1}{2}\sqrt{n}$ in order to conclude that for a vector $\boldsymbol{v}$ to be shorter than $\boldsymbol{w}$ it should satisfy $v_{n+1} = 0$. Hence, Coster et al. only need to consider lattice vectors v in their proof that satisfy $\boldsymbol{v}_{n+1} = 0$. In the theorem we assume that the density $\delta(\boldsymbol{a})$ of the subset sum problems is less than 0.9408. Using the definition of $\delta(\boldsymbol{a})$ we obtain $\delta(\boldsymbol{a}) = n/\log_2(\max_{1\leq j\leq n}\{a_j\}) < 0.9408$, which implies that $\max_{1\leq j\leq n}\{a_j\} > 2^{n/0.9408}$, giving $A > 2^{c_0 n}$. For $A > 2^{c_0 n}$, the bound (61) goes to zero as n goes to infinity, which shows that "almost all" subset sum problems having density $\delta(\boldsymbol{a}) < 0.9408$ can be solved in polynomial time given the existence of a shortest vector oracle. Coster et al. also gave another lattice $L''_{\boldsymbol{a},a_0} \in \mathbb{Z}^{n+2}$ that could be used to obtain the result given in Theorem 21. The lattice $L''_{\boldsymbol{a},a_0}$ consists of vectors

$$L''_{\boldsymbol{a},a_0} = \begin{pmatrix} (n+1)x_1 - \sum_{\substack{k=1 \\ k \neq 1}}^n x_k - \xi \\ \vdots \\ (n+1)x_n - \sum_{\substack{k=1 \\ k \neq n}}^n x_k - \xi \\ (n+1)\xi - \sum_{j=1}^n x_j \\ N(\boldsymbol{ax} - a_0\xi) \end{pmatrix}$$

and is spanned by the basis

$$\begin{pmatrix} (n+1) & -1 & -1 & \cdots & -1 \\ -1 & (n+1) & -1 & \cdots & -1 \\ \vdots & & \ddots & & \vdots \\ -1 & \cdots & -1 & (n+1) & -1 \\ -1 & \cdots & \cdots & -1 & (n+1) \\ Na_1 & Na_2 & \cdots & Na_n & -Na_0 \end{pmatrix}. \tag{62}$$

Note that the lattice $L''_{\boldsymbol{a},a_0}$ is not full dimensional as the basis consists of $n+1$ vectors. Given a reduced basis vector $\boldsymbol{b} = (b_1, \ldots, b_{n+1}, 0)$, we solve the system of equations

$$\begin{aligned} b_j &= (n+1)x_j - \sum_{\substack{k=1 \\ k \neq j}}^n x_k - \xi, \ 1 \leq j \leq n, \\ b_{n+1} &= (n+1)\xi - \sum_{j=1}^n x_j \end{aligned}$$

and check whether $\xi = 1$, and the vector $\boldsymbol{x} \in \{0,1\}^n$. If so, $\boldsymbol{x}$ solves the subset sum problem (55). Coster et al. show that for $\boldsymbol{x} \in \{0,1\}^n$, $\xi = 1$, we obtain $||\boldsymbol{b}||^2 \leq \frac{n^3}{4}$, and they indicate how to show that most of the time there will be no shorter vectors in $L''_{\boldsymbol{a},a_0}$.

6.2 Solving systems of linear Diophantine equations

Aardal, Hurkens, and Lenstra [2], [3] considered the following integer feasibility problem:

$$\text{Does there exist a vector } \boldsymbol{x} \in \mathbb{Z}^n \text{ such that } \boldsymbol{Ax} = \boldsymbol{d},\ \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}? \tag{63}$$

Here $\boldsymbol{A}$ is an integer $m \times n$-matrix, with $m \leq n$, and the integer vectors $\boldsymbol{d}$, $\boldsymbol{l}$, and $\boldsymbol{u}$ are of compatible dimensions. Problem (63) is NP-complete, but if we remove the bound constraints $\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$, it is polynomially solvable. A standard way of tackling problem (63) is by branch-and-bound, but for the applications considered by Aardal et al. this method did not work well. Let $X = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{d},\ \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$. Instead of using a method based on the linear relaxation of the problem, they considered the following integer relaxation of X, $X_{\mathrm{IR}} = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{d}\}$. Determining whether X_{IR} is empty can be carried out in polynomial time for instance by generating the Hermite normal form of the matrix $\boldsymbol{A}$. Assume that X_{IR} is nonempty. Let $\boldsymbol{x}_f$ be an integer vector satisfying $\boldsymbol{A}\boldsymbol{x}_f = \boldsymbol{d}$, and let $\boldsymbol{B}^{\mathbf{0}}$ be an $n \times (n-m)$-matrix consisting of integer, linearly independent column vectors $\boldsymbol{b}^{\mathbf{0}}_j$, $1 \leq j \leq n-m$, such that $\boldsymbol{A}\boldsymbol{b}^{\mathbf{0}}_j = \mathbf{0}$ for $1 \leq j \leq n-m$. Notice that the matrix $\boldsymbol{B}^{\mathbf{0}}$ is a basis for the lattice $L_{\mathbf{0}} = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \boldsymbol{A}\boldsymbol{x} = \mathbf{0}\}$. We can now rewrite X_{IR} as

$$X_{\mathrm{IR}} = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \boldsymbol{x} = \boldsymbol{x}_f + \boldsymbol{B}^{\mathbf{0}}\boldsymbol{\lambda},\ \boldsymbol{\lambda} \in \mathbb{Z}^{n-m}\}\,. \tag{64}$$

Since a lattice has infinitely many bases if the dimension is greater than 1, reformulation (64) is not unique if $n - m > 1$.

The intuition behind the approach of Aardal et al. is as follows. Suppose it is possible to obtain a vector $\boldsymbol{x}_f$ that is short with respect to the bounds. Then, we may hope that $\boldsymbol{x}_f$ satisfies $\boldsymbol{l} \leq \boldsymbol{x}_f \leq \boldsymbol{u}$, in which case we are done. If $\boldsymbol{x}_f$ does not satisfy the bounds, then one can observe that $\boldsymbol{A}(\boldsymbol{x}_f + \lambda\boldsymbol{y}) = \boldsymbol{d}$ for any integer multiplier λ and any vector $\boldsymbol{y}$ satisfying $\boldsymbol{A}\boldsymbol{y} = \mathbf{0}$. Hence, it is possible to derive an enumeration scheme in which we branch on integer linear combinations of vectors $\boldsymbol{b}^{\mathbf{0}}_j$ satisfying $\boldsymbol{A}\boldsymbol{b}^{\mathbf{0}}_j = \mathbf{0}$, which explains the reformulation (64) of X_{IR}. Similar to Lagarias and Odlyzko, Aardal et al. choose a lattice, different from the standard lattice $\mathbb{Z}^n$, and then apply basis reduction to the initial basis of the chosen lattice. Since they obtain both $\boldsymbol{x}_f$ and the basis $\boldsymbol{B}^{\mathbf{0}}$ by basis reduction, $\boldsymbol{x}_f$ is relatively short and the columns of $\boldsymbol{B}^{\mathbf{0}}$ are near-orthogonal.

Aardal et al. [3] suggested a lattice $L_{\boldsymbol{A},\boldsymbol{d}} \in \mathbb{Z}^{n+m+1}$ that contains vectors of the following form:

$$(\boldsymbol{x}^T,\ N_1\xi,\ N_2(\boldsymbol{a}^1\boldsymbol{x} - d_1\xi), \ldots, N_2(\boldsymbol{a}^m\boldsymbol{x} - d_m\xi))^T, \tag{65}$$

where $\boldsymbol{a}^i$ is the i-th row of the matrix $\boldsymbol{A}$, where N_1 and N_2 are natural numbers, and where ξ, as in Section 6.1, is a variable associated with the right-hand side vector $\boldsymbol{d}$. The basis $\boldsymbol{B}$ given below spans the lattice $L_{\boldsymbol{A},\boldsymbol{d}}$:

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{I}^{(n)} & \mathbf{0}^{(n\times 1)} \\ \mathbf{0}^{(1\times n)} & N_1 \\ N_2\boldsymbol{A} & -N_2\boldsymbol{d} \end{pmatrix}. \tag{66}$$

The lattice $L_{\boldsymbol{A},\boldsymbol{d}} \subset \mathbb{Z}^{m+n+1}$ is not full-dimensional as $\boldsymbol{B}$ only contains $n+1$ columns. The numbers N_1 and N_2 are chosen so as to guarantee that certain elements of the reduced basis are equal to zero (cf. the similar role of the number N used in the bases (60) and (62)). The following proposition states precisely which type of vectors one wishes to obtain.

Proposition 8 ([3]). *The integer vector $\boldsymbol{x}_f$ satisfies $\boldsymbol{A}\boldsymbol{x}_f = \boldsymbol{d}$ if and only if the vector*

$$((\boldsymbol{x}_f)^T, N_1, \boldsymbol{0}^{(1\times m)})^T = \boldsymbol{B}\begin{pmatrix} \boldsymbol{x}_f \\ 1 \end{pmatrix} \tag{67}$$

belongs to the lattice L, and the integer vector $\boldsymbol{y}$ satisfies $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}$ if and only if the vector

$$(\boldsymbol{y}^T, 0, \boldsymbol{0}^{(1\times m)})^T = \boldsymbol{B}\begin{pmatrix} \boldsymbol{y} \\ 0 \end{pmatrix} \tag{68}$$

belongs to the lattice L.

Let $\hat{\boldsymbol{B}}$ be the basis obtained by applying Lovász' basis reduction algorithm to the basis $\boldsymbol{B}$, and let $\hat{\boldsymbol{b}}_j = (\hat{b}_j^1, \ldots, \hat{b}_j^{n+m+1})$ be the j-th column vector of $\hat{\boldsymbol{B}}$. Aardal et al. [3] prove that if the numbers N_1 and N_2 are chosen appropriately, then the $(n-m+1)$-st column of $\hat{\boldsymbol{B}}$ is of type (67), and the first $n-m$ columns of $\hat{\boldsymbol{B}}$ are of type (68), i.e., the first $n-m+1$ columns of $\hat{\boldsymbol{B}}$ are of the following form:

$$\begin{pmatrix} \boldsymbol{B}^{\boldsymbol{0}} & \boldsymbol{x}_f \\ \boldsymbol{0}^{(1\times(n-m))} & N_1 \\ \boldsymbol{0}^{(m\times(n-m))} & \boldsymbol{0}^{(m\times 1)} \end{pmatrix}. \tag{69}$$

This result is stated in the following theorem.

Theorem 22 ([3]). *Assume that there exists an integer vector $\boldsymbol{x}$ satisfying the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{d}$. There exist numbers N_{01} and N_{02} such that if $N_1 > N_{01}$, and if $N_2 > 2^{n+m}N_1^2 + N_{02}$, then the vectors $\hat{\boldsymbol{b}}_j \in \mathbb{Z}^{n+m+1}$ of the reduced basis $\hat{B}$ have the following properties:*

1. $\hat{b}_j^{n+1} = 0$ *for* $1 \le j \le n-m$,
2. $\hat{b}_j^i = 0$ *for* $n+2 \le i \le n+m+1$ *and* $1 \le j \le n-m+1$,
3. $|\hat{b}_{n-m+1}^{n+1}| = N_1$.

Moreover, the sizes of N_{01} *and* N_{02} *are polynomially bounded in the sizes of* $\boldsymbol{A}$ *and* $\boldsymbol{d}$*.*

In the proof of Properties 1 and 2 of Theorem 22, Aardal et al. make use of inequality (15) of Proposition 2.

Once we have obtained the matrix $\boldsymbol{B^0}$ and the vector $\boldsymbol{x}_f$, we can derive the following equivalent formulation of problem (63):

$$\text{Does there exist a vector } \boldsymbol{\lambda} \in \mathbb{Z}^{n-m} \text{ such that } \boldsymbol{l} \leq \boldsymbol{x}_f + \boldsymbol{B^0}\boldsymbol{\lambda} \leq \boldsymbol{u}? \tag{70}$$

Aardal, Hurkens, and Lenstra [3], and Aardal, Bixby, Hurkens, Lenstra, and Smeltink [1] investigated the effect of the reformulation on the number of nodes of a linear programming based branch-and-bound algorithm. They considered three sets of instances: instances obtained from Philips Research Labs, the Frobenius instances of Cornuéjols, Urbaniak, Weismantel, and Wolsey [33], and the market split instances of Cornuéjols and Dawande [31]. The results were encouraging. For instance, after transforming problem (63) to problem (70), the size of the market split instances that could be solved doubled.

Aardal et al. [1] also investigated the performance of *integer branching.* They implemented a branching-on-hyperplanes search algorithm, such as the algorithms in Section 4. Instead of finding provably good directions they branched on hyperplanes in the directions of the unit vectors $\boldsymbol{e}_j$, $1 \leq j \leq n-m$ in the space of the $\boldsymbol{\lambda}$-variables.

Their computational study indicated that integer branching on the unit vectors taken in the order $j = n-m, \ldots, 1$, was quite effective, and in general much better than the order $1, \ldots, n-m$. This can be explained as follows. Due to Lovász' algorithm, the vectors of $\boldsymbol{B^0}$ are more or less in order of increasing length, so typically, the $(n-m)$-th vector of $\boldsymbol{B^0}$ is the longest one. Branching on this vector first should generate relatively few hyperplanes intersecting the linear relaxation of X, if this set has a regular shape, or equivalently, the polytope $P = \{\boldsymbol{\lambda} \in \mathbb{R}^{n-m} \mid \boldsymbol{l} \leq \boldsymbol{x}_f + \boldsymbol{B^0}\boldsymbol{\lambda} \leq \boldsymbol{u}\}$ is relatively thin in the unit direction $\boldsymbol{e}_{n-m}$ compared to direction $\boldsymbol{e}_1$. In this context Aardal and Lenstra [4] studied infeasible instances of the knapsack problem

$$\text{Does there exist a vector } \boldsymbol{x} \in \mathbb{Z}^n_{\geq 0} \text{ such that } \boldsymbol{a}\boldsymbol{x} = a_0\,?$$

Write a_j as $a_j = p_j M + r_j$ with $p_j, M \in \mathbb{N}_{>0}$, and $r_j \in \mathbb{Z}$. Aardal and Lenstra showed the following:

Theorem 23 ([4]). *Let* $\boldsymbol{b}^{\boldsymbol{0}}_{n-1}$ *be the last vector of the basis matrix* $\boldsymbol{B^0}$ *as obtained in (69). The following holds:*

- $d(L_{\mathbf{0}}) = ||\boldsymbol{a}^T||$,
- $||\boldsymbol{b}_{n-1}^{\mathbf{0}}|| \geq \frac{|\boldsymbol{a}^T|}{\sqrt{|\boldsymbol{p}|^2 \cdot |\boldsymbol{r}|^2 - (\boldsymbol{p}\boldsymbol{r}^T)^2}}$.

If M is large, then $d(L_{\mathbf{0}}) = ||\boldsymbol{a}^T||$ will be large, and if $\boldsymbol{p}$ and $\boldsymbol{r}$ are short compared to $\boldsymbol{a}$ the vector $\boldsymbol{b}_{n-1}^{\mathbf{0}}$ is going to be long, so in this case the value of $d(L_{\mathbf{0}})$ essentially comes from the length of the last basis vector. In their computational study it was clear that branching in the direction of the last basis vector first gave rise to extremely small search trees.

Example 3. Let $\boldsymbol{a} = (12223,\ 12224,\ 36671)$. We can decompose $\boldsymbol{a}$ as

$$\begin{array}{lcl} a_1 & = & M + 0 \\ a_2 & = & M + 1 \\ a_3 & = & 3M + 2 \end{array}$$

with $M = 12223$. For this example we obtain

$$\boldsymbol{x}_f = \begin{pmatrix} -4075 \\ 4074 \\ 4074 \end{pmatrix} \qquad \boldsymbol{B}^0 = \begin{pmatrix} -1 & 14261 \\ -2 & -8149 \\ 1 & -2037 \end{pmatrix}.$$

The polytope P is:

$$P = \{\boldsymbol{y} \in \mathbb{R}^2 \mid -\lambda_1 + 14261\lambda_2 \geq 4075,\ -2\lambda_1 - 8149\lambda_2 \geq -4074,\ \lambda_1 - 2037\lambda_2 \geq -4074\}.$$

The constraints imply that that $0 < \lambda_2 < 1$, so branching first in the direction of $\boldsymbol{e}_2$ immediately yields a certificate of infeasibility. Searching in direction $\boldsymbol{e}_1$ first yields 4752 search nodes at the first level of our search tree. Solving the instance using the original formulation in $\boldsymbol{x}$-variables requires 1,262,532 search nodes using CPLEX 6.5 with default settings. □

Recently, Louveaux and Wolsey [78] considered the problem: "Does there exist a matrix $\boldsymbol{X} \in \mathbb{Z}_{\geq 0}^{m \times n}$ such that $\boldsymbol{X}\boldsymbol{A} = \boldsymbol{C}$, and $\boldsymbol{B}\boldsymbol{X} = \boldsymbol{D}$?", where $\boldsymbol{A} \in \mathbb{Z}^{n \times p}$ and $\boldsymbol{B} \in \mathbb{Z}^{q \times m}$. Their study was motivated by a portfolio planning problem, where variable x_{ij} denotes the number of shares of type j included in portfolio i. This problem can be written in the same form as problem (63), so in principle the approach discussed in this section could be applied. For reasonable problem sizes Louveaux and Wolsey observed that the basis reduction step became too time consuming. Instead they determined reduced bases for the lattices $L_{\mathbf{0}}^A = \{\boldsymbol{y} \in \mathbb{Z}^n \mid \boldsymbol{y}^T\boldsymbol{A} = \mathbf{0}\}$, and $L_{\mathbf{0}}^B = \{\boldsymbol{z} \in \mathbb{Z}^m \mid \boldsymbol{B}\boldsymbol{z} = \mathbf{0}\}$. Let $\boldsymbol{B}_A$ be a basis for the lattice $L_{\mathbf{0}}^A$, and let $\boldsymbol{B}_B$ be a basis for the lattice $L_{\mathbf{0}}^B$. They showed that taking the so-called *Kronecker product* of the matrices $\boldsymbol{B}_A^T$ and $\boldsymbol{B}_B$ yields a basis for the lattice

$L_{\mathbf{0}} = \{\boldsymbol{X} \in \mathbb{Z}^{m\times n} \mid \boldsymbol{XA} = \mathbf{0},\ \boldsymbol{BX} = \mathbf{0}\}$. The Kronecker product of two matrices $\boldsymbol{M} \in \mathbb{R}^{m\times n}$, and $\boldsymbol{N} \in \mathbb{R}^{p\times q}$ is defined as:

$$\boldsymbol{M} \otimes \boldsymbol{N} = \begin{pmatrix} m_{11}\boldsymbol{N} & \cdots & m_{1n}\boldsymbol{N} \\ \ldots & \ddots & \ldots \\ m_{m1}\boldsymbol{N} & \cdots & m_{mn}\boldsymbol{N} \end{pmatrix}.$$

Moreover, they showed that the basis of $L_{\mathbf{0}}$ obtained by taking the Kronecker product between $\boldsymbol{B}_A^T$ and $\boldsymbol{B}_B$ is reduced, up to a reordering of the basis vectors, if the bases $\boldsymbol{B}_A$ and $\boldsymbol{B}_B$ are reduced. Computational experience is reported.

7 Integer hulls and cutting plane closures in fixed dimension

An integer optimization problem $\max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{Ax} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}^n\}$, for integral $\boldsymbol{A}$ and $\boldsymbol{b}$, can be interpreted as the linear programming problem $\max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{A}'\boldsymbol{x} \leq \boldsymbol{b}',\ \boldsymbol{x} \in \mathbb{R}^n\}$, where $\boldsymbol{A}'\boldsymbol{x} \leq \boldsymbol{b}'$ is an inequality description of the integer hull of the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{Ax} \leq \boldsymbol{b}\}$. We have seen that the integer optimization problem in fixed dimension can be solved in polynomial time. The question now is, how large can the integer hull of a polyhedron be if the dimension is fixed? Can the integer hull be described with a polynomial number of inequalities and if the answer is yes, can these inequalities be computed in polynomial time? It turns out that the answer to both questions is "yes", as we will see in the following section.

One of the most successful methods to attack an integer optimization problem in practice is branch-and-bound combined with the addition of cutting planes. Cutting planes are valid inequalities for the integer hull, which are not necessarily valid for the linear relaxation of the problem. A famous family of cutting planes, also historically the first ones, are Gomory-Chvátal cutting planes [53]. In the second part of this section, we consider the question, whether the polyhedron that results from the application of all possible Gomory-Chvátal cutting planes, the so-called elementary closure, has a polynomial representation in fixed dimension. Furthermore we address the problem of constructing the elementary closure in fixed dimension.

7.1 The integer hull

In this section we describe a result of Hayes and Larman [56] and its generalization by Schrijver [99] which states that P_I can be described with a polynomial number of inequalities in fixed dimension, provided that P is rational.

We start by proving a polynomial upper bound on the number of vertices of the integer hull of a full-dimensional simplex $\Sigma = \mathrm{conv}\{\mathbf{0}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$. Let φ denote the maximum binary encoding length of a vertex $\varphi = \max_{i=1,\ldots,n} \mathrm{size}(\boldsymbol{v}_i)$. A full dimensional simplex in $\mathbb{R}^n$ is defined by $n+1$ inequalities. Each choice of n inequalities in such a definition has linearly independent normal vectors, defining one of the vertices of Σ. Since $\mathbf{0}$ is one of the vertices, Σ is the set of all $\boldsymbol{x} \in \mathbb{R}^n$ satisfying $\boldsymbol{B}\boldsymbol{x} \geq \mathbf{0}$, $\boldsymbol{c}^T\boldsymbol{x} \leq \beta$, where $\boldsymbol{B} \in \mathbb{Z}^{n\times n}$ is a nonsingular matrix, and $\boldsymbol{c}^T\boldsymbol{x} \leq \beta$ is an inequality. It follows form the Hadamard bound that we can choose $\boldsymbol{B}$ such that $\mathrm{size}(\boldsymbol{B}) = \mathrm{O}(\varphi)$. The inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \beta$ can be rewritten as $\boldsymbol{a}^T\boldsymbol{B}\boldsymbol{x} \leq \beta$, with $\boldsymbol{a}^T = \boldsymbol{c}^T\boldsymbol{B}^{-1} \in \mathbb{Q}^n$. Let K be the knapsack polytope $K = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \geq 0, \boldsymbol{a}^T\boldsymbol{x} \leq \beta\}$. The vertices of Σ_I correspond exactly to the vertices of $\mathrm{conv}(K \cap L(\boldsymbol{B}))$.

Proposition 9. *Let $K \subseteq \mathbb{R}^n$ be a knapsack polytope given by the inequalities $\boldsymbol{x} \geq 0$ and $\boldsymbol{a}^T\boldsymbol{x} \leq \beta$. Let $L(\boldsymbol{B})$ be a lattice with integer and nonsingular $\boldsymbol{B} \subseteq \mathbb{Z}^{n\times n}$, then:*

1. *A vector $\boldsymbol{B}\hat{\boldsymbol{x}} \in L(\boldsymbol{B})$ is a vertex of $\mathrm{conv}(K \cap L(\boldsymbol{B}))$ if and only if $\hat{\boldsymbol{x}}$ is a vertex of the integer hull of the simplex Σ defined by $\boldsymbol{B}\boldsymbol{x} \geq \mathbf{0}$ and $\boldsymbol{a}^T\boldsymbol{B}\boldsymbol{x} \leq \beta$;*

2. *if $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are distinct vertices of $\mathrm{conv}(K \cap L(\boldsymbol{B}))$, then there exists an index $i \in \{1, \ldots, n\}$ such that $\mathrm{size}(\boldsymbol{v}_1^i) \neq \mathrm{size}(\boldsymbol{v}_2^i)$.*

Proof. The convex hull of $K \cap L(\boldsymbol{B})$ can be written as

$$\begin{aligned} \mathrm{conv}(K \cap L(\boldsymbol{B})) &= \mathrm{conv}(\{\boldsymbol{x} \mid \boldsymbol{x} \geq \mathbf{0},\, \boldsymbol{a}^T\boldsymbol{x} \leq \beta\,,\, \boldsymbol{x} = \boldsymbol{B}\boldsymbol{y},\, \boldsymbol{y} \in \mathbb{Z}^n) \\ &= \mathrm{conv}(\{\boldsymbol{B}\boldsymbol{y} \mid \boldsymbol{B}\boldsymbol{y} \geq \mathbf{0},\, \boldsymbol{a}^T\boldsymbol{B}\boldsymbol{y} \leq \beta,\, \boldsymbol{y} \in \mathbb{Z}^n\}). \end{aligned}$$

If one transforms this set with $\boldsymbol{B}^{-1}$, one is faced with the integer hull of the described simplex Σ. Thus Point (1) in the proposition follows.

For Point (2) assume that $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are vertices of $\mathrm{conv}(K \cap L(\boldsymbol{B}))$, with $\mathrm{size}(\boldsymbol{v}_1^i) = \mathrm{size}(\boldsymbol{v}_2^i)$ for all $i \in \{1, \ldots, n\}$. Then clearly $2\boldsymbol{v}_1 - \boldsymbol{v}_2 \geq 0$ and $2\boldsymbol{v}_2 - \boldsymbol{v}_1 \geq 0$. Also

$$\boldsymbol{a}^T(2\boldsymbol{v}_1 - \boldsymbol{v}_2 + 2\boldsymbol{v}_2 - \boldsymbol{v}_1) = \boldsymbol{a}^T(\boldsymbol{v}_1 + \boldsymbol{v}_2) \leq 2\beta,$$

therefore one of the two lattice points lies in K. Assume without loss of generality that $2\boldsymbol{v}_1 - \boldsymbol{v}_2 \in K \cap L(\boldsymbol{B})$. Then $\boldsymbol{v}_1$ cannot be a vertex since

$$\boldsymbol{v}_1 = 1/2\,(2\boldsymbol{v}_1 - \boldsymbol{v}_2) + 1/2\,\boldsymbol{v}_2.$$

□

If $K = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{a}^T\boldsymbol{x} \leq \beta\}$ is the corresponding knapsack polytope to the simplex Σ, then any component $\hat{x}_j$, $j = 1, \dots, n$ of an arbitrary point $\hat{\boldsymbol{x}}$ in K satisfies $0 \leq \hat{x}_j \leq \beta/a_j$. Thus the size of a vertex $\hat{\boldsymbol{x}}$ of $\mathrm{conv}(K \cap L(\boldsymbol{B}))$ is of $\mathrm{O}(\mathrm{size}(K)) = \mathrm{O}(\mathrm{size}(\Sigma))$ in fixed dimension. This is because $\mathrm{size}(\boldsymbol{B}^{-1}) = \mathrm{O}(\mathrm{size}(\boldsymbol{B}))$ in fixed dimension. It follows from Proposition 9 that Σ_I can have at most $\mathrm{O}(\mathrm{size}(\Sigma)^n)$ vertices.

By translation with the vertex $-\boldsymbol{v}_0$, we can assume that $\Sigma = \mathrm{conv}(\boldsymbol{v}_0, \dots, \boldsymbol{v}_n)$ is a simplex whose first vertex $\boldsymbol{v}_0$ is *integral*.

Lemma 6 ([56, 99]). *Let $\Sigma = \mathrm{conv}(\boldsymbol{v}_0, \dots, \boldsymbol{v}_n)$ be a rational simplex with $\boldsymbol{v}_0 \in \mathbb{Z}^n$, $\boldsymbol{v}_i \in \mathbb{Q}^n$, $i = 1, \dots, n$. The number of vertices of the integer hull Σ_I is bounded by $\mathrm{O}(\varphi^n)$, where $\varphi = \max_{i=0,\dots,n} \mathrm{size}(\boldsymbol{v}_i)$.*

A polynomial bound for general polyhedra can then be found by triangulation.

Theorem 24 ([56, 99]). *Let $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$, where $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ and $\boldsymbol{d} \in \mathbb{Z}^m$, be a rational polyhedron where each inequality in $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ has size at most φ. The integer hull P_I of P has at most $\mathrm{O}(m^{n-1}\varphi^n)$ vertices.*

The following upper bound on the number of vertices of P_I was proved by Cook et al. [28]. Barany et al. [10] showed that this bound is tight if P is a simplex.

Theorem 25. *If $P \subseteq \mathbb{R}^n$ is a rational polyhedron that is the solution set of a system of at most m linear inequalities whose size is at most φ, then the number of vertices of P_I is at most $2m^d(6n^2\varphi)^{d-1}$, where $d = \dim(P_I)$ is the dimension of the integer hull of P.*

Tight bounds for varying number of inequalities m seem to be unknown.

7.2 Cutting planes

Rather than computing the integer hull P_I of P, the objective pursued by the *cutting plane method* is a better approximation of P_I. Here the idea is to intersect P with the integer hull of halfspaces containing P. These will still include P_I but not necessarily P.

In the following we will study the theoretical framework of Gomory's cutting plane method [53] as given by Chvátal [23] and Schrijver [98] and derive a polynomiality result on the number of facets of the polyhedron that results from the application of all possible cutting planes.

If the halfspace $(\boldsymbol{c}^T\boldsymbol{x} \leq \delta)$, $c \in \mathbb{Z}^n$, with $\gcd(c_1, \dots, c_n) = 1$, contains the polyhedron P, i.e. if $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ is valid for P, then $\boldsymbol{c}^T\boldsymbol{x} \leq \lfloor \delta \rfloor$ is valid for the integer hull P_I of P. The inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \lfloor\delta\rfloor$ is called a *cutting plane* or *Gomory-Chvátal cut* of P. The geometric interpretation behind this process is that $(\boldsymbol{c}^T\boldsymbol{x} \leq \delta)$ is "shifted inwards" until an integer point of the lattice is in the boundary of the halfspace.

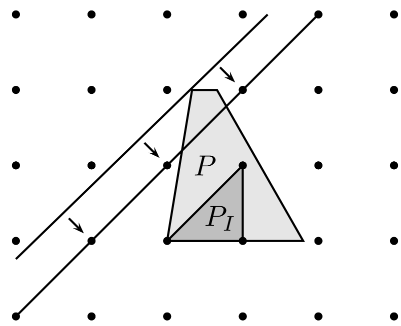


Figure 9: The halfspace $(-x_1 + x_2 \leq \delta)$ containing P is replaced by its integer hull $(-x_1 + x_2 \leq \lfloor\delta\rfloor)$. The darker region is the integer hull P_I of P.

The idea, pioneered by Gomory [53], is to apply these cutting planes to the integer optimization problem. Cutting planes tighten the linear relaxation of an integer program and Gomory showed how to apply cutting planes successively until the resulting relaxation has an integer optimal solution.

7.2.1 The elementary closure

Cutting planes $\boldsymbol{c}^T\boldsymbol{x} \leq \lfloor\delta\rfloor$ of $P(\boldsymbol{A}, \boldsymbol{d})$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ obey a simple *inference rule*. Clearly $\max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\} \leq \delta$ and it follows from duality and Carathéodory's theorem that there exists a weight vector $\boldsymbol{\lambda} \in \mathbb{Q}^m_{\geq 0}$ with at most n positive entries such that $\boldsymbol{\lambda}^T\boldsymbol{A} = \boldsymbol{c}^T$ and $\boldsymbol{\lambda}^T\boldsymbol{d} \leq \delta$. Thus $\boldsymbol{c}^T\boldsymbol{x} \leq \lfloor\delta\rfloor$ follows from the following inequalities by weakening the right-hand side if necessary:

$$\boldsymbol{\lambda}^T\boldsymbol{A}\boldsymbol{x} \leq \lfloor\boldsymbol{\lambda}^T\boldsymbol{d}\rfloor,\ \boldsymbol{\lambda} \in \mathbb{Q}^m_{\geq 0},\ \boldsymbol{\lambda}^T\boldsymbol{A} \in \mathbb{Z}^n. \tag{71}$$

Instead of applying cutting planes successively, one can apply all possible cutting planes at once. P intersected with all Gomory-Chvátal cutting planes

$$P' = \bigcap_{\substack{(\boldsymbol{c}^T\boldsymbol{x}\leq\delta)\supseteq P \\ \boldsymbol{c}\in\mathbb{Z}^n}} (\boldsymbol{c}^T\boldsymbol{x} \leq \lfloor\delta\rfloor) \tag{72}$$

is called the *elementary closure* of P.

The set of inequalities in (71) that describe P' is infinite. However, as observed by Schrijver [98], a finite number of inequalities in (71) imply the rest.

Lemma 7. *Let P be the polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$ with $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$ and $\boldsymbol{d} \in \mathbb{Z}^m$. The elementary closure P' is the polyhedron defined by $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ and the set of all inequalities $\boldsymbol{\lambda}^T\boldsymbol{A}\boldsymbol{x} \leq \lfloor \boldsymbol{\lambda}^T\boldsymbol{d}\rfloor$, where $\boldsymbol{\lambda} \in [0,1)^m$ and $\boldsymbol{\lambda}^T\boldsymbol{A} \in \mathbb{Z}^n$.*

Proof. An inequality $\boldsymbol{\lambda}^T\boldsymbol{A}\boldsymbol{x} \leq \lfloor \boldsymbol{\lambda}^T\boldsymbol{d}\rfloor$ with $\boldsymbol{\lambda} \in \mathbb{Q}^m_{\geq 0}$ and $\boldsymbol{\lambda}^T\boldsymbol{A} \in \mathbb{Z}^n$ is implied by $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ and $(\boldsymbol{\lambda} - \lfloor\boldsymbol{\lambda}\rfloor)^T\boldsymbol{A}\boldsymbol{x} \leq \lfloor(\boldsymbol{\lambda} - \lfloor\boldsymbol{\lambda}\rfloor)^T\boldsymbol{d}\rfloor$, since

$$\boldsymbol{\lambda}^T\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{\lambda}-\lfloor\boldsymbol{\lambda}\rfloor)^T\boldsymbol{A}\boldsymbol{x}+\lfloor\boldsymbol{\lambda}\rfloor^T\boldsymbol{A}\boldsymbol{x} \leq \lfloor(\boldsymbol{\lambda}-\lfloor\boldsymbol{\lambda}\rfloor)^T\boldsymbol{d}\rfloor+\lfloor\boldsymbol{\lambda}\rfloor^T\boldsymbol{d} = \lfloor\boldsymbol{\lambda}^T\boldsymbol{d}\rfloor. \tag{73}$$

□

Corollary 2 ([98]). *If P is a rational polyhedron, then P' is a rational polyhedron.*

Proof. P can be described as $P(\boldsymbol{A}, \boldsymbol{d})$ with integral $\boldsymbol{A}$ and $\boldsymbol{d}$. There is only a finite number of vectors $\boldsymbol{\lambda}^T\boldsymbol{A} \in \mathbb{Z}^n$ with $\boldsymbol{\lambda} \in [0,1)^m$. □

This yields an exponential upper bound on the number of facets of the elementary closure of a polyhedron. The infinity norm $\|\boldsymbol{c}\|_\infty$ of a possible candidate $\boldsymbol{c}^T\boldsymbol{x} \leq \lfloor\delta\rfloor$ is bounded by $\|\boldsymbol{A}^T\|_\infty$, where the matrix norm $\|\cdot\|_\infty$ is the row sum norm. Therefore we have an upper bound of $\mathrm{O}(\|\boldsymbol{A}^T\|_\infty^n)$ for the number of facets of the elementary closure of a polyhedron. We will later prove a polynomial upper bound of the size of P' in fixed dimension.

7.2.2 The Chvátal-Gomory procedure

The elementary closure operation can be iterated, so that successively tighter relaxations of the integer hull P_I of P are obtained. We define $P^{(0)} = P$ and $P^{(i+1)} = (P^{(i)})'$, for $i \geq 0$. This iteration of the elementary closure operation is called the *Chvátal-Gomory procedure*. The *Chvátal rank* of a polyhedron P is the smallest $t \in \mathbb{N}_0$ such that $P^{(t)} = P_I$. In analogy, the *depth* of an inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ which is valid for P_I is the smallest $t \in \mathbb{N}_0$ such that $(\boldsymbol{c}^T\boldsymbol{x} \leq \delta) \supseteq P^{(t)}$.

Chvátal [23] showed that every bounded polyhedron $P \subseteq \mathbb{R}^n$ has finite rank. Schrijver [98] extended this result to rational polyhedra. The main ingredient of his result is the following result.

Lemma 8 ([98]). *Let F be a face of a rational polyhedron P. If $\boldsymbol{c}_F^T\boldsymbol{x} \leq \lfloor \delta_F \rfloor$ is a cutting plane for F, then there exists a cutting plane $\boldsymbol{c}_P^T\boldsymbol{x} \leq \lfloor \delta_P \rfloor$ for P with*

$$F \cap (\boldsymbol{c}_P^T\boldsymbol{x} \leq \lfloor \delta_P \rfloor) = F \cap (\boldsymbol{c}_F^T\boldsymbol{x} \leq \lfloor \delta_F \rfloor).$$

Intuitively, this result means that that a cutting plane of a face F of a polyhedron P can be "rotated" so that it becomes a cutting plane of P and has the same effect on F. This implies that a face F of P behaves under its closure F' as it behaves under the closure P' of P.

Corollary 3. *Let F be a face of a rational polyhedron P. Then*

$$F' = P' \cap F.$$

From this, one can derive that the Chvátal rank of rational polyhedra is finite.

Theorem 26 ([98]). *If P is a rational polyhedron, then there exists some $t \in \mathbb{N}$ such that $P^{(t)} = P_I$.*

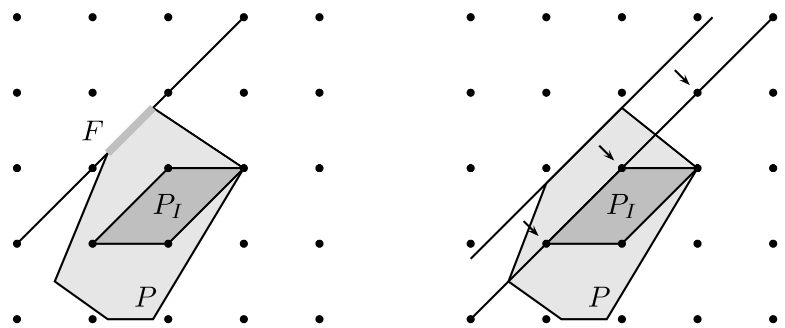


Figure 10: After a finite number of iterations F is empty. Then the halfspace defining F can be pushed further down. This is basically the argument why every inequality, valid for P_I, eventually becomes valid for the outcome of the successive application of the elementary closure operation.

Already in dimension 2, there exist rational polyhedra of arbitrarily large Chvátal rank [23]. To see this, consider the class of polytopes

$$P_k = \operatorname{conv}\{(0,0),(0,1)(k,\tfrac{1}{2})\},\ k \in \mathbb{N}. \tag{74}$$

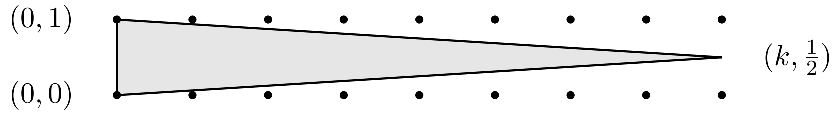


Figure 11: The polytope P_k.

One can show that $P_{(k-1)} \subseteq P'_k$. For this, let $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ be valid for P_k with $\delta = \max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in P_k\}$. If $c_1 \leq 0$, then the point $(0,0)$ or $(0,1)$ maximizes $\boldsymbol{c}^T\boldsymbol{x}$, thus $(\boldsymbol{c}^T\boldsymbol{x} = \delta)$ contains integer points. If $c_1 > 0$, then $\boldsymbol{c}^T(k, \frac{1}{2}) \geq \boldsymbol{c}^T(k-1, \frac{1}{2}) + 1$. Therefore the point $(k-1, \frac{1}{2})$ is in the halfspace $(\boldsymbol{c}^T\boldsymbol{x} \leq \delta - 1) \subseteq (\boldsymbol{c}^T\boldsymbol{x} \leq \lfloor\delta\rfloor)$. Unfortunately, this lower bound on the Chvátal rank of P_k is exponential in the encoding length of P_k which is $\mathrm{O}(\log(k))$.

Bockmayr et al. [16] have shown that the Chvátal rank of polytopes in the 0/1 cube is polynomial. The current best bound [44] on the Chvátal rank of polytopes in the 0/1 cube is $\mathrm{O}(n^2 \log n)$. Lower bounds on the Chvátal rank for polytopes stemming from combinatorial optimization problems have been provided by Chvátal, Cook and Hartmann [24]. Cook and Dash [30] provided lower bounds on the matrix-cut rank of polytopes in the 0/1 cube. In particular they provide examples with rank n and so do Cornuéjols and Li [32] for the split closure in the 0/1 cube.

7.2.3 Cutting plane proofs

An important property of polyhedra is the following rule to derive valid inequalities, which is a consequence of linear programming duality. If P is defined by the inequalities $\boldsymbol{Ax} \leq \boldsymbol{d}$, then the inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ is valid for P if and only if there exists some $\boldsymbol{\lambda} \in \mathbb{R}^m_{\geq 0}$ with

$$\boldsymbol{c} = \boldsymbol{\lambda}^T\boldsymbol{A} \text{ and } \delta \geq \boldsymbol{\lambda}^T\boldsymbol{d}. \tag{75}$$

This implies that linear programming (in its decision version) belongs to the class $\mathrm{NP} \cap \mathrm{co} - \mathrm{NP}$, because $\max\{\boldsymbol{c}^T x \mid \boldsymbol{Ax} \leq \boldsymbol{d}\} \leq \delta$ if and only if $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ is valid for $P(\boldsymbol{A}, \boldsymbol{d})$. A "No" certificate would be some vertex of P which violates $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$.

In integer programming there is an analogy to this rule. A sequence of inequalities

$$\boldsymbol{c}_1^T\boldsymbol{x} \leq \delta_1, \boldsymbol{c}_2^T\boldsymbol{x} \leq \delta_2, \ldots, \boldsymbol{c}_m^T\boldsymbol{x} \leq \delta_m \tag{76}$$

is called a *cutting-plane proof* of $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ from a given system of linear inequalities $\boldsymbol{Ax} \leq \boldsymbol{d}$, if $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m$ are integral, $\boldsymbol{c}_m = \boldsymbol{c}$, $\delta_m = \delta$, and if $\boldsymbol{c}_i^T\boldsymbol{x} \leq$

δ_i' is a nonnegative linear combination of $\boldsymbol{A}x \le \boldsymbol{d}, \boldsymbol{c}_1^T\boldsymbol{x} \le \delta_1, \ldots, \boldsymbol{c}_{i-1}^T\boldsymbol{x} \le \delta_{i-1}$ for some δ_i' with $\lfloor \delta_i' \rfloor \le \delta_i$. In other words, if $\boldsymbol{c}_i^T\boldsymbol{x} \le \delta_i$ can be obtained from $\boldsymbol{Ax} \le \boldsymbol{d}$ and the previous inequalities as a Gomory-Chvátal cut, by weakening the right-hand-side if necessary. Obviously, if there is a cutting-plane proof of $\boldsymbol{c}^T\boldsymbol{x} \le \delta$ from $\boldsymbol{Ax} \le \boldsymbol{d}$ then every integer solution to $\boldsymbol{Ax} \le \boldsymbol{d}$ must satisfy $\boldsymbol{c}^T\boldsymbol{x} \le \delta$. The number m here, is the *length* of the cutting plane proof.

The following proposition shows a relation between the length of cutting plane proofs and the depth of inequalities (see also [24]). It comes in two flavors, one for the case $P_I \neq \emptyset$ and one for $P_I = \emptyset$. The latter can then be viewed as an analogy to Farkas' lemma.

Proposition 10 ([24]). *Let $P(\boldsymbol{A}, \boldsymbol{d}) \subseteq \mathbb{R}^n$, $n \ge 2$ be a rational polyhedron.*

1. *If $P_I \neq \emptyset$ and $\boldsymbol{c}^T\boldsymbol{x} \le \delta$ with integer $\boldsymbol{c}$ has depth t, then $\boldsymbol{c}^T\boldsymbol{x} \le \delta$ has a cutting plane proof of length at most $(n^{t+1} - 1)/(n-1)$.*

2. *If $P_I = \emptyset$ and $\operatorname{rank}(P) = t$, then there exists a cutting plane proof of $\boldsymbol{0}^T\boldsymbol{x} \le -1$ of length at most $(n+1)(n^t - 1)/(n-1) + 1$.*

We have seen for the class of polytopes P_k (74) that, even in fixed dimension, a cutting plane proof of minimal length can be exponential in the binary encoding length of the given polyhedron.

Yet, if $P_I = \emptyset$ and $P \subseteq \mathbb{R}^n$, Cook, Coullard and Turán [27] showed that there exists a number $t(n)$, such that $P^{(t(n))} = \emptyset$.

Theorem 27 ([27]). *There exists a function $t(d)$, such that if $P \subseteq \mathbb{R}^n$ is a d-dimensional rational polyhedron with empty integer hull, then $P^{t(d)} = \emptyset$.*

Proof. If P is not full dimensional, then there exists a rational hyperplane $(\boldsymbol{c}^T\boldsymbol{x} = \delta)$ with $\boldsymbol{c} \in \mathbb{Z}^n$ and $\gcd(c_1, \ldots, c_n) = 1$ such that $P \subseteq (\boldsymbol{c}^T\boldsymbol{x} = \delta)$. If $\delta \notin \mathbb{Z}$, then $P' = \emptyset$. If $\delta \in \mathbb{Z}$, then there exists a unimodular matrix, transforming $\boldsymbol{c}$ into the first unit vector $\boldsymbol{e}_1$. Thus P can be transformed via a unimodular transformation into a polyhedron where the first variable is fixed to an integer.

Thus we can assume that P is full-dimensional. The function $t(d)$ is inductively defined. Let $t(0) = 1$. For $d > 0$, let $\boldsymbol{c} \in \mathbb{Z}^n$, $\boldsymbol{c} \neq \boldsymbol{0}$ be a direction in which P is flat (c.f. Theorem 9), i.e., $\max\{\boldsymbol{c}^T\boldsymbol{x} \mid x \in P\} - \min\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in P\} \le f(d)$. We "slice off" in this direction using Corollary 3. If $\boldsymbol{c}^T\boldsymbol{x} \le \delta$, $\delta \in \mathbb{Z}$ is valid for P, then $\boldsymbol{c}^T\boldsymbol{x} \le \delta - 1$ is valid for $P^{(t(d-1)+1)}$, since the face $F = P \cap (\boldsymbol{c}^T\boldsymbol{x} = \delta)$ has at most dimension $d-1$. Thus $\boldsymbol{c}^T\boldsymbol{x} \le \delta - k$ is valid for $P^{(k\,(t(d-1)+1))}$. Since the integer vector $\boldsymbol{c}$ is chosen such that

$\max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in P\} - \min\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in P\} \leq f(d)$, $t(d) = (f(d)+2)(t(d-1)+1)$ satisfies our needs. $\square$

The validity of an inequality $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ for P_I can be established by showing that $P \cap (\boldsymbol{c}^T\boldsymbol{x} \geq \delta + 1)$ is integer infeasible. A cutting plane proof for the integer infeasibility of $P \cap (\boldsymbol{c}^T\boldsymbol{x} \geq \delta + 1)$ is called an *indirect cutting plane proof* of $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$. Combining Proposition 10 and Theorem 27 one obtains the following result.

Theorem 28 ([27]). *Let P be a rational polyhedron in fixed dimension n and let $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ be a valid inequality for P, then $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ has an indirect cutting plane proof of constant length.*

In varying dimension, the length of a cutting plane proof of infeasibility of 0/1 systems can be exponential. This was shown by Pudlák [88]. Exponential lower bounds for other types of cutting-plane proofs provided by lift-and-project or Lovász-Schrijver cuts were derived by Dash [35].

7.3 The elementary closure in fixed dimension

In this section we will show that the elementary closure of rational polyhedra in fixed dimenison can be described with a polynomial number of inequalities.

7.3.1 Simplicial cones

Consider a *rational simplicial cone*, i.e., a polyhedron $P = \{x \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$, where $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$, $\boldsymbol{d} \in \mathbb{Z}^m$ and $\boldsymbol{A}$ has full row rank. If $\boldsymbol{A}$ is a square matrix, then P is called *pointed*.

Observe that P, P' and P_I are all full-dimensional. The elementary closure P' is given by the inequalities

$$(\boldsymbol{\lambda}^T\boldsymbol{A})\boldsymbol{x} \leq \lfloor \boldsymbol{\lambda}^T\boldsymbol{d} \rfloor, \text{ where } \boldsymbol{\lambda} \in [0,1]^m, \text{ and } \boldsymbol{\lambda}^T\boldsymbol{A} \in \mathbb{Z}^n. \tag{77}$$

Since P' is full-dimensional, there exists a unique (up to scalar multiplication) minimal subset of the inequalities in (77) that suffices to describe P'. These inequalities are the facets of P'. We will derive a polynomial upper bound on their number in fixed dimension.

The vectors $\boldsymbol{\lambda}$ in (77) belong to the dual lattice $L^*(\boldsymbol{A})$ of the lattice $L(\boldsymbol{A})$. Recall that each element in $L^*(\boldsymbol{A})$ is of the form $\boldsymbol{\mu}/d_L$, where $d_L = d(L(\boldsymbol{A}))$ is the lattice determinant. It follows from the Hadamard inequality that size(d_L) is polynomial in size($\boldsymbol{A}$), even for varying n. Now (77) can be rewritten as

$$\frac{\boldsymbol{\mu}^T \boldsymbol{A}}{d_L} \boldsymbol{x} \leq \left\lfloor \frac{\boldsymbol{\mu}^T \boldsymbol{d}}{d_L} \right\rfloor, \text{ where } \boldsymbol{\mu} \in [0, \ldots, d]^m, \text{ and } \boldsymbol{\mu}^T \boldsymbol{A} \in (d_L \cdot \mathbb{Z})^n. \quad (78)$$

Notice here that $\boldsymbol{\mu}^T \boldsymbol{d}/d_L$ is a rational number with denominator d_L. There are two cases: either $\boldsymbol{\mu}^T \boldsymbol{d}/d_L$ is an integer, or $\boldsymbol{\mu}^T \boldsymbol{d}/d_L$ misses the nearest integer by at least $1/d_L$. Therefore $\lfloor \boldsymbol{\mu}^T \boldsymbol{d}/d_L \rfloor$ is the only integer in the interval

$$\left[\frac{\boldsymbol{\mu}^T \boldsymbol{d} - d_L + 1}{d_L}, \frac{\boldsymbol{\mu}^T \boldsymbol{d}}{d_L} \right].$$

These observations enable us to construct a polytope Q, whose integer points will correspond to the inequalities (78). Let Q be the set of all $(\boldsymbol{\mu}, \boldsymbol{y}, z)$ in $\mathbb{R}^{2n+1}$ satisfying the inequalities

$$\begin{array}{rcl} \boldsymbol{\mu} & \geq & \mathbf{0} \\ \mu_i & \leq & d_L, \quad i = 1, \ldots, n \\ \boldsymbol{\mu}^T \boldsymbol{A} & = & d_L \, \boldsymbol{y}^T \\ (\boldsymbol{\mu}^T \boldsymbol{d}) - d_L + 1 & \leq & d_L \, z \\ (\boldsymbol{\mu}^T \boldsymbol{d}) & \geq & d_L \, z. \end{array} \quad (79)$$

If $(\boldsymbol{\mu}, \boldsymbol{y}, z)$ is integral, then $\boldsymbol{\mu} \in [0, \ldots, d]^m$, $\boldsymbol{y} \in \mathbb{Z}^n$ enforces $\boldsymbol{\mu}^T \boldsymbol{A} \in (d_L \cdot \mathbb{Z})^n$ and z is the only integer in the interval $[(\boldsymbol{\mu}^T \boldsymbol{d} + 1 - d_L)/d_L,\ \boldsymbol{\mu}^T \boldsymbol{d}/d_L]$. It is not hard to see that Q is indeed a polytope. We call Q the *cutting plane polytope* of the simplicial cone $P(\boldsymbol{A}, \boldsymbol{d})$

The correspondence between inequalities (their syntactic representation) in (78) and integer points in the cutting plane polytope Q is obvious. We now show that the facets of P' are among the vertices of Q_I.

Proposition 11 ([15]). *Each facet of P' is represented by an integer vertex of Q_I.*

Proof. Consider a facet $\boldsymbol{c}^T \boldsymbol{x} \leq \delta$ of P'. If we remove this inequality (possibly several times, because of scalar multiples) from the set of inequalities in (78), then the polyhedron defined by the resulting set of inequalities differs from P', since P' is full-dimensional. Thus there exists a point $\hat{\boldsymbol{x}} \in \mathbb{Q}^n$ that is violated by $\boldsymbol{c}^T \boldsymbol{x} \leq \delta$, but satisfies any other inequality in (78) (see Figure 12). Consider the following integer program:

$$\max\{(\boldsymbol{\mu}^T \boldsymbol{A}/d_L)\, \hat{\boldsymbol{x}} - z \mid (\boldsymbol{\mu}, \boldsymbol{y}, z) \in Q_I\}. \quad (80)$$

Since $\hat{\boldsymbol{x}} \notin P'$ there exists an inequality $(\boldsymbol{\mu}^T \boldsymbol{A}/d_L)\boldsymbol{x} \leq \lfloor \boldsymbol{\mu}^T \boldsymbol{d}/d_L \rfloor$ in (78) with

$$(\boldsymbol{\mu}^T \boldsymbol{A}/d_L)\hat{\boldsymbol{x}} - \lfloor \boldsymbol{\mu}^T \boldsymbol{d}/d_L \rfloor > 0.$$

Therefore, the optimal value will be strictly positive, and an integer optimal solution $(\boldsymbol{\mu}, \boldsymbol{y}, z)$ must correspond to the facet $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ of P'. Since the optimum of the integer linear program (80) is attained at a vertex of Q_I, the assertion follows. □

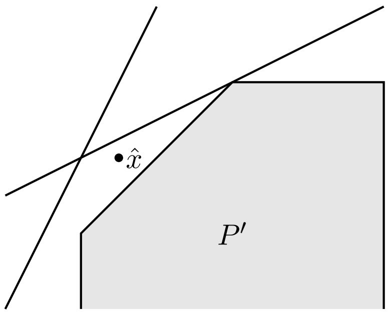


Figure 12: The point $\hat{\boldsymbol{x}}$ lies "above" the facet $\boldsymbol{c}^T\boldsymbol{x} \leq \delta$ and "below" each other inequality in (78).

Not each vertex of Q_I represents a facet of P'. In particular, if P is defined by nonnegative inequalities only, then $\boldsymbol{0}$ is a vertex of Q_I but not a facet of P'.

Lemma 9 ([15]). *The elementary closure of a rational simplicial cone* $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$*, where* $\boldsymbol{A}$ *and* $\boldsymbol{d}$ *are integral and* $\boldsymbol{A}$ *has full row rank, is polynomially bounded in* size(P) *when the dimension is fixed.*

Proof. Each facet of P' corresponds to a vertex of Q_I by Proposition 11. Recall from the Hadamard bound that $d_L \leq \|\boldsymbol{a}_1\| \cdots \|\boldsymbol{a}_n\|$, where $\boldsymbol{a}_i$ are the columns of $\boldsymbol{A}$. Thus the number of bits needed to encode d_L is in $\mathrm{O}(n\,\mathrm{size}(P))$. Therefore the size of Q is in $\mathrm{O}(n\,\mathrm{size}(P))$. It follows from Theorem 25 that the number of vertices of Q_I is in $\mathrm{O}(\mathrm{size}(P)^n)$ for fixed n, since the dimension of Q is $n+1$. □

It is possible to explicitly construct, in polynomial time, a minimal inequality system defining P' when the dimension is fixed.

Observe first that the lattice determinant d_L in (79) can be computed with some polynomial Hermite normal form algorithm. If H is the HNF of $\boldsymbol{A}$, then $L(\boldsymbol{A}) = L(H)$ and the determinant of H is simply the product of its diagonal elements. Notice then that the system (79) can be written down. In particular its size is polynomial in the size of $\boldsymbol{A}$ and $\boldsymbol{d}$, even in varying dimension, which follows from the Hadamard bound.

As noted in [28], one can construct the vertices of Q_I in polynomial time. This works as follows. Suppose one has a list of vertices $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ of Q_I. Let Q_k denote the convex hull of these vertices. Find an inequality description of Q_k, $\boldsymbol{C}\boldsymbol{x} \leq \boldsymbol{d}$. For each row-vector $\boldsymbol{c}_i$ of $\boldsymbol{C}$, find with Lenstra's algorithm a vertex of Q_I maximizing $\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in Q_I\}$. If new vertices are found, add them to the list and repeat the preceding steps, otherwise the list of vertices is complete. The list of vertices of Q_I yields a list of inequalities defining P'. With the ellipsoid method or your favorite linear programming algorithm in fixed dimension, one can decide for each individual inequality, whether it is necessary. If not, remove it. What remains are the facets of P'.

Proposition 12. *There exists an algorithm which, given a matrix $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ of full row rank and a vector $\boldsymbol{d} \in \mathbb{Z}^m$, constructs the elementary closure P' of $P(\boldsymbol{A}, \boldsymbol{d})$ in polynomial time when the dimension n is fixed.*

7.3.2 Rational polyhedra

Let $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}\}$, with integer $\boldsymbol{A}$ and $\boldsymbol{d}$, be a rational polyhedron. Any Gomory-Chvátal cut can be derived from a set of $\mathrm{rank}(\boldsymbol{A})$ inequalities out of $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}$ where the corresponding rows of $\boldsymbol{A}$ are linear independent. Such a choice represents a simplicial cone C and it follows from Theorem 9 that the number of inequalities of C' is polynomially bounded by $\mathrm{size}(C) \leq \mathrm{size}(P)$.

Theorem 29 ([15]). *The number of inequalities needed to describe the elementary closure of a rational polyhedron $P = P(\boldsymbol{A}, \boldsymbol{d})$ with $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ and $\boldsymbol{d} \in \mathbb{Z}^m$, is polynomial in $\mathrm{size}(P)$ in fixed dimension.*

Following the discussion at the end of Section 7.3.1 and using again Lenstra's algorithm, it is now easy to come up with a polynomial algorithm for constructing the elementary closure of a rational polyhedron $P(\boldsymbol{A}, \boldsymbol{d})$ in fixed dimension. For each choice of $\mathrm{rank}(\boldsymbol{A})$ rows of $\boldsymbol{A}$ defining a simplicial cone C, compute the elementary closure C' and put the corresponding inequalities in the partial list of inequalities describing P'. At the end, redundant inequalities can be deleted.

Theorem 30. *There exists a polynomial algorithm that, given a matrix* $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ *and a vector* $\boldsymbol{d} \in \mathbb{Z}^m$*, constructs an inequality description of the elementary closure of* $P(\boldsymbol{A}, \boldsymbol{d})$*.*

References

[1] K. Aardal, R. E. Bixby, C. A. J. Hurkens, A. K. Lenstra, and J. W. Smeltink. Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances. *INFORMS Journal on Computing*, 12(3):192–202, 2000.

[2] K. Aardal, C. Hurkens, and A. K. Lenstra. Solving a linear diophantine equation with lower and upper bounds on the variables. In R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, editors, *Integer Programming and Combinatorial Optimization, 6th International IPCO Conference*, volume 1412 of *Lecture Notes in Computer Science*, pages 229–242, Berlin, 1998. Springer-Verlag.

[3] K. Aardal, C. A. J. Hurkens, and A. K. Lenstra. Solving a system of linear Diophantine equations with lower and upper bounds on the variables. *Mathematics of Operations Research*, 25(3):427–442, 2000.

[4] K. Aardal and A. K. Lenstra. Hard equality constrained integer knapsacks. *Mathematics of Operations Research.* To appear.

[5] K. Aardal, R. Weismantel, and L. A. Wolsey. Non-standard approaches to integer programming. *Discrete Applied Mathematics*, 123(1-3):5–74, 2002.

[6] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, 1974.

[7] M. Ajtai. The shortest vector problem in *L2* is NP-hard for randomized reductions. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 10–19, New York, 1998. ACM Press.

[8] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd Annual ACM symposium on Theory of Computing*, pages 601–610, New York, 2001. ACM Press.

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

[10] I. Bárány, R. Howe, and L. Lovász. On integer points in polyhedra: A lower bound. *Combinatorica*, 12(2):135–142, 1992.

[11] A. Barvinok and J. E. Pommersheim. An algorithmic theory of lattice points in polyhedra. *New Perspectives in Algebraic Combinatorics, MSRI Publications*, 38:91–147, 1999.

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

[13] D. E. Bell. A theorem concerning the integer lattice. *Studies in Applied Mathematics*, 56(2):187–188, 1976/77.

[14] J. Blömer. Closest vectors, successive minima, and dual HKZ-bases of lattices. In *Proceedings of the 17th ICALP*, volume 1853 of *Lecture Notes in Computer Science*, pages 248–259, Berlin, 2000. Springer-Verlag.

[15] A. Bockmayr and F. Eisenbrand. Cutting planes and the elementary closure in fixed dimension. *Mathematics of Operations Research*, 26(2):304–312, 2001.

[16] A. Bockmayr, F. Eisenbrand, M. E. Hartmann, and A. S. Schulz. On the Chvátal rank of polytopes in the 0/1 cube. *Discrete Applied Mathematics*, 98:21–27, 1999.

[17] I. Borosh and L. B. Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proceedings of the American Mathematical Society*, 55:299–304, 1976.

[18] J. Bourgain and V. D. Milman. Sections Euclidiennes et volume des corps symétriques convexes dans $\mathbf{R}^n$. *Comptes Rendus de l'Académie des Sciences. Série I. Mathématique*, 300(13):435–438, 1985.

[19] M. Brion. Points entiers dans polyèdres convexes. *Annales Scientifiques de l'École Normale Supérieure*, 21(4):653–663, 1988.

[20] J.-Y. Cai. Some recent progress on the complexity of lattice problems. *Electronic Colloquium on Computational Complexity*, (6), 1999. ECCC is available at:
`http://www.eccc.uni-trier.de/eccc/`.

[21] J.-Y. Cai and A. P. Nerurkar. Approximating the svp to within a factor $(1+1/dim^{\varepsilon})$ is NP-hard under randomized reductions. In *Proceedings of the 38th IEEE Conference on Computational Complexity*, pages 46–55, Pittsburgh, 1998. IEEE Computer Society Press.

[22] J. W. S. Cassels. *An Introduction to the Geometry of Numbers*. Classics in Mathematics. Springer-Verlag, Berlin, 1997. Second Printing, Corrected, Reprint of the 1971 ed.

[23] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[24] V. Chvátal, W. Cook, and M. Hartmann. On cutting-plane proofs in combinatorial optimization. *Linear Algebra and its Applications*, 114/115:455–499, 1989.

[25] K. L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *Journal of the Association for Computing Machinery*, 42:488–499, 1995.

[26] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, 1971. ACM Press.

[27] W. Cook, C. R. Coullard, and G. Turán. On the complexity of cutting plane proofs. *Discrete Applied Mathematics*, 18:25–38, 1987.

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

[29] W. Cook, T. Rutherford, H. E. Scarf, and D. Shallcross. An implementation of the generalized basis reduction algorithm for integer programming. *ORSA Journal on Computing*, 5(2):206–212, 1993.

[30] W. J. Cook and S. Dash. On the matrix-cut rank of polyhedra. *Mathematics of Operations Research*, 26(1):19–30, 2001.

[31] G. Cornuéjols and M. Dawande. A class of hard small 0-1 programs. In R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, editors, *Integer Programming and Combinatorial Optimization, 6th International IPCO Conference*, volume 1412 of *Lecture Notes in Computer Science*, pages 284–293, Berlin, 1998. Springer-Verlag.

[32] G. Cornuéjols and Y. Li. On the rank of mixed 0,1 polyhedra. *Mathematical Programming*, 91(2):391–397, 2002.

[33] G. Cornuéjols, R. Urbaniak, R. Weismantel, and L. Wolsey. Decomposition of integer programs and of generating sets. In R. Burkard and G. Woeginger, editors, *Algorithms—ESA '97*, volume 1284 of *Lecture Notes in Computer Science*, pages 92–103. Springer-Verlag, Berlin, 1997.

[34] M. J. Coster, A. Joux, B. A. LaMacchia, A. M. Odlyzko, C.-P. Schnorr, and J. Stern. Improved low-density subset sum algorithms. *Computational Complexity*, 2(2):111–128, 1992.

[35] S. Dash. An exponential lower bound on the length of some classes of branch-and-cut proofs. In W. J. Cook and A. S. Shulz, editors, *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference*, volume 2337 of *Lecture Notes in Computer Science*, Berlin, 2002. Springer-Verlag. To appear.

[36] J. A. De Loera, R. Hemmecke, J. Tauzer, and R. Yoshida. Effective lattice point counting in rational polytopes. *Journal of Symbolic Computation*. To appear. Available at:
`http://www.math.ucdavis.edu/~deloera`.

[37] M. E. Dyer. On integer points in polyhedra. *SIAM Journal on Computing*, 20:695–707, 1991.

[38] M. E. Dyer and R. Kannan. On Barvinok's algorithm for counting lattice points in fixed dimension. *Mathematics of Operations Research*, 22(3):545–549, 1997.

[39] F. Eisenbrand. Short vectors of planar lattices via continued fractions. *Information Processing Letters*, 79(3):121–126, 2001.

[40] F. Eisenbrand. Fast integer programming in fixed dimension. In G. D. Battista and U. Zwick, editors, *Algorithms – ESA 2003*, volume 2832 of *Lecture Notes in Computer Science*, pages 196–207, Berlin, 2003. Springer-Verlag.

[41] F. Eisenbrand and S. Laue. A linear algorithm for integer programming in the plane. *Mathematical Programming*, 2004. to appear.

[42] F. Eisenbrand and G. Rote. Fast 2-variable integer programming. In K. Aardal and B. Gerards, editors, *Integer Programming and Combinatorial Optimization, 8th International IPCO Conference*, volume 2081 of *Lecture Notes in Computer Science*, pages 78–89, Berlin, 2001. Springer-Verlag.

[43] F. Eisenbrand and G. Rote. Fast reduction of ternary quadratic forms. In J. Silverman, editor, *Cryptography and Lattices, International Conference, CaLC 2001*, volume 2146 of *Lecture Notes in Computer Science*, pages 32–44, Berlin, 2001. Springer-Verlag.

[44] F. Eisenbrand and A. S. Schulz. Bounds on the Chvátal rank of polytopes in the 0/1 cube. In G. Cornuéjols, R. E. Burkard, and G. J. Woeginger, editors, *Integer Programming and Combinatorial Optimization, 7th International IPCO Conference*, volume 1610 of *Lecture Notes in Computer Science*, pages 137–150. Springer-Verlag, 1999.

[45] P. van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report MI-UvA-81-04, Mathematical Institute, University of Amsterdam, Amsterdam, 1981.

[46] S. D. Feit. A fast algorithm for the two-variable integer programming problem. *Journal of the Association for Computing Machinery*, 31(1):99–113, 1984.

[47] L. Gao and Y. Zhang. Computational experience with lenstra's algorithm. Technical Report TR02-12, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2002.

[48] B. Gärtner and E. Welzl. Linear programming—randomization and abstract frameworks. In *STACS 96*, volume 1046 of *Lecture Notes in Computer Science*, pages 669–687, Berlin, 1996. Springer-Verlag.

[49] C. F. Gauß. *Disquisitiones arithmeticae*. Gerh. Fleischer Iun., 1801.

[50] J.-L. Goffin. Variable metric relaxation methods. II. The ellipsoid method. *Mathematical Programming*, 30(2):147–162, 1984.

[51] O. Goldreich and S. Goldwasser. On the limits of non-approximability of lattice problems. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 1–9, New York, 1998. ACM Press.

[52] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

[53] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

[54] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

[55] M. Grötschel, L. Lovász, and A. Schrijver. Geometric methods in combinatorial optimization. In W. R. Pulleyblank, editor, *Progress in Combinatorial Optimization*, pages 167–183. Academic Press, Toronto, 1984.

[56] A. C. Hayes and D. G. Larman. The vertices of the knapsack polytope. *Discrete Applied Mathematics*, 6:135–138, 1983.

[57] B. Helfrich. Algorithms to construct Minkowski reduced and Hermite reduced lattice basis. *Theoretical Computer Science*, 41:125–139, 1985.

[58] C. Hermite. Extraits de lettres de M. Ch. Hermite à M. Jacobi sur différents objets de la théorie des nombres. *Journal für die reine und angewandte Mathematik*, 40, 1850.

[59] C. Hermite. Deuxième lettre à Jacobi. In *Oevres de Hermite I*, pages 122–135. Gauthier-Villary, Paris, 1905.

[60] D. S. Hirschberg and C. K. Wong. A polynomial algorithm for the knapsack problem in two variables. *Journal of the Association for Computing Machinery*, 23(1):147–154, 1976.

[61] A. Joux and J. Stern. Lattice reduction: a toolbox for the cryptanalyst. *Journal of Cryptology*, 11(3):161–185, 1998.

[62] N. Kanamaru, T. Nishizeki, and T. Asano. Efficient enumeration of grid points in a convex polygon and its application to integer programming. *International Journal of Computational Geometry & Applications*, 4(1):69–85, 1994.

[63] R. Kannan. A polynomial algorithm for the two-variable integer programming problem. *Journal of the Association for Computing Machinery*, 27(1):118–122, 1980.

[64] R. Kannan. Improved algorithms for integer programming and related problems. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 193–206, New York, 1983. ACM Press.

[65] R. Kannan. Algorithmic geometry of numbers. *Annual Review of Computer Science*, 2:231–267, 1987.

[66] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

[67] R. Kannan and L. Lovász. Covering minima and lattice point free convex bodies. In *Foundations of Software Technology and Theoretical Computer Science*, volume 241 of *Lecture Notes in Computer Science*, pages 193–213. Springer-Verlag, Berlin, 1986.

[68] R. Kannan and L. Lovász. Covering minima and lattice-point-free convex bodies. *Annals of Mathematics*, 128:577–602, 1988.

[69] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum Press, New York, 1972.

[70] A. Khinchine. A quantitative formulation of Kronecker's theory of approximation (in russian). *Izvestiya Akademii Nauk SSR Seriya Matematika*, 12:113–122, 1948.

[71] D. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley, Reading, 1969.

[72] A. Korkine and G. Zolotareff. Sur les formes quadratiques. *Mahematische Annalen*, 6:366–389, 1873.

[73] J. C. Lagarias, H. W. Lenstra, Jr., and C. P. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.

[74] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the Association for Computing Machinery*, 32(1):229–246, 1985.

[75] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515 – 534, 1982.

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

[77] LiDIA – a Library for Computational Number Theory. TH Darmstadt/Universität des Saarlandes, Fachbereich Informatik, Institut für Theoretische Informatik.
`http://www.informatik.th-darmstadt.de/pub/TI/LiDIA`.

[78] Q. Louveaux and L. A. Wolsey. Combining problem structure with basis reduction to solve a class of hard integer programs. *Mathematics of Operations Research*, 27(3):470–484, 2002.

[79] L. Lovász and H. E. Scarf. The generalized basis reduction algorithm. *Mathematics of Operations Research*, 17(3):751 – 764, 1992.

[80] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498–516, 1996.

[81] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 92–98, Los Alamitos, CA, 1998. IEEE Computer Society.

[82] H. Minkowski. Über die positiven quadratischen Formen und über kettenbruchänliche Algorithmen. *Journal für die reine und angewandte Mathematik*, 107:278–297, 1891.

[83] H. Minkowski. *Geometrie der Zahlen.* Teubner, Leipzig, 1896.

[84] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, Cambridge, 1995.

[85] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, New York, 1988.

[86] P. Q. Nguyen and J. Stern. Lattice reduction in cryptology: An update. In W. Bosma, editor, *Algorithmic Number Theory, 4th International Symposium, ANTS-IV*, volume 1838 of *Lecture Notes in Computer Science*, pages 85–112, Berlin, 2000. Springer-Verlag.

[87] P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In J. H. Silverman, editor, *Cryptography and Lattices, International Conference, CaLC 2001*, volume 2146 of *Lecture Notes in Computer Science*, pages 146–180, Berlin, 2001. Springer-Verlag.

[88] P. Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–988, 1997.

[89] H. E. Scarf. An observation on the structure of production sets with indivisibilities. *Proceedings of the National Academy of Sciences, U.S.A.*, 74(9):3637–3641, 1977.

[90] H. E. Scarf. Production sets with indivisibilities. Part I: generalities. *Econometrica*, 49:1–32, 1981.

[91] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2-3):201–224, 1987.

[92] C.-P. Schnorr. Block reduced lattice bases and successive minima. *Combinatorics Probability and Computing*, 3(4):507–522, 1994.

[93] C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(2):181–199, 1994.

[94] C. P. Schnorr and H. H. Hörner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Advances in Cryptology—EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, Berlin, 1995.

[95] A. Schönhage. Schnelle Berechnung von Kettenbruchentwicklungen. (Speedy computation of expansions of continued fractions). *Acta Informatica*, 1:139–144, 1971.

[96] A. Schönhage. Fast reduction and composition of binary quadratic forms. In *International Symposium on Symbolic and Algebraic Computation, ISSAC 91*, pages 128–133, New York, 1991. ACM Press.

[97] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen (Fast multiplication of large numbers). *Computing*, 7:281–292, 1971.

[98] A. Schrijver. On cutting planes. *Annals of Discrete Mathematics*, 9:291–296, 1980.

[99] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, Chichester, 1986.

[100] I. Semaev. A 3-dimensional lattice reduction algorithm. In J. H. Silverman, editor, *Cryptography and Lattices, International Conference, CaLC 2001*, volume 2146 of *Lecture Notes in Computer Science*, pages 181–193, Berlin, 2001. Springer-Verlag.

[101] M. Seysen. Simultaneous reduction of a lattice basis and its reciprocal basis. *Combinatorica*, 13(3):363–376, 1993.

[102] V. Shoup. NTL: A Library for doing Number Theory, Courant Institute, New York.
`http://www.shoup.net/`.

[103] O. van Sprang. *Basisreduktionsalgorithmen für Gitter kleiner Dimension.* PhD thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1994. in german.

[104] X. Wang. *A New Implementation of the Generalized Basis Reduction Algorithm for Convex Integer Programming.* PhD thesis, Yale University, 1997.

[105] C. K. Yap. Fast unimodular reduction: Planar integer lattices. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 437–446, Pittsburgh, 1992. IEEE Computer Society Press.

[106] L. Y. Zamanskij and V. D. Cherkasskij. A formula for determining the number of integral points on a straight line and its application. *Ehkon. Mat. Metody*, 20:1132–1138, 1984.