Non-Standard Approaches to Integer Programming *

Karen Aardal † Robert Weismantel ‡ Laurence Wolsey

Introduction

The standard approach to the integer programming optimization problem

$$(IP) \quad \min\{c^T x : x \in X\} \text{ where } X = \{x \in \mathbb{Z}^n : Ax = b\}$$

or the equivalent integer programming feasibility problem

$$(FP) \quad \text{Is } X \neq \emptyset?$$

is to use linear programming within a branch-and-bound or branch-and-cut framework, using whenever possible polyhedral results about the structure of $\mathrm{conv}(X)$ or approximations to $\mathrm{conv}(X)$. Here we examine alternative approaches depending explicitly on the *discrete* nature of the set X.

Given a specific point $x^0 \in X$ and a generic point $x \in X$, the vector $y = x - x^0$ lies in

$$L = \{x \in \mathbb{Z}^n : Ax = 0\},$$

the set of integer points in a subspace. Every such set can be shown to form an *integer lattice*, namely it can be rewritten in the form

$$L = \{x : x = B\lambda, \lambda \in \mathbb{Z}^p\}.$$

In Section 1 we introduce lattices and the basis reduction algorithms of Lovász [62] and of Lovász & Scarf [68] . Every n-dimensional lattice can be generated by n linearly independent vectors, called the *lattice basis.* A *reduced* basis is a basis in which the vectors are short and nearly orthogonal. The basis reduction algorithm of Lovász runs in polynomial time and produces basis vectors of short Euclidean length. The algorithm of Lovász and Scarf works with a more general norm and runs in polynomial time for fixed n.

Lattice basis reduction has played an important role in the theory of integer programming. It was first introduced by H. W. Lenstra, Jr. in 1983 [63] who proved that the integer

*Work carried out as part of DONET (Discrete Optimization Network) TMR project nr. ERB FMRX-CT98-0202 of the EU

†Research partially supported by the ESPRIT Long Term Research Project nr. 20244 (Project ALCOM-IT: *Algorithms and Complexity in Information Technology* of the EU and by NSF through the Center for Research on Parallel Computation, Rice University, under Cooperative Agreement No. CCR-9120008.

‡Supported by a Gerhard-Hess-Forschungsförderpreis and grant WE1462 of the German Science Foundation (DFG), and grants FKZ 0037KD0099 and FKZ 2945A/0028G of the Kultusministerium Sachsen-Anhalt.

programming problem can be solved in polynomial time for a fixed number of variables. The proof was algorithmic and consisted of two main steps: a linear transformation, and Lovász' basis reduction algorithm [62]. Later, Grötschel, Lovász, & Schrijver [44], Kannan [56], and Lovász & Scarf [68] developed algorithms using similar principles to Lenstra's algorithm. In computational integer programming, however, basis reduction has received less attention. One of the few implementations that we are aware of is reported on by Cook, Rutherford, Scarf, & Shallcross [20] in which some difficult, not previously solved, network design problems were solved using the generalized basis reduction algorithm of Lovász and Scarf. Recently Aardal, Hurkens, & Lenstra [2], [3] developed an algorithm for solving a system of diophantine equations with bounds on the variables. They used basis reduction to reformulate a certain integer relaxation of the problem, and were able to solve several integer programming instances that proved hard, or even unsolvable, for several other algorithms. Their algorithm was partly inspired by algorithms used in cryptography to solve subset sum problems that occur in knapsack public-key cryptosystems. In the area of cryptography, basis reduction has been used successfully to solve such subset sum problems, see for instance the survey article by Joux & Stern [52]. These lattice based algorithms are presented in Section 2.

Alternatively given a point $x^0 \in X$, suppose that there exists a point $x \in X$ having a smaller objective value $c^T x < c^T x^0$, and also satisfying the condition $x \geq x^0$. Now $y = x - x^0$ lies in the set

$$X^0 = \{x \in \mathbb{Z}^n : Ax = 0, x \geq 0\},$$

the set of non-negative integer points in a cone. Here again such sets can be finitely generated, and rewritten in the form

$$X^0 = \{x : x = H\lambda, \lambda \in \mathbb{Z}^p_+\},$$

where the minimal set of columns H is known as a *Hilbert basis*. Note that it follows that there is some column h of H for which h is an improving vector for x^0, i.e. $x' = x^0 + h \in X$ with $x' \geq x^0$ and $c^T x' > c^T x^0$.

Generalizing this idea leads to test sets for families of integer programs. Test sets are collections of integral vectors with the property that every feasible non-optimal point of any integer program in the family can be improved by a vector in the test set. Given a test set T explicitly, there is a straightforward algorithm for solving (IP). Starting with a feasible point x, one searches for an element $t \in T$ such that $x + t$ is feasible and has a smaller objective function value, replaces x by $x + t$ and iterates until no such t exists. In general one cannot expect that a test set is polynomial in the size of a given integer program. This raises the question of designing an efficient augmentation algorithm:

Let x be any feasible point of the linear integer program. While x is not optimal, determine an integral vector z and a non-negative integer λ such that (i) $x + \lambda z$ is feasible and (ii) $x + \lambda z$ attains a smaller objective function value than x. Set $x := x + \lambda z$.

Augmentation algorithms have been designed for and applied to a range of linear integer programming problems: augmenting path methods for solving maximum flow problems or algorithms for solving the minimum cost flow problem via augmentation along negative cycles are of this type. Other examples include the greedy algorithm for solving the matroid optimization problem, alternating path algorithms for solving the maximum weighted matching problem or primal methods for optimizing over the intersection of two matroids. In Section 3 we investigate these primal approaches to solving (IP).

We next consider a family of relaxations for (IP) in which we drop the nonnegativity constraints on a subset of the variables, namely relaxations of the form:

$$\min\{c^T x : Ax = b, x_j \in \mathbb{Z}^1_+ \text{ for } j \in V \setminus S, x_j \in \mathbb{Z}^1 \text{ for } j \in S\}.$$

Note that when $S = V$, this leads us back to the lattice viewpoint.

Gomory [35, 36, 37] in the 1960's studied the "asymptotic group" relaxation in which $S = A_B$ is an optimal linear programming basis. The resulting solution set $X^G = \{x = (x_B, x_N) \in \mathbb{Z}^n : A_B x_B + A_N x_N = b, x_N \geq 0\}$ can be reformulated in the space of the nonbasic variables as the $\tilde{X}^G = \{x_N \in \mathbb{Z}^{n-m}_+ : A_B^{-1} A_N x_N \equiv A_B^{-1} b \pmod 1\}$. Optimization over $\tilde{X}^G$ reduces to a shortest path problem in a graph with $|\det(A_B)|$ nodes, known as the *group problem*. Gomory's study of the convex hull of solutions of $\tilde{X}^G$, known as the *corner polyhedron*, showed up the crucial role of subaditivity in describing strong valid inequalities. These developments are discussed in Section 4.

Notation

For $z \in \mathbb{R}^n$, $||z||$, $||z||_1$ and $||z||_\infty$ denote the Euclidean, L_1 and maximum norms of z respectively. x^T denotes the transpose of the vector x such that $x^T y$ is the inner product on $\mathbb{R}^n$ of the vectors x and y. The symbol e^j represents the unit vector in the corresponding Euclidean space having a one in position j and zeros everywhere else.

For $x \in \mathbb{R}$, $\lfloor x \rfloor$ is the largest integer not greater than x, $\lceil x \rceil$ is the smallest integer not less than x, and $\lceil a \rfloor = \lceil a - \frac{1}{2} \rceil$, i.e., the nearest integer to a, where we round up if the fraction is equal to one half.

If $\lambda_1, \ldots, \lambda_n$ are integers, $\gcd(\lambda_1, \ldots, \lambda_n)$ denotes the greatest common divisor of these integers. If $x, y \in \mathbb{Z}^1_+ \setminus \{0\}$, $x|y$ means that y is an integer multiple of x. A square integral matrix C is *unimodular* if $|\det C| = 1$.

Finally for $v \in \mathbb{R}^n$ we denote by $\operatorname{supp}(v) := \{i \in \{1, \ldots, n\} : v_i \neq 0\}$ the support of v. v^+ is the vector such that $v_i^+ = v_i$ if $v_i > 0$ and $v_i^+ = 0$, otherwise. Similarlily, v^- is the vector with $v_i^- = -v_i$ if $v_i < 0$ and $v_i^- = 0$, otherwise. So $v = v^+ - v^-$.

1 Lattices and Basis Reduction

Here we define a lattice and a lattice basis. In the lattice approaches to integer programming that will be discussed in Section 2 we need lattice representations using bases with short, almost orthogonal basis vectors. Such bases are called reduced. We describe two algorithms for finding a reduced basis. The algorithm of Lovász, as presented by Lenstra, Lenstra, & Lovász [62], is described in Subsection 1.1, and the algorithm of Lovász & Scarf [68] is presented in Subsection 1.2. We also discuss some recent implementations.

Note that in Sections 1 and 2 we exceptionally use b_j, b_j^*, b_j' to denote distinct vectors associated with the basis of a lattice, and not the j^{th} coefficient of the vectors b, b^* and b' respectively.

1.1 Lovász' basis reduction algorithm

Given a set of l linearly independent vectors $b_1, \ldots, b_l \in \mathbb{R}^n$ with $l \leq n$, let B be the matrix with column vectors $b_1, \ldots, b_l$.

Definition 1.1. *The* lattice L *spanned by* $b_1, \ldots, b_l$ *is the set of vectors that can be obtained by taking integer linear combinations of the vectors* $b_1, \ldots, b_l$*,*

$$L = \{x : x = \sum_{j=1}^{l} \lambda_j b_j,\ \lambda_j \in \mathbb{Z},\ 1 \leq j \leq l\}. \tag{1}$$

The set of vectors $b_1, \ldots, b_l$ *is called a* basis *of the lattice.*

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

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

Theorem 1.1. *An integral matrix* U *is unimodular if and only if* U *can be derived from the identity matrix by elementary column operations.*

A lattice may have several bases.

Observation 1.1. *If* B *and* B' *are bases for the same lattice* L*, then* $B' = BU$ *for some* $l \times l$ *unimodular matrix* U*.*

Lovász' basis reduction algorithm [62] consists of a series of elementary column operations on an initial basis B for a given lattice and produces a so-called *reduced basis* B' such that the basis vectors $b_1', \ldots, b_l'$ are short and nearly orthogonal, and such that b_1' is an approximation of the shortest vector in the lattice. So, B' is obtained as $B' = BU$ for some unimodular matrix U. Given a basis B one can obtain orthogonal vectors by applying Gram-Schmidt orthogonalization. The Gram-Schmidt vectors, however, do not necessarily belong to the lattice, but they do span the same real vector space as $b_1, \ldots, b_l$, so they are used as a "reference" for the basis reduction algorithm.

Definition 1.2. *The Gram-Schmidt process derives orthogonal vectors b_j^*, $1 \leq j \leq l$, from linearly independent vectors b_j, $1 \leq j \leq l$. The vectors b_j^*, $1 \leq j \leq l$, and the real numbers μ_{jk}, $1 \leq k < j \leq l$, are determined from b_j, $1 \leq j \leq l$, by the recursion*

$$b_1^* = b_1 \tag{2}$$

$$b_j^* = b_j - \sum_{k=1}^{j-1} \mu_{jk} b_k^*, \quad 2 \leq j \leq n \tag{3}$$

$$\mu_{jk} = \frac{b_j^T b_k^*}{||b_k^*||^2}, \quad 1 \leq k < j \leq l. \tag{4}$$

Example 1.1. Here we illustrate the Gram-Schmidt vectors obtained by applying the orthogonalization procedure given in Definition 1.2 to the vectors

$$b_1 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}, \quad b_2 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, \quad b_3 = \begin{pmatrix} -1 \\ 1 \\ 2 \end{pmatrix}$$

shown in Figure 1 a.

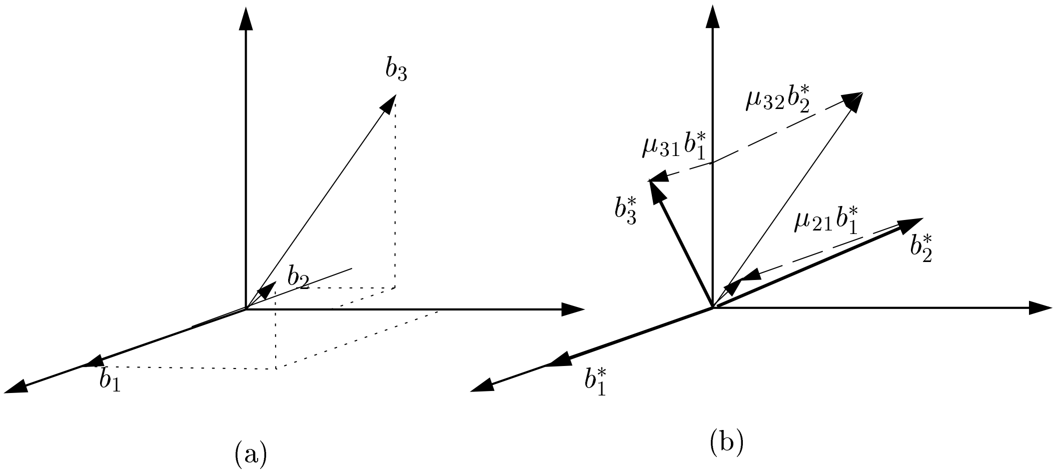

Figure 1:

We obtain $\mu_{21} = 1,\ \mu_{31} = -\frac{1}{2},\ \mu_{32} = \frac{4}{5}$, and

$$b_1^* = b_1 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}, \quad b_2^* = b_2 - \mu_{21} b_1^* = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}, \quad b_3^* = b_3 - \mu_{31} b_1^* - \mu_{32} b_2^* = \begin{pmatrix} 0 \\ -\frac{3}{5} \\ \frac{6}{5} \end{pmatrix}.$$

The Gram-Schmidt vectors are shown in Figure 1 b. $\square$

As mentioned above, the vectors $b_1^*, \ldots, b_j^*$, span the same real vector space as the vectors $b_1, \ldots, b_j$, $1 \le j \le n$. The vector b_j^* is the projection of b_j on the orthogonal complement of $\sum_{k=1}^{j-1} \mathbb{R} b_k$, i.e., b_j^* is the component of b_j orthogonal to the real subspace spanned by $b_1, \ldots, b_{j-1}$. Thus, any pair b_i^*, b_k^* of the Gram-Schmidt vectors are mutually orthogonal. The multiplier μ_{jk} gives the length, relative to b_k^*, of the component of the vector b_j in direction b_k^*. The multiplier μ_{jk} is equal to zero if and only if b_j is orthogonal to b_k^*.

Definition 1.3. *[62]. A basis $b_1, b_2, \ldots, b_l$ is called* reduced *if*

$$|\mu_{jk}| \le \frac{1}{2} \quad \textit{for } 1 \le k < j \le l, \tag{5}$$

$$||b_j^* + \mu_{j,j-1} b_{j-1}^*||^2 \ge \frac{3}{4} ||b_{j-1}^*||^2 \quad \textit{for } 1 < j \le l. \tag{6}$$

A reduced basis according to Lovász is a basis in which the vectors are short and nearly orthogonal. Below we explain why vectors satisfying Conditions (5) and (6) have these characteristics.

Condition (5) is satisfied if the component of vector b_j in direction b_k^* is short relative to b_k^*. This is the case if b_j and b_k^* are nearly orthogonal, or if b_j is short relative to b_k^*. If condition (5) is violated, i.e., the component of vector b_j in direction b_k^* is *relatively long*, then Lovász' basis reduction algorithm will replace b_j by $b_j - \lceil \mu_{jk} \rfloor b_k$. Such a step is called *size reduction* and will ensure relatively short basis vectors. Next, suppose that (5) is satisfied because b_j is short relative to b_k^*, $k < j$. Then we may end up with a basis where the vectors are not at all orthogonal, and where the first vector is very long, the next one relatively short compared to the first one, and so on. To prevent this from happening we enforce Condition (6). Here we relate to the interpretation of the Gram-Schmidt vectors above, and notice that the vectors $b_j^* + \mu_{j,j-1} b_{j-1}^*$ and b_{j-1}^* are the projections of b_j and b_{j-1} on the orthogonal complement of $\sum_{k=1}^{j-2} \mathbb{R} b_k$. Consider the case where $k = j - 1$, i.e., suppose that b_j is short compared to b_{j-1}^*, which implies that b_j^* is short compared to b_{j-1}^* as $||b_j^*|| \le ||b_j||$. Suppose we *interchange* b_j and b_{j-1}. Then the new b_{j-1}^* will be the vector $b_j^* + \mu_{j,j-1} b_{j-1}^*$, which will be short compared to the old b_{j-1}^*, i.e., Condition (6) will be violated. To summarize, Conditions (5) and (6) ensure that we obtain a basis in which the vectors are short and nearly orthogonal. To achieve such a basis, Lovász' algorithm applies a sequence of *size reductions* and *interchanges* in order to reduce the length of the vectors, and to prevent us from obtaining non-orthogonal basis vectors of decreasing length, where the first basis vector may be arbitrarily long. The constant $\frac{3}{4}$ in inequality (6) 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.

A brief outline of Lovász' basis reduction algorithm is as follows. For precise details we refer to [62]. First compute the Gram-Schmidt vectors b_j^*, $1 \le j \le l$ and the numbers μ_{jk}, $1 \le k < j \le l$. Initialize $i := 2$. Perform, if necessary, a size reduction to obtain $|\mu_{i,i-1}| \le 1/2$. Update $\mu_{i,i-1}$. Then check whether Condition (6) holds for $j = i$. If Condition (6) is violated, then exchange b_i and b_{i-1}, and update the relevant Gram-Schmidt vectors and numbers μ_{jk}. If $i > 2$, then let $i := i - 1$. Next, achieve $|\mu_{im}| \le 1/2$ for $m = i - 2, i - 3, \ldots, 1$. If $i = n$, stop. Otherwise, let $i := i + 1$.

From this short description, it is not obvious that the algorithm is efficient, but as the following theorem states, Lovász' basis reduction algorithm runs in polynomial time.

Theorem 1.2. *[62]. Let $L \subseteq \mathbb{Z}^n$ be a lattice with basis $b_1, \ldots, b_n$, and let $\beta \in \mathbb{R}$, $\beta \geq 2$, be such that $||b_j||^2 \leq \beta$ for $1 \leq j \leq n$. Then the number of arithmetic operations needed by the basis reduction algorithm as described in [62] 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 1.2 implies that Lovász' basis reduction algorithm has a running time of $O(n^6 (\log \beta)^3)$ using classical algorithms for addition and multiplication. There are reasons to believe that it is possible in practice to find a reduced basis in $O(n (\log \beta)^3)$ bit operations, see Section 4 of Kaltofen [53], and Odlyzko [73].

Example 1.2. 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

$$b_1 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \quad b_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The Gram-Schmidt vectors are $b_1^* = b_1$ and $b_2^* = b_2 - \mu_{21} b_1^* = (1,1)^T - \frac{1}{17} b_1^* = \frac{1}{17}(-3, 12)^T$, see Figure 2 a. Condition (5) is satisfied since b_2 is short relative to b_1^*. However, Condition (6) is violated, so we exchange b_1 and b_2, giving

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad b_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}.$$

We now have $b_1^* = b_1$, $\mu_{21} = \frac{5}{2}$ and $b_2^* = \frac{1}{2}(3, -3)^T$, see Figure 2 b.

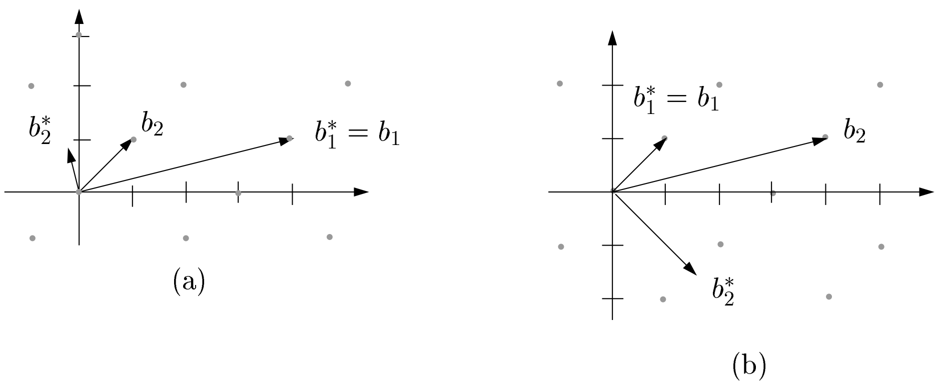

Figure 2:

Condition (5) is now violated, so we replace b_2 by $b_2 - 2b_1 = (2, -1)^T$. Conditions (5) and (6) are satisfied for the resulting basis

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad b_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix},$$

and hence this basis is reduced, see Figure 3.

□

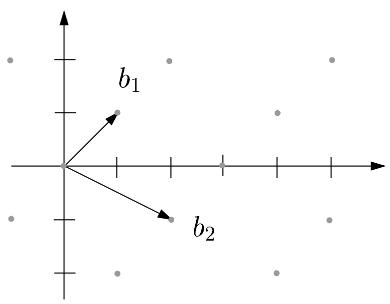

Figure 3: The reduced basis.

Let W be the vector space spanned by the lattice L, and let B_W be an orthonomal basis for W. The determinant of the lattice L, $\det(L)$, is defined as the absolute value of the determinant of any nonsingular mapping $W \to W$ that maps B_W on a basis of L. Below we give three different formulae for computing $\det(L)$. Let $B = (b_1, \dots, b_m)$ be a basis for the lattice $L \subset \mathbb{R}^n$, with $m \leq n$, and let $b_1^*, \dots b_m^*$ be the vectors obtained from applying the Gram-Schmidt orthogonalization procedure, see Definition 1.2, to $b_1, \dots, b_m$.

$$\det(L) = ||b_1^*|| \cdot ||b_2^*|| \cdot \; \cdots \; \cdot ||b_m^*||, \tag{7}$$

$$\det(L) = \sqrt{\det(b_i^T b_j)_{i,j}} \;, \tag{8}$$

$$\det(L) = \lim_{r \to \infty} \frac{|\{x \in L : ||x|| < r\}|}{\mathrm{vol}(B_m(r))}, \tag{9}$$

where $\mathrm{vol}(B_m(r))$ is the volume of the m-dimensional ball with radius r. If L is full-dimensional, $\det(L)$ can be interpreted as the volume of the parallelepiped $\sum_{j=1}^n [0,1) b_j$. In this case the determinant of the lattice can be computed straightforwardly as $\det(L) = |\det(b_1, \dots, b_n)|$. Note that the determinant of a lattice depends only on the lattice and not on the choice of basis (cf. Observation 1.1, and expression (9)). The determinant of $\mathbb{Z}^n$ is equal to one.

In Propositions 1.3 and 1.4 below we assume that the lattice L is full-dimensional.

Proposition 1.3. *[62]. Let $b_1, \dots, b_n$ be a reduced basis for the lattice $L \subset \mathbb{R}^n$. Then,*

$$\det(L) \leq \Pi_{j=1}^n ||b_j|| \leq c_1 \cdot \det(L), \tag{10}$$

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

The first inequality in (10) is the so called *inequality of Hadamard* that holds for any basis of L. Hadamard's inequality holds with equality if and only if the basis is orthogonal. Hermite [48] proved that each lattice $L \subset \mathbb{R}^n$ has a basis $b_1, \dots, b_n$ such that $\Pi_{j=1}^n ||b_j|| \leq c \cdot \det(L)$, where c is a constant depending only on n. The basis produced by Lovász' basis reduction algorithm yields the constant $c = c_1$ in Proposition 1.3. 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 1.3 is that if we consider a basis that satisfies (10), then the distance of the basis vector b_n to the hyperplane generated by the reduced basis vectors $b_1, \ldots, b_{n-1}$ is not too small as stated in the following Corollary.

Corollary 1.1. *[63]. Assume that $b_1, \ldots, b_n$ is a basis such that (10) holds, and that, after possible reordering, $||b_n|| = \max_{1 \le j \le n}\{||b_j||\}$. Let $H = \sum_{j=1}^{n-1} \mathbb{R} b_j$ and let h be the distance of basis vector b_n to H. Then*

$$c_1^{-1} \cdot ||b_n|| \le h \le ||b_n||, \tag{11}$$

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

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

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

Expressions (10) and (12) give

$$\Pi_{j=1}^{n} ||b_j|| \le c_1 \cdot \det(L) = c_1 \cdot h \cdot \det(L') \le c_1 \cdot h \cdot \Pi_{j=1}^{n-1} ||b_j||, \tag{13}$$

where the first inequality follows from the second inequality of (10), and where the last inequality follows from the inequality of Hadamard (first inequality of (10)). From (13) we obtain $h \ge c_1^{-1} ||b_n||$. From the definition of h we have $h \le ||b_n||$, and this bound holds with equality if and only if the vector b_n is perpendicular to H. □

The lower bound on h given in Corollary 1.1 plays a crucial role in the algorithm of H. W. Lenstra, Jr., that is described in Section 2.1.1.

Proposition 1.4. *[62]. Let $L \subset \mathbb{R}^n$ be a lattice with reduced basis $b_1, \ldots, b_n \in \mathbb{R}^n$. Let $x^1, \ldots, x^t \in L$ be linearly independent. Then we have*

$$||b_1||^2 \le 2^{n-1} ||x||^2 \quad \textit{for all } x \in L, x \ne 0, \tag{14}$$

$$||b_j||^2 \le 2^{n-1} \max\{||x^1||^2, ||x^2||^2, \ldots, ||x^t||^2\} \quad \textit{for } 1 \le j \le t. \tag{15}$$

Inequality (14) implies that the first reduced basis vector b_1 is an approximation of the shortest nonzero vector in L. Kannan [56] presents an algorithm based on Lovász' basis reduction algorithm that computes the shortest nonzero lattice vector in polynomial time for fixed n. It is not known whether the problem of finding the shortest nonzero vector in a given lattice is NP-hard. Micciancio [70] showed that computing the approximate length of the shortest vector in a lattice within a factor less than $\sqrt{2}$ is NP-hard for randomized problem transformations. In his proof he used a randomized transformation from a variant of the so-called *closest vector problem* to the shortest vector problem. The closest vector problem is defined as follows. Given n linearly independent vectors $b_1, \ldots, b_n \in \mathbb{Q}^n$, and a further vector $q \in \mathbb{Q}^n$, find a vector x in the lattice generated by $b_1, \ldots, b_n$ with $||q - x||$ minimal. Van Emde Boas [96] showed that finding the shortest vector with respect to the maximum norm in a given lattice is NP-hard, and that the closest vector problem is NP-hard for any norm. 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 1.5. *[62]. Let $\nu_1, \ldots, \nu_l$ denote the successive minima of $\|\ \|$ on L, and let $b_1, \ldots, b_l$ be a reduced basis for L. Then*

$$2^{(1-j)/2}\nu_j \leq \|b_j\| \leq 2^{(l-1)/2}\nu_j \quad \text{for } 1 \leq j \leq l. \tag{16}$$

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 & Euchner [81] for a more detailed overview. Schnorr [79] extended Lovász' algorithm to a family of polynomial time algorithms that, given $\varepsilon > 0$, finds a non-zero vector in an n-dimensional lattice that is no longer than $(1+\varepsilon)^n$ 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 [87] 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 & Euchner [81] 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 tends to become unstable. They propose a floating point version with good stability, but cannot prove that the algorithm always terminates. Empirical studies indicate 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 [79], [80] also developed a variant of Lovász' algorithm in which not only two vectors are interchanged during the reduction process, but where blocks $b_j, b_{j+1}, \ldots, b_{j+\beta-1}$ of β consecutive vectors are transformed so as to minimize the j^{th} Gram Schmidt vector b_j^*. This so called block reduction produces shorter basis vectors but needs more computing time. The shortest vector b_j^* in a block of size β is determined by complete enumeration of all short lattice vectors. Schnorr & Hörner [82] 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 [64], developed at TH Darmstadt, and NTL - A Library for doing Number Theory [88], developed by V. Shoup, IBM, Zürich.

1.2 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 $C \in \mathbb{R}^n$ that is symmetric about the origin gives rise to a norm $F(c) = \inf\{t \geq 0 : c/t \in C\}$. Lovász & Scarf [68] 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 $C^* = \{y : y^T c \leq 1 \text{ for all } c \in C\}$. We also define a distance function associated with a projection of C. Let $b_1, \ldots, b_n$ be a basis for $\mathbb{Z}^n$, and let C_j be the projection of C on the orthogonal complement of $b_1, \ldots, b_{j-1}$. We have that $c = \beta_j b_j + \cdots + \beta_n b_n \in C_j$ if and only if there exist $\alpha_1, \ldots, \alpha_{j-1}$ such that $c + \alpha_1 b_1 + \cdots + \alpha_{j-1} b_{j-1} \in C$. The distance function associated with C_j is defined as:

$$F_j(c) = \min_{\alpha_1,\ldots,\alpha_{j-1}} F(c + \alpha_1 b_1 + \cdots + \alpha_{j-1} b_{j-1}). \tag{17}$$

Using duality, one can show that expression (17) is equivalent to the following problem:

$$F_j(c) = \max\{c^T z : z \in C^*,\ b_1^T z = 0, \ldots, b_{j-1}^T z = 0\}. \tag{18}$$

In expression (18), note that only vectors z that are orthogonal to the basis vectors $b_1, \ldots, b_{j-1}$ 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 (18) is a linear program, which can be solved in polynomial time. The distance function F has the following properties:

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

Lovász and Scarf use the following definition of a reduced basis.

Definition 1.4. *A basis $b_1, \ldots, b_n$ is called* reduced *if*

$$F_j(b_{j+1} + \mu b_j) \geq F_j(b_{j+1}) \quad \text{for } 1 \leq j \leq n-1 \text{ and all integers } \mu, \tag{19}$$

$$F_j(b_{j+1}) \geq (1-\varepsilon) F_j(b_j) \quad \text{for } 1 \leq j \leq n-1 \tag{20}$$

where ε satisfies $0 < \varepsilon < \frac{1}{2}$.

Definition 1.5. *A basis $b_1, \ldots, b_n$, not necessarily reduced, is called* proper *if*

$$F_k(b_j + \mu b_k) \geq F_k(b_j) \quad \text{for } 1 \leq k < j \leq n. \tag{21}$$

Remark 1.1. 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 reduced basis according to Lenstra, Lenstra, & Lovász [62] (cf. Definition 1.3).

An important question is how to check whether Condition (19) is satisfied for all integers μ. Here we make use of the dual relationship between formulations (17) and (18). We have the following equality: $\min_{\alpha \in \mathbb{R}} F_j(b_{j+1} + \alpha b_j) = F_{j+1}(b_{j+1})$. Let α^* denote the optimal α in the minimization. The function F_j is convex, and hence the integer μ that minimizes $F_j(b_{j+1} + \mu b_j)$ 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 $b_j^T z = 0$, which implies that the integral μ that minimizes $F_j(b_{j+1} + \mu b_j)$ can be determined by solving two additional linear programs, unless α^* is integral.

Condition (21) is analogous to Condition (5) of Lovász' basis reduction algorithm, and is violated if adding an integer multiple of b_k to b_j yields a distance function value $F_k(b_j + \mu b_k)$ that is smaller than $F_k(b_j)$. In the generalized basis reduction algorithm we only check whether the condition is satisfied for $k = j - 1$ (cf. Condition (19)), and we use the value of μ that minimizes $F_j(b_{j+1} + \mu b_j)$ as mentioned above. If Condition (19) is violated we do a *size reduction*, i.e., we replace b_{j+1} by $b_{j+1} + \mu b_j$.

Condition (20) corresponds to Condition (6) in Lovász' algorithm, and ensures that the basis vectors are in the order of increasing distance function value, aside from the factor $(1 - \varepsilon)$. Recall that we want the first basis vector to be an approximation of the shortest lattice vector. If Condition (20) is violated we interchange vectors b_j and b_{j+1}.

The algorithm works as follows. Let $b_1, \ldots, b_n$ be an initial basis for $\mathbb{Z}^n$. Typically $b_j = e^j$, where e^j is the j^{th} column of the identity matrix. Let j be the first index for which Conditions (19) or (20) are not satisfied. If (19) is violated, we replace b_{j+1} by $b_{j+1} + \mu b_j$ with the appropriate value of μ. If Condition (20) is satisfied after the replacement, we let $j := j + 1$. If Condition (20) is violated, we interchange b_j and 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 1.6. *[68]. Let ε be chosen as in (20), let $\gamma = 2 + 1/\log(1/(1 - \varepsilon))$, and let $B(R)$ be a ball with radius R containing C. Moreover, let $U = \max_{1 \leq j \leq n}\{F_j(b_j)\}$, where $b_1, \ldots, b_n$ 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

$$\left(\frac{\gamma^n - 1}{\gamma - 1}\right)\left(\frac{\log(U/V)}{\log(1/(1-\varepsilon))}\right). \tag{22}$$

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 1.2).

We now give a few properties of a Lovász-Scarf reduced basis. If one can obtain a basis $b_1, \ldots, b_n$, given C, such that $F_1(b_1) \leq F_2(b_2) \leq \cdots \leq F_n(b_n)$, then one can prove that b_1 is the shortest integral 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 1.7. *[68]. Let $0 < \varepsilon < \frac{1}{2}$, and let $b_1, \ldots, b_n$ be a Lovász-Scarf reduced basis. Then*

$$F_{j+1}(b_{j+1}) \geq (\frac{1}{2} - \varepsilon)F_j(b_j) \quad \textit{for } 1 \leq j \leq n - 1. \tag{23}$$

We can use this theorem to obtain a result analogous to (14) of Proposition 1.4.

Proposition 1.8. *[68]. Let $0 < \varepsilon < \frac{1}{2}$, and let $b_1, \ldots, b_n$ be a Lovász-Scarf reduced basis. Then*

$$F(b_1) \leq (\frac{1}{2} - \varepsilon)^{1-n} F(x) \quad \text{for all } x \in \mathbb{Z}^n, x \neq 0. \tag{24}$$

We can also relate the distance function $F_j(b_j)$ to the j^{th} *successive minimum* of F on the lattice $\mathbb{Z}^n$ (cf. Proposition 1.5). $\nu_1, \ldots, \nu_n$ are the *successive minima* of F on $\mathbb{Z}^n$ if there are vectors $x^1, \ldots, x^n \in \mathbb{Z}^n$ with $\nu_j = F(x^j)$, such that for each $1 \leq j \leq n$, x^j is the shortest lattice vector (with respect to F) that is linearly independent of $x^1, \ldots, x^{j-1}$.

Proposition 1.9. *Let $\nu_1, \ldots, \nu_n$ denote the successive minima of F on the lattice $\mathbb{Z}^n$, let $0 < \varepsilon < \frac{1}{2}$, and let $b_1, \ldots, b_n$ be a Lovász-Scarf reduced basis. Then*

$$(\frac{1}{2} - \varepsilon)^{j-1} \nu_j \leq F_j(b_j) \leq (\frac{1}{2} - \varepsilon)^{j-n} \nu_j \quad \text{for } 1 \leq j \leq n. \tag{25}$$

The first reduced basis vector is an approximation of the shortest lattice vector (Proposition 1.8). 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 2.1.3. To find the shortest lattice vector we proceed as follows. If the basis $b_1, \ldots, b_n$ is Lovász-Scarf reduced, we can obtain a bound on the coordinates of lattice vectors c that satisfy $F_1(c) \leq F_1(b_1)$. We express the vector c as an integer linear combination of the basis vectors, i.e., $c = \lambda_1 b_1 + \cdots + \lambda_n b_n$, where $\lambda_j \in \mathbb{Z}$. We have

$$F_1(b_1) \geq F_1(c) \geq F_n(c) = F_n(\lambda_n b_n) = |\lambda_n| F_n(b_n), \tag{26}$$

where the second inequality holds since $F_n(c)$ is more constrained than $F_1(c)$, the first equality holds due to the constraints $b_i^T z = 0,\ 1 \leq i \leq n-1$, and the second equality holds as $F(tx) = tF(x)$ for $t > 0$. We can now use (26) to obtain the following bound on $|\lambda_n|$:

$$|\lambda_n| \leq \frac{F_1(b_1)}{F_n(b_n)} \leq \frac{1}{(\frac{1}{2} - \varepsilon)^{n-1}}, \tag{27}$$

where the last inequality is obtained by applying Theorem 1.7 iteratively. Notice that the bound on λ_n is polynomial 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 $F_j(\lambda_n b_n + \cdots + \lambda_{j+1} b_{j+1} + \gamma b_j)$. If this minimum is greater than $F_1(b_1)$, then there does not exist a vector c, with $\lambda_n, \ldots, \lambda_{j+1}$ fixed such that $F_1(c) \leq F_1(b_1)$, since in that case $F_1(b_1) < F_j(\lambda_n b_n + \cdots + \lambda_{j+1} b_{j+1} + \gamma^* b_j) \leq F_j(\lambda_n b_n + \cdots + \lambda_j b_j) = F_j(c) \leq F_1(c)$, which yields a contradiction. If the minimum is less than or equal to $F_1(b_1)$, then we can obtain the bound:

$$|\lambda_j - \gamma^*| \leq 2\frac{F_1(b_1)}{F_j(b_j)} \leq \frac{2}{(\frac{1}{2} - \varepsilon)^{j-1}}. \tag{28}$$

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 number of nodes that is polynomial if n is fixed.

The generalized basis reduction algorithm was implemented by Cook, Rutherford, Scarf, & Shallcross [20], and by Wang [99]. Cook et al. used generalized basis reduction to derive a heuristic version of the integer programming algorithm by Lovász and Scarf (see Section 2.1.3) to solve difficult integer network design instances. Wang solved both linear and nonlinear integer programming problems using the generalized basis reduction algorithm as a subroutine.

An example illustrating a few iterations of the generalized basis reduction algorithm is given in Section 2.1.3.

2 Basis Reduction in Integer Programming

The main ideas behind the integer programming algorithms by Lenstra [63], Grötschel, Lovász, & Schrijver [44], Kannan [56], and Lovász & Scarf [68] described in Subsection 2.1 are as follows. A lattice is contained in countably many parallel hyperplanes. If one wants to decide whether or not a certain polyhedron contains an integral vector, then one can enumerate some of these lattice hyperplanes. To avoid an unnecessarily large enumeration tree one wants to find a representation of the lattice hyperplanes such that the distance between them is not too small. In particular, for given dimension n one should only need to enumerate a polynomial number of hyperplanes. To find a suitable representation of the lattice, basis reduction is used.

The use of basis reduction in cryptography will be briefly discussed in Subsection 2.2 since several interesting theoretical and computational results have been obtained in this area using basis reduction, and since the lattices and the bases that have been used in attacking knapsack cryptosystems are related to the lattice used by Aardal, Hurkens, & Lenstra [2], [3]. Their algorithm is outlined in Subsection 2.3. The basic idea behind the algorithms discussed in Subsections 2.2 and 2.3 is to reformulate the problem as a problem of finding a short vector in a certain lattice. One therefore needs to construct a lattice in which any feasible vector is provably short.

For the reader wishing to study this topic in more detail we refer to the articles mentioned in this introduction, to the survey article by Kannan [55], and to the textbooks by Lovász [67], Schrijver [83], Grötschel, Lovász, & Schrijver [43], Nemhauser & Wolsey [71], and Cohen [14]. In these references, and in the article by Lenstra, Lenstra, & Lovász [62], several applications of basis reduction are mentioned, other than integer programming, such as finding a short nonzero vector in a lattice, finding the Hermite normal form of a matrix, simultaneous diophantine approximation, factoring polynomials with rational coefficients, and finding $\mathbb{Q}$-linear relations among real numbers $\alpha_1, \alpha_1, \ldots, \alpha_n$. Reviewing these other topics is outside the scope of our section.

2.1 Integer programming in fixed dimension

Let A be a rational $m \times n$-matrix and let d be a rational m-vector. We consider the integer programming problem in the following form:

$$\text{Does there exist an integral vector } x \text{ such that } Ax \leq d? \tag{29}$$

Karp [59] showed that the zero-one integer programming problem is NP-complete, and Borosh & Treybig proved that the integer programming problem (29) belongs to NP. Combining these results implies that (29) is NP-complete. The NP-completeness of the zero-one version is a fairly straightforward consequence of the proof by Cook [16] that the Satisfiability problem is NP-complete. An important open question was still: Can the integer programming 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 & Wong [49], and by Kannan [54]. Scarf [77] showed that (29), for the general case $n = 2$, is polynomially solvable. Both Hirschberg & Wong, and Scarf conjectured that the integer programming problem could be solved in polynomial time if the dimension is fixed. The proof of this conjecture was given by H. W. Lenstra, Jr. [63]. Below we first illustrate in Example 2.1 why linear programming based branch-and-bound is not a polynomial algorithm for $n = 2$. Next we describe three algorithms for solving the integer programming problem in fixed dimension: Lenstra's algorithm [63] and the algorithm of Grötschel, Lovász, & Schrijver [44], which are both based on Lovász' basis reduction algorithm [62], and, finally, the algorithm of Lovász & Scarf [68], which is based on the generalized basis reduction algorithm.

It is worthwhile pointing out here that Barvinok [5] showed that there exists a polynomial time algorithm for *counting* the number of integral points in a polyhedron if the dimension is fixed. Barvinok's result therefore generalizes the result of Lenstra. Barvinok, however, based his algorithm on an identity by Brion for exponential sums over polytopes. Later, Dyer & Kannan [27] developed a simpler algorithm for counting the number of integral points in fixed dimension. Their algorithm uses only elementary properties of exponential sums. To describe Barvinok's result and the improvement by Dyer and Kannan is outside the scope of this chapter.

Example 2.1. Consider the 2-dimensional polytope in Figure 4. If we use branch-and-

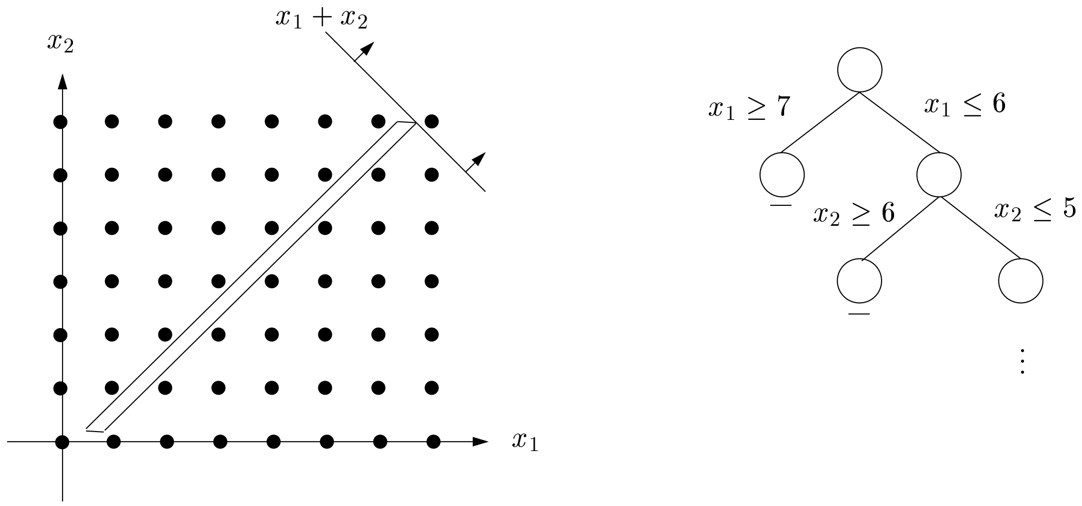

Figure 4: A difficult type of instance for branch-and-bound

bound on this instance with objective function max $x_1 + x_2$, then we see that the variables

x_1 and x_2 alternately take fractional values, which forces us to branch. If we extend the polytope arbitrarily far, then the branch-and-bound tree will become arbitrarily deep. It is easy to construct an example that is equally bad for branch-and-bound in which the polytope contains an integer vector. $\square$

2.1.1 Lenstra's algorithm

We pose the integer programming problem in a slightly different way from (29). Let $X = \{x \in \mathbb{R}^n : Ax \leq d\}$. The question we consider is:

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

An observation made by Lenstra was that "thin" polytopes as in Example 2.1 were "bad" from the worst-case perspective. He therefore 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 $B(p, z) = \{x \in \mathbb{R}^n : ||x - p|| \leq z\}$ be the closed ball with center p and radius z. The transformation τ that we apply to the polytope is constructed such that $B(p, r) \subset \tau X \subset B(p, R)$ for some $p \in \tau X$ and such that

$$\frac{R}{r} \leq c_2, \tag{31}$$

where c_2 is a constant that depends only on the dimension n. Relation (31) is the measure of "roundness" that Lenstra uses. For an illustration, see Figure 5. Once we have transformed

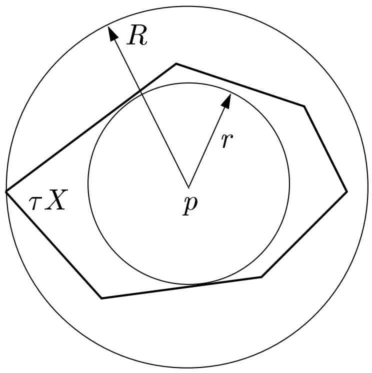

Figure 5:

the polytope, we need to apply the same transformation to the lattice, which gives us the following problem:

$$\text{Is } \tau\mathbb{Z}^n \cap \tau X \neq \emptyset? \tag{32}$$

Note that problems (30) and (32) are equivalent. The vectors τe^j, $1 \leq j \leq n$, where e^j is the j^{th} column of the identity matrix, 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 τe^j, $1 \leq j \leq n$ 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 2.1. *[63]. Let $b_1, \ldots, b_n$ be any basis for L. Then for all $x \in \mathbb{R}^n$ there exists a vector $y \in L$ such that*

$$||x - y||^2 \leq \frac{1}{4}(||b_1||^2 + \cdots + ||b_n||^2). \tag{33}$$

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

Next, let $L = \tau\mathbb{Z}^n$, and let $b_1, \ldots, b_n$ be a basis for L such that (10) holds. Notice that (10) holds if the basis is reduced. Also, reorder the vectors such that $||b_n|| = \max_{1 \leq j \leq n}\{||b_j||\}$. Let $x = p$ where p is the center of the closed balls $B(p, r)$ and $B(p, R)$. Apply Lemma 2.1 to the given x. This gives a lattice vector $y \in \tau\mathbb{Z}^n$ such that

$$||p - y||^2 \leq \frac{1}{4}(||b_1||^2 + \cdots + ||b_n||^2) \leq \frac{1}{4} \cdot n \cdot ||b_n||^2 \tag{34}$$

in polynomial time. We now distinguish two cases. Either $y \in \tau X$ or $y \notin \tau X$. The first case implies that τX is relatively large, and if we are in this case, then we are done, so we assume we are in the second case. Since $y \notin \tau X$ we know that y is not inside the ball $B(p, r)$ as $B(p, r)$ is completely contained in τX. Hence we know that $||p - y|| > r$, or using (34), that

$$r < \frac{1}{2} \cdot \sqrt{n} \cdot ||b_n||. \tag{35}$$

We now create t subproblems by considering intersections between the polytope τX with t parallel hyperplanes containing the lattice L. Each of these 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 at most n deep, and the number of nodes at each level is polynomially 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 1.1 and its proof. We can write L as

$$L = L' + \mathbb{Z}b_n \subset H + \mathbb{Z}b_n = \cup_{k \in \mathbb{Z}}(H + kb_n). \tag{36}$$

So the lattice L is contained in countably many parallel hyperplanes. For an example we refer to Figure 6. The distance between two consecutive hyperplanes is h, and Corollary 1.1 says that h is bounded from below by a constant depending only on n, which implies that not too many hyperplanes intersect τX. To determine precisely how many hyperplanes intersect τX, we approximate τX by the ball $B(p, R)$. If t is the number of hyperplanes intersecting $B(p, R)$ we have

$$t - 1 \leq \frac{2R}{h}. \tag{37}$$

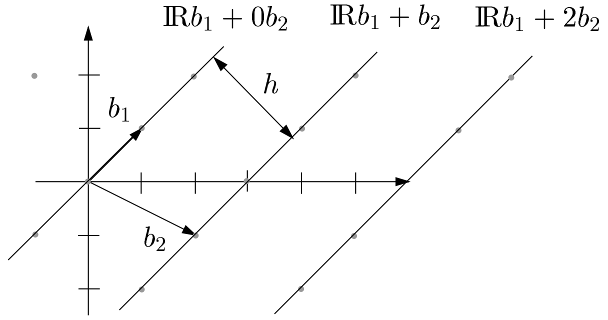

Figure 6:

Using the relationship (31) between the radii R and r we have $2R \leq 2rc_2 < c_2\sqrt{n}||b_n||$, where the last inequality follows from (35). Since $h \geq c_1^{-1}||b_n||$ (cf. Corollary 1.1), we get the following bound on the number of hyperplanes that we need to consider:

$$t - 1 \leq \frac{2R}{h} < c_1 c_2 \sqrt{n}, \tag{38}$$

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 [63] discusses ways of improving these values. To determine the values of k in expression (36), we express p as a linear combination of the basis vectors $b_1, \ldots, b_n$. Recall that p is the center of the ball $B(p, R)$ that was used to approximate τX.

So far we have not mentioned how to determine the transformation τ and hence the balls $B(p, r)$ and $B(p, R)$. We give the general idea here without going into detail. First, determine an n-simplex contained in X. This can be done by repeated calls to the ellipsoid algorithm. The resulting simplex is described by its extreme points $v_0, \ldots, v_n$. By applying the ellipsoid algorithm repeatedly we can decide whether there exists an extreme point x of X such that if we replace v_j by 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 [63] 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 [63] shows that τ has the property that if we let $p = 1/(n+1)\sum_{j=0}^{n} e^j$, where e^j is the j^{th} column of the identity matrix (i.e., p is the center of the regular simplex), then there exists closed balls $B(p, r)$ and $B(p, R)$ such that $B(p, r) \subset \tau X \subset B(p, R)$ for some $p \in \tau X$, and such that $R/r \leq c_2$.

Kannan [56] 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 poytope X and obtained a polytope τX such that $B(p, r) \subset \tau X \subset B(p, R)$ for some $p \in \tau X$. 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 in a different sense compared to 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 $O(n^{5/2})$. As far as we know, no implementation of Lenstra's or Kannan's algorithms has been reported on in the literature.

2.1.2 The algorithm of Grötschel, Lovász, and Schrijver

Grötschel, Lovász, & Schrijver [44] used ellipsoidal approximations of the feasible set X and derived an algorithm based on the same principles as Lenstra's algorithm. Here we will give a sketch of their approach. Assume without loss of generality that $X = \{x \in \mathbb{R}^n : Ax \leq d\}$ is bounded and full-dimensional. The key idea is to rapidly find a vector $y \in \mathbb{Z}^n$, as Lenstra does through Lemma 2.1, and if y does not belong to X, to find a nonzero integral direction c such that the width of the polytope X in this direction is bounded by a constant depending only on n. This is expressed in the following theorem.

Theorem 2.2. *[44]. Let $Ax \leq d$ be a system of m rational inequalities in n variables, and let $X = \{x : Ax \leq d\}$. There exists a polynomial algorithm that finds either an integral vector $y \in X$, or a vector $c \in \mathbb{Z}^n \setminus 0$ such that*

$$\max\{c^T x : x \in X\} - \min\{c^T x : x \in X\} \leq 2n(n+1)2^{n(n-1)/4} \tag{39}$$

Remark 2.1. Grötschel, Lovász, and Schrijver in fact gave the polytope $\{x : Ax \leq d\}$ in terms of a separation oracle, and not by an explicit description. This gives rise to a slightly more involved proof. Here we follow the presentation of Schrijver [83]. Notice that the algorithm referred to in Theorem 2.2 is polynomial for *arbitrary* n.

Here we will not make a transformation to a lattice $\tau\mathbb{Z}^n$, but remain in the lattice $\mathbb{Z}^n$. The first step is to find two ellipsoids; one contained in X, and one containing X. Let D be a positive semidefinite $n \times n$-matrix, and let $p \in \mathbb{R}^n$. The *ellipsoid* associated with p and D is defined as $E(p, D) = \{x \in \mathbb{R}^n : (x-p)^T D^{-1}(x-p) \leq 1\}$. The vector p is called the *center* of the ellipsoid $E(p, D)$. Goffin [34] showed that it is possible to find ellipsoids $E(p, (1/(n+1)^2)D)$, $E(p, D)$ in polynomial time such that

$$E(p, \frac{1}{(n+1)^2}D) \subseteq X \subseteq E(p, D). \tag{40}$$

Next, we apply basis reduction, but instead of using the Euclidean norm to measure the length of the basis vectors, as described in Section 1.1, we use a norm defined by the positive definite matrix D^{-1} describing the ellipsoids, see Schrijver [83] Chapters 6 and 18. The norm $//\ //$ defined by the matrix D^{-1} is given by $//x// = \sqrt{x^T D^{-1} x}$. Given a positive definite rational matrix D^{-1}, we can apply basis reduction to the unit basis to obtain a basis $b_1, \ldots, b_n$ for the lattice $\mathbb{Z}^n$ in polynomial time that satisfies (cf. the second inequality of (10))

$$\Pi_{j=1}^{n} //b_j// \leq 2^{n(n-1)/4}\sqrt{\det(D^{-1})}. \tag{41}$$

Next, reorder the basis vectors such that $//b_n// = \max_{1\le j\le n}\{//b_j//\}$. After reordering, inequality (41) still holds. Suppose that the vector $y \in \mathbb{Z}^n$, which can be found by applying Lemma 2.1 with $x = p$, does not belong to X. We then have that $y \notin E(p, (1/(n+1)^2)D)$ as this ellipsoid is contained in X, which implies that $//p-y// > 1/(n+1)$. Using (34) we obtain $1/2 \cdot \sqrt{n} \cdot //b_n// \ge //p-y// > 1/(n+1)$ which gives the following bound on the length of the n^{th} basis vector:

$$//b_n// > \frac{2}{\sqrt{n}(n+1)} > \frac{1}{n(n+1)}. \tag{42}$$

Choose a direction c such that the components of c are relatively prime integers, and such that c is orthogonal to the subspace generated by the basis vectors $b_1, \ldots, b_{n-1}$. One can show, see Schrijver [83] pp 257–258, that if we consider a vector z such that $z^T D^{-1} z \le 1$, then

$$|c^T z| \le \sqrt{\det(D)}//b_1// \cdot \; \cdots \; \cdot //b_{n-1}// \le 2^{n(n-1)/4}//b_n//^{-1} < n(n+1)2^{n(n-1)/4}, \tag{43}$$

where the second inequality follows from inequality (41), and the last inequality follows from (42). If a vector z satisfies $z^T D^{-1} z \le 1$, then $z \in E(p, D)$, which implies that $|c^T(z-p)| \le n(n+1)2^{n(n-1)/4}$. We then obtain

$$\begin{aligned} &\max\{c^T x : x \in X\} - \min\{c^T x : x \in X\} \\ &\le \max\{c^T x : x \in E(p,D)\} - \min\{c^T x : x \in E(p,D)\} \le 2n(n+1)2^{n(n-1)/4}, \end{aligned} \tag{44}$$

which gives the desired result.

Lenstra's result that the integer programming problem can be solved in polynomial time for fixed n follows from Theorem 2.2. If we apply the algorithm implied by Theorem 2.2, we either find an integral point $y \in X$ or a thin direction c. Assume that the direction c is the outcome of the algorithm. Let $\mu = \lceil \min\{c^T x : x \in X\} \rceil$. All points in $X \cap \mathbb{Z}^n$ are contained in the parallel hyperplanes $c^T x = t$ where $t = \mu, \ldots, \mu + 2n(n+1)2^{n(n-1)/4}$, so, if n is fixed we get polynomially many hyperplanes, each giving rise to a subproblem of dimension less than or equal to $n-1$: does there exist an integral vector $x \in \{X : c^T x = t\}$? For each of these lower-dimensional problems we repeat the algorithm of Theorem 2.2. The search tree has at most n levels and each level has polynomially many nodes if the dimension is fixed.

2.1.3 The algorithm of Lovász and Scarf

The integer programming algorithm of Lovász & Scarf [68] determines, in polynomial time for fixed n, whether there exists a thin direction for the polytope X. If X is not thin in any direction, then X has to contain an integral vector. 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 polynomially bounded for fixed n. If the algorithm indicates that X contains an integral vector, then one needs to determine a so-called Korkine-Zolotarev 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 by Grötschel, Lovász, and Schrijver. Again, we assume that $X = \{x \in \mathbb{R}^n : Ax \le d\}$ is bounded, rational, and full-dimensional.

Definition 2.1. *The* width *of the polytope X in the nonzero direction c is determined as*

$$\max\{c^T x : \ x \in X\} - \min\{c^T x : \ x \in X\} = \max\{c^T(x-y) : x \in X, \ y \in X\}. \quad (45)$$

Let $(X - X) = \{(x-y) : x \in X, \ y \in X)\}$ 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:

$$\begin{aligned} F_j(c) &= \min_{\alpha_1,\ldots,\alpha_{j-1}\in\mathbb{Q}} F(c + \alpha_1 b_1 + \cdots + \alpha_{j-1} b_{j-1}) && (46)\\ &= \max\{c^T(x-y) : x \in X, \ y \in X, \ b_1^T(x-y) = 0, \ldots, b_{j-1}^T(x-y) = 0\}, && (47)\end{aligned}$$

(cf. expressions (17) and (18)). Here, we notice that $F(c) = F_1(c)$ is the width of X in the direction c. From the above we see that a lattice vector 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 2.3 and 2.4 below, and the definition of a so-called *generalized Korkine-Zolotarev basis*. Let b_j, $1 \le j \le n$ be defined recursively as follows. Given $b_1, \ldots, b_{j-1}$, the vector b_j minimizes $F_j(x)$ over all lattice vectors that are linearly independent of $b_1, \ldots, b_{j-1}$. A generalized Korkine-Zolotarev (KZ) basis is defined to be any proper basis $b'_1, \ldots, b'_n$ associated with b_j, $1 \le j \le n$, see Definition 1.5 for the definition of a proper basis. The notion of a generalized KZ basis was introduced by Kannan & Lovász [57], [58]. Kannan & Lovász [57] gave an algorithm for computing a generalized KZ basis in polynomial time for fixed n.

Theorem 2.3. *[58]. Let $F(c)$ be the length of the shortest lattice vector c with respect to the set $(X - X)^*$, and let $\rho_{\mathrm{KZ}} = \sum_{j=1}^n F_j(b'_j)$, where b'_j, $1 \le j \le n$ is a generalized Korkine-Zolotarev basis. There exists a universal constant c_0 such that*

$$F(c)\rho_{\mathrm{KZ}} \le c_0 \cdot n \cdot (n+1)/2. \quad (48)$$

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 [9], is equal to $\frac{c_{\mathrm{BM}}^n}{n^n}$, where c_{BM} is a constant depending only on n. In Theorem 2.3 we have $c_0 = \frac{4}{c_{\mathrm{BM}}}$. See also the remark below.

Theorem 2.4. *[58]. Let $b_1, \ldots, b_n$ be any basis for $\mathbb{Z}^n$, and let X be a bounded convex set that is symmetric about the origin. If $\rho = \sum_{j=1}^n F_j(b_j) \le 1$, then X contains an integral vector.*

The first step of the Lovász-Scarf algorithm is to compute the shortest vector c with respect to $(X - X)^*$ using the algorithm described in Section 1.2. If $F(c) \ge c_0 \cdot n \cdot (n+1)/2$, then $\rho_{\mathrm{KZ}} \le 1$, which by Theorem 2.4 implies that X contains an integral vector. If $F(c) < c_0 \cdot n \cdot (n+1)/2$, then we need to branch. Due to the definition of $F(c)$ we have in this case that $\max\{c^T x : x \in X\} - \min\{c^T x : x \in X\} < c_0 \cdot n \cdot (n+1)/2$, which implies that the polytope X in the direction c is "thin". As in the algorithm by Grötschel, Lovász, and Schrijver, we create one subproblem for every hyperplane $c^T x = \mu, \ldots, c^T x = \mu + c_0 \cdot n \cdot (n+1)/2$, where $\mu = \lceil \min\{c^T x : x \in X\} \rceil$. Once we have fixed a hyperplane $c^T x = t$, 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 polynomial number of branches at each level. 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 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 c. According to Proposition 1.8, $F(b_1) \leq (\frac{1}{2}-\varepsilon)^{1-n}F(c)$. In this version of the algorithm we would first check whether $F(b_1) \geq c_0 \cdot n \cdot (n+1)/(2(\frac{1}{2}-\varepsilon)^{1-n})$. If yes, then X contains an integral vector, and if no, we need to branch, and we create at most $c_0 \cdot n \cdot (n+1)/(2(\frac{1}{2}-\varepsilon)^{n-1})$ hyperplanes. We again obtain a search tree of at most n levels, but in this version the number of branches created at each level is polynomially bounded for fixed n only.

If the algorithm terminates with the result that X contains an integral vector, then Lovász and Scarf describe how such a vector can be constructed by using the Korkine-Zolotarev basis (see [68], proof of Theorem 10).

Remark 2.2. Lagarias, Lenstra, & Schnorr [60] derive bounds on the Euclidean length of Korkine-Zolotarev reduced basis vectors of a lattice and its dual lattice. Let W be the vector space spanned by the lattice L. The lattice L^* *dual* to L is defined as $L^* = \{w \in W : w^T v$ is an integer for all $v \in L\}$. The bounds are given in terms of the successive minima of L and L^*. These bounds, in turn, imply bounds on the product of successive minima of L and L^*. Later, Kannan & Lovász [57], [58] introduced the generalized Korkine-Zolotarev basis, as defined above, and derived bounds such as developed in the paper by Lagarias et al. These bounds were used to study covering minima of a convex set with respect 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 2.3 above. There are many interesting results on properties of various lattice constants. Many of them are described in the survey by Kannan [55], and will not be discussed further here.

Example 2.2. The following example demonstrates a few iterations with the generalized basis reduction algorithm. Consider the polytope $X = \{x \in \mathbb{R}^2_{\geq 0} : x_1 + 7x_2 \geq 7,\ 2x_1 + 7x_2 \leq 14,\ -5x_1 + 4x_2 \leq 4\}$. Let $j = 1$ and $\varepsilon = \frac{1}{4}$. 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 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

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

The initial basis is

$$b_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad b_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

We obtain $F_1(b_1) = 7.0$, $F_1(b_2) = 1.8$, $F_2(b_2) = 0.9$, $\mu = 0$, and $F_1(b_2 + 0b_1) = 1.8$, see Figure 7. Notice that the widths F_j are not the geometric widths, but the widths with respect to the indicated directions.

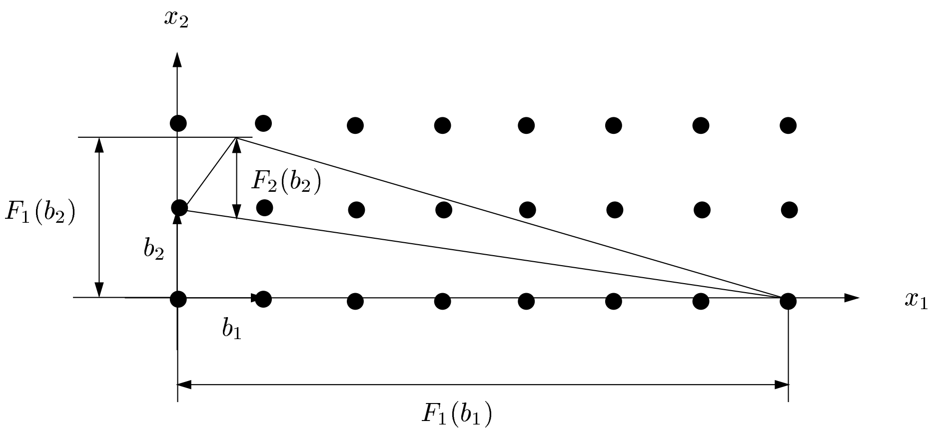

Figure 7:

Checking Conditions (19) and (20) shows that Condition (19) is satisfied as $F_1(b_2+0b_1) \geq F_1(b_2)$, but that Condition (20) is violated as $F_1(b_2) \not\geq (3/4)F_1(b_1)$, so we interchange b_1 and b_2 and remain at $j = 1$.

Now we have $j = 1$ and

$$b_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad b_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

$F_1(b_1) = 1.8,\ F_1(b_2) = 7.0,\ F_2(b_2) = 3.5,\ \mu = 4$, and $F_1(b_2 + 4b_1) = 3.9$, see Figure 8.

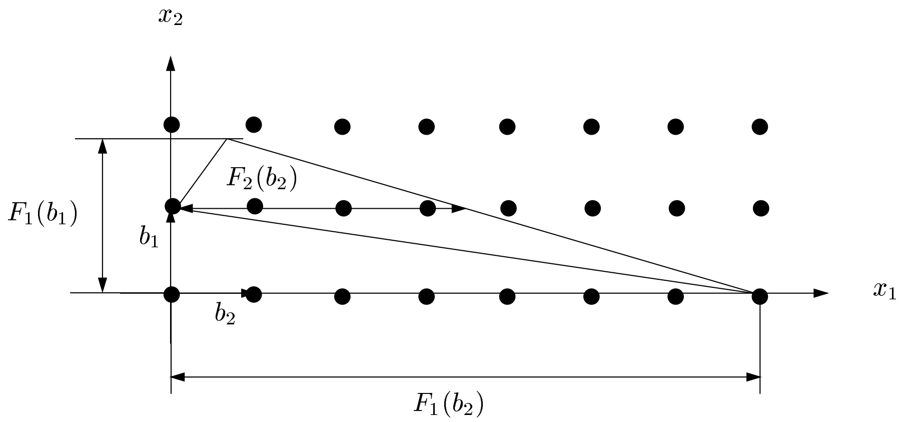

Figure 8:

Condition (19) is violated as $F_1(b_2 + 4b_1) \not\geq F_1(b_2)$, so we replace b_2 by $b_2 + 4b_1 = (1, 4)^T$. Given the new basis vector b_2 we check Condition (20) and we conclude that this condition is satisfied. Hence the basis

$$b_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad b_2 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

is Lovász-Scarf reduced, see Figure 9. The vectors b_1 and b_2 indicate directions in which the polytope X is thin.

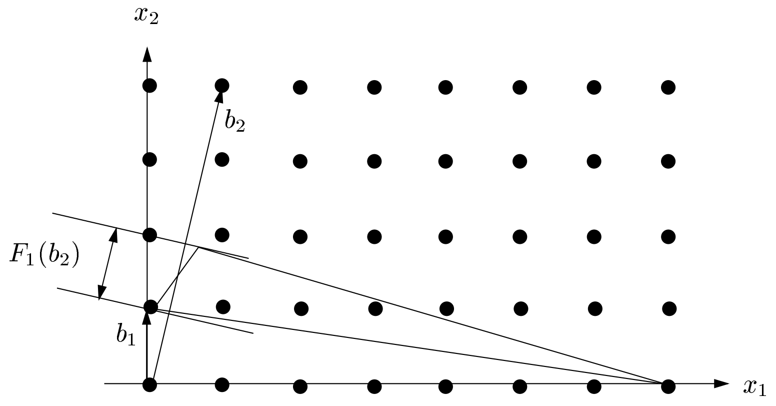

Figure 9:

□

2.2 Basis reduction and knapsack cryptosystems

Basis reduction has been used successfully to find solutions to subset sum problems arising in knapsack cryptosystems. For a recent excellent overview we refer to Joux and Stern [52].

A sender wants to transmit a message to a receiver. The plaintext message of the sender consists of a 0-1 vector $x_1, \ldots, x_n$, and this message is encrypted by using integral 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:

$$\text{Determine a 0-1 vector } x \text{ such that } \sum_{j=1}^{n} a_j x_j = a_0 \tag{49}$$

can be solved easily. For an eavesdropper that 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

$$d(a) = d(\{a_1, \ldots, a_n\}) = \frac{n}{\log_2(\max_{1 \le j \le n}\{a_j\})}. \tag{50}$$

The density, as defined above, is an approximation of the information rate at which bits are transmitted. The interesting case is $d(a) \le 1$, since for $d(a) > 1$ the subset sum problem (49) will in general have several solutions, which makes it unsuitable for generating encrypted messages. Lagarias and Odlyzko [61] proposed an algorithm based on basis reduction that often finds a solution to the subset sum problem (49) for instances having relatively low density. Earlier research had found methods based on recovering trapdoor information. If the information rate is high, i.e., $d(a)$ 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:

$$L_{a,a_0} = \{(x_1, \ldots, x_n, \ (ax - a_0\xi))^T\} \tag{51}$$

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

$$B = \begin{pmatrix} I^{(n)} & 0^{(n\times 1)} \\ a & -a_0 \end{pmatrix}, \tag{52}$$

where $I^{(n)}$ denotes the n-dimensional identity matrix, and where $0^{(n\times 1)}$ denotes an $(n \times 1)$ matrix (i.e. a column vector) consisting only of zeros. To see that B is a basis for L_{a,a_0}, we note that taking integer linear combinations of the column vectors of B generates vectors of type (51). Let $x \in \mathbb{Z}^n$ and $\xi \in \mathbb{Z}$. We obtain

$$\begin{pmatrix} x \\ ax - a_0\xi \end{pmatrix} = B \begin{pmatrix} x \\ \xi \end{pmatrix}. \tag{53}$$

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

1. Apply Lovász' basis reduction algorithm to the basis B (52), which yields a reduced basis B'.

2. Check if any of the columns $b'_k = (b'_{1k}, \ldots, b'_{n+1,k})$ has all $b'_{jk} = 0$ or γ for some fixed constant γ, for $1 \leq j \leq n$. If such a reduced basis vector is found, check if the vector $x_j = b'_{jk}/\gamma$, $1 \leq j \leq n$ 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 B with $a_0 = \sum_{j=1}^n a_j - a_0$, which corresponds to complementing all x_j-variables.

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 2.5 below shows, however, we can expect algorithm SV to work well on instances of (49) having low density. Consider a 0-1 vector x, which we will consider as fixed. We assume that $\sum_{j=1}^n x_j \leq \frac{n}{2}$. The reason for this assumption is that either $\sum_{j=1}^n x_j \leq \frac{n}{2}$, or $\sum_{j=1}^n x'_j \leq \frac{n}{2}$, 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 $\bar{x} = (x_1, \ldots, x_n, 0)$. Let the sample space $\Lambda(A, \bar{x})$ of lattices be defined to consist of all lattices L_{a,a_0} generated by the basis (52) such that

$$1 \leq a_j \leq A, \quad \text{for } 1 \leq j \leq n, \tag{54}$$

and

$$a_0 = \sum_{j=1}^n a_j \bar{x}_j. \tag{55}$$

There is precisely one lattice in the sample space for each vector a satisfying (54). Therefore the sample space consists of A^n lattices.

Theorem 2.5. *[61]. Let $\bar{x}$ 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_{a,a_0} in $\Lambda(A, \bar{x})$ that contain a vector v such that $v \neq k\bar{x}$ for all $k \in \mathbb{Z}$, and such that $||v||^2 \leq \frac{n}{2}$ is*

$$O(A^{n-c_1(\beta)}(\log A)^2), \tag{56}$$

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 2.5 applies to lattices having density $d(a) < (1.54725)^{-1} \approx 0.6464$. Expression (56) 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 $\Lambda(A, \bar{x})$ for which $\bar{x}$ is the *shortest* non-zero vector. Here we notice that the term (56) 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 $\Lambda(A, \bar{x})$ have $\bar{x}$ as the shortest vector. So, the subset sum problems (49) with density $d(a) < 0.6464$ could be solved in polynomial time if we had an oracle that could compute the shortest vector in the lattice L_{a,a_0}. Lagarias and Odlyzko also prove that the algorithm SV actually finds a solution to "almost all" feasible subset sum problems (49) having density $d(a) < (2 - \varepsilon)(\log(\frac{4}{3}))^{-1} n^{-1}$ for any fixed $\varepsilon > 0$.

Coster, Joux, LaMacchia, Odlyzko, Schnorr, & Stern [24] proposed two ways of improving Theorem 2.5. They showed that "almost all" subset sum problems (49) having density $d(a) < 0.9408$ 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 $L'_{a,a_0} \in \mathbb{Q}^{n+1}$ considered by Coster et al. is defined as

$$L'_{a,a_0} = \{(x_1 - \frac{1}{2}\xi, \ldots, x_n - \frac{1}{2}\xi,\ N(ax - a_0\xi))^T\}, \tag{57}$$

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

$$\bar{B} = \begin{pmatrix} I^{(n)} & (-\frac{1}{2})^{(n\times 1)} \\ Na & -Na_0 \end{pmatrix}. \tag{58}$$

Here $(-\frac{1}{2})^{(n\times 1)}$ denotes the $(n \times 1)$-matrix consisting of elements $-\frac{1}{2}$ only. As in the analysis by Lagarias and Odlyzko, we consider a fixed vector $x \in \{0,1\}^n$, and we let $\bar{x} = (x_1, \ldots, x_n, 0)$. The vector $\bar{x}$ does not belong to the lattice L', but the vector $w = (w_1, \ldots, w_n, 0)$, where $w_j = x_j - \frac{1}{2}$, $1 \leq j \leq n$ does. So, if Lovász' basis reduction algorithm is applied to $\bar{B}$ and if the reduced basis $\bar{B}'$ 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 (49). 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 2.5.

Theorem 2.6. *[24]. 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 $x = (x_1, \ldots, x_n)$, $x_j \in \{0, 1\}$, be fixed, and let $a_0 = \sum_{j=1}^{n} a_j x_j$. If the density $d(a) < 0.9408$, then the subset sum problem (49) 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'_{a,a_0}.*

Coster et al. prove Theorem 2.6 by showing that the probability that the lattice L'_{a,a_0} contains a vector $v = (v_1, \ldots, v_{n+1})$ satisfying

$$v \neq kw \quad \text{for all } k \in \mathbb{Z}, \text{ and } ||v||^2 \leq ||w||^2 \tag{59}$$

is bounded by

$$n(4n\sqrt{n} + 1)\frac{2^{c_0 n}}{A} \tag{60}$$

for $c_0 = 1.0628$. Using the lattice L', note that $||w||^2 \leq \frac{n}{4}$. The number N in basis (58) is used in the following sense. Any vector in the lattice L' is an integer linear combination of the basis vectors. Hence, the $(n+1)^{\text{st}}$ 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 $(n+1)^{\text{st}}$ element is equal to zero. Since it is known that the length of 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 v to be shorter than w it should satisfy $v_{n+1} = 0$. Hence, Coster et al. only need to consider lattice vectors v in their proof that satisfy $v_{n+1} = 0$. In the theorem we assume that the density $d(a)$ of the subset sum problems is less than 0.9408. Using the definition of $d(a)$ we obtain $d(a) = n/\log_2(\max_{1 \leq j \leq n}\{a_j\}) < 0.9408$, 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 (60) goes to zero as n goes to infinity, which shows that "almost all" subset sum problems having density $d(a) < 0.9408$ can be solved in polynomial time given the existence of a shortest vector oracle. Coster et al. also gave another lattice $L''(a, a_0) \in \mathbb{Z}^{n+2}$ that could be used to obtain the result given in Theorem 2.6. The lattice $L''(a, a_0)$ consists of vectors

$$L''(a, a_0) = \tag{61}$$

$$\left\{ \left((n+1)x_1 - \sum_{\substack{k=1 \\ k \neq 1}}^{n} x_k - \xi, \ldots, (n+1)x_n - \sum_{\substack{k=1 \\ k \neq n}}^{n} x_k - \xi, (n+1)\xi - \sum_{j=1}^{n} x_j,\ N(ax - a_0\xi) \right)^T \right\},$$

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 $L''(a, a_0)$ is not full dimensional as the basis consists of $n+1$ vectors. Given a reduced basis vector $w = (w_1, \ldots, w_{n+1}, 0)$, we solve the system of equations

$$w_j = (n+1)x_j - \sum_{\substack{k=1 \\ k \neq j}}^{n} x_k - \xi,\ 1 \leq j \leq n,\ w_{n+1} = (n+1)\xi - \sum_{j=1}^{n} x_j$$

and check whether $\xi = 1$, and the vector $x \in \{0,1\}^n$. If so, x solves the subset sum problem (49). Coster et al. show that for $x \in \{0,1\}^n$, $\xi = 1$, we obtain $||w||^2 \leq \frac{n^3}{4}$, and they indicate how to show that most of the time there will be no shorter vectors in $L''(a, a_0)$.

2.3 Solving diophantine equations using basis reduction

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

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

Here A is an $m \times n$-matrix, with $m \leq n$, and the vectors d, l, and u are of conformable dimensions. We assume that all input data is integral. Problem (63) is NP-complete, but if we remove the bound constraints $l \leq x \leq u$, 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 $X = \{x \in \mathbb{Z}^n : Ax = d,\ l \leq x \leq u\}$. Instead of using a method based on the linear relaxation of the problem, they considered the following integer relaxation of X, $X_{\mathrm{IR}} = \{x \in \mathbb{Z}^n : Ax = d\}$. Determining whether X_{IR} is empty can be carried out in polynomial time for instance by generating the Hermite normal form of the matrix A. Let x^0 be an integral vector satisfying $Ax^0 = d$, and let Y be an $n \times (n-m)$-matrix consisting of integer, linearly independent column vectors y^j, $1 \leq j \leq n-m$, such that $Ay^j = 0$ for $1 \leq j \leq n-m$. We can now rewrite X_{IR} as

$$X_{\mathrm{IR}} = \{x \in \mathbb{Z}^n : x = x^0 + Y\lambda,\ \lambda \in \mathbb{Z}^{n-m}\}, \tag{64}$$

that is, we express any vector x that satisfies $Ax = d$ as a vector x^0, satisfying $Ax^0 = d$, plus an integer linear combination of vectors that form a basis of the lattice $L_0 = \{x \in \mathbb{Z}^n : Ax = 0\}$. Since a lattice may have several bases, reformulation (64) is not unique.

The intuition behind the approach of Aardal et al. is as follows. Suppose that we are able to obtain a vector x^0 that is short with respect to the bounds. Then, we may hope that x^0 satisfies $l \leq x^0 \leq u$, in which case we are done. If x^0 does not satisfy the bounds, then we observe that $A(x^0 + \lambda y) = d$ for any integer multiplier λ and any vector y satisfying $Ay = 0$. Hence, we can derive an enumeration scheme in which we branch on integer linear combinations of vectors y satisfying $Ay = 0$, which explains the reformulation (64) of X_{IR}. Similar to Lagarias and Odlyzko, we choose a lattice, different from the standard lattice $\mathbb{Z}^n$, in which solutions to our problem (63) are relatively short vectors, and then apply basis reduction to the initial basis of the chosen lattice.

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

$$(x^T,\ N_1\xi,\ N_2(a_1 x - d_1\xi), \ldots, N_2(a_m x - d_m\xi))^T, \tag{65}$$

where a_i is the i^{th} row of the matrix A, where N_1 and N_2 are natural numbers, and where ξ, as in Section 2.2, is a variable associated with the right-hand side vector d. The basis B given below spans the lattice $L_{A,d}$:

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

The lattice $L_{A,d} \subset \mathbb{Z}^{m+n+1}$ is not full-dimensional as 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 different role of the number N used in the bases (58) and (62)). The following proposition states precisely which type of vectors we wish to obtain.

Proposition 2.7. *The integer vector x^0 satisfies $Ax^0 = d$ if and only if the vector*

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

belongs to the lattice L, and the integer vector y satisfies $Ay = 0$ if and only if the vector

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

belongs to the lattice L.

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

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

This result is stated in the following theorem.

Theorem 2.8. *[3]. Assume that there exists an integral vector x satisfying the system $Ax = d$. 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 $\hat{b}_j \in \mathbb{Z}^{n+m+1}$ of the reduced basis $\hat{B}$ have the following properties:*

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

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

In the proof of Properties 1 and 2 of Theorem 2.8, Aardal et al. make use of inequality (15) of Proposition 1.4.

Once we have obtained the matrix Y and the vector x^0, we can derive the following equivalent formulation of problem (63):

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

Aardal, Hurkens, & Lenstra [3], and Aardal, Bixby, Hurkens, Lenstra, & 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, & Wolsey [22], and the market split instances of Cornuéjols & Dawande [21]. The results were encouraging. After transforming problem (63) to problem (70), the size of for instance the market split instances that could be solved doubled. Aardal et al. [1] also investigated the performance of integer branching. Let $P = \{\lambda \in \mathbb{Z}^{n-m} : \ l \leq x^0 + Y\lambda \leq u\}$. At node k of the enumeration tree they choose a unit vector e^j, $1 \leq j \leq n-m$ that has not yet been chosen at any of the predecessors of node k. Then, they compute $\mu_k = \lceil \min\{(e^j)^T\lambda : \lambda \in P \cap \{\lambda_j\text{'s fixed at predecessors of } k\}\} \rceil$ and $\gamma_k = \lfloor \max\{(e^j)^T\lambda : \lambda \in P \cap \{\lambda_j\text{'s fixed at predecessors of } k\}\} \rfloor$. At node k, $\gamma_k - \mu_k + 1$ subproblems, or branches, are created by fixing λ_j to $\mu_k, \mu_k + 1, \ldots, \gamma_k$. Different strategies for choosing a unit direction e^j were considered. This branching scheme can be viewed as a heuristic version of the integer programming algorithms described Section 2.1. Instead of using vectors that give provably thin directions, only unit vectors were used. The experiments indicated that the unit vectors yield good directions, i.e., only few nodes were created at each branch, and typically, at a modest depth of the search tree only one branch was created. One way of explaining why the reformulated problem was so much easier to solve is that the index of the lattice $L_0 = \{x \in \mathbb{Z}^n : Ax = 0\}$ in $\mathbb{Z}^n$ is, in general, larger than one. Let Λ be a sublattice of the lattice M. The index I of Λ in M is defined as $I = \det(\Lambda)/\det(M)$. If the index of Λ in M is large, then M contains a large number of vectors that are different from the vectors in Λ, which means that a certain "scaling effect" is obtained. We illustrate this effect in the following example.

Example 2.3. Consider the polytope $X = \{x \in \mathbb{R}^3 : 2x_1 + 4x_2 + 5x_3 = 8,\ 0 \leq x_j \leq 1,\ 1 \leq j \leq 3\}$. The set X, is illustrated in grey in Figure 10. The question is: does X contain

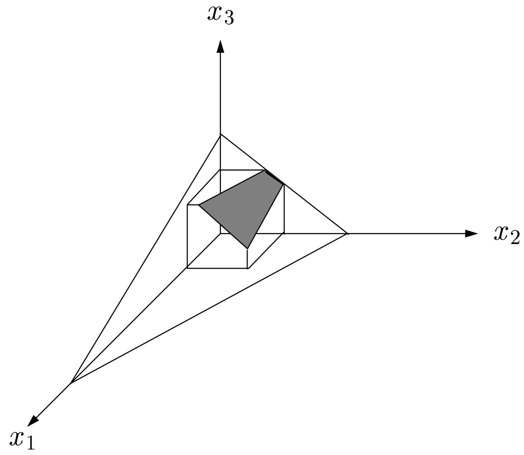

Figure 10:

an integral vector? To use branch-and-bound we need to introduce an objective function. Here we have chosen min $(x_1 + x_2 + x_3)$. The optimal solution to the linear relaxation of this instance is $x = (0, \frac{3}{4}, 1)^T$. Two branch-and-bound nodes are created by adding the constraints $x_2 = 0$ and $x_2 = 1$. The subproblem implied by $x_2 = 0$ is infeasible, but if we impose $x_2 = 1$ we obtain the solution $x = (0, 1, \frac{4}{5})$, and we need to branch on variable x_3.

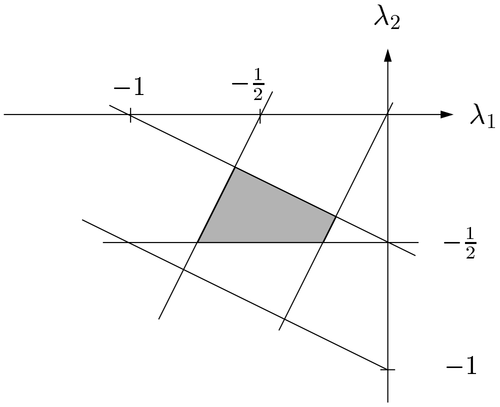

Figure 11:

If we reformulate the integer feasibility problem according to (70) we obtain, through basis reduction, the vector $x^0 = (0, 2, 0)^T$ and the matrix

$$Y = \begin{pmatrix} -2 & 1 \\ 1 & 2 \\ 0 & -2 \end{pmatrix}.$$

The question is: Does there exist a vector $\lambda \in \mathbb{Z}^2$ such that $\lambda \in P$, where $P = \{\lambda \in \mathbb{Z}^2 : 0 \leq -2\lambda_1 + \lambda_2 \leq 1,\ -2 \leq \lambda_1 + 2\lambda_2 \leq -1,\ 0 \leq -2\lambda_2 \leq 1\}$. The linear relaxation of P is given in Figure 11. If we use min $(\lambda_1 + \lambda_2)$ as objective function, we obtain the fractional point $\lambda = (-\frac{3}{4}, -\frac{1}{2})^T$, but, the subproblems created by branching on λ_1 as well as on λ_2 are infeasible. In fact, regardless of the objective function that is used, integer infeasibility is detected at the root node. This example is of course so small that it is hard to draw any conclusions, but if we draw the coordinate system corresponding to the formulation in λ-variables in the coordinate system of the x-variables, we can observe the scaling effect discussed above. This is done by translating the lattice $L_0 = \{x \in \mathbb{Z}^3 : 2x_1 + 4x_2 + 5x_3 = 0\}$ to the point x^0, i.e., the origin of the λ-coordinate system is located at the vector x^0. The unit vector $\lambda = (-1, 0)^T$ corresponds to the vector $x = (2, 1, 0)^T$, and the vector $\lambda = (0, -1)^T$ corresponds to the vector $x = (-1, 0, 2)^T$, see Figure 12. The determinant of the lattice L_0 is equal to $\sqrt{45}$, whereas the determinant of $\mathbb{Z}^3$ is equal to 1.

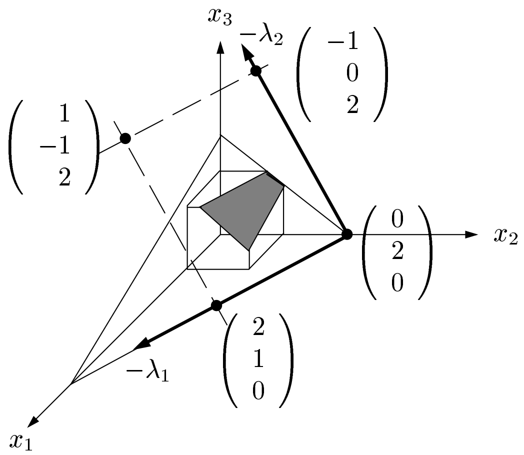

Figure 12:

□

The computational study by Aardal et al. [1] indicated that integer branching on the unit vectors in the space of the λ-variables taken in the order $j = n - m, \dots, 1$ was quite effective, and in general much better than the order $1, \dots, n - m$. This can be explained as follows. Due to Lovász' algorithm, the vectors of Y are more or less in order of increasing length, so typically, the $(n - m)^{\text{th}}$ vector of Y 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. Note, that to branch on the j^{th} vector of Y corresponds to branching on the j^{th} unit vector in the space of the λ-variables.

3 Augmentation Algorithms and Test Sets

A natural approach to attack a linear integer program is via the following augmentation algorithm.

Algorithm 3.1. An Augmentation Algorithm for a minimization problem

> Let x be any feasible point of the linear integer program.
>
> While x is not optimal, determine an integral vector z and a non-negative integer number λ such that (i) $x + \lambda z$ is feasible and (ii) $x + \lambda z$ attains a smaller objective function value than x. Set $x := x + \lambda z$.

One question that arises immediately is whether this algorithm can be made effective in terms of the number of augmentations that one needs to find an optimal solution. This topic is addressed in Section 3.1. We will see that one can solve the optimization problem with a polynomial number of calls of a *directed augmentation oracle*. From a mathematical point of view a study of the augmentation problem leads naturally to an investigation of Hilbert bases of pointed polyhedral rational cones and test sets. This approach is discussed in Section 3.2. Test sets for families of integer programs are collections of integral vectors with the property that every feasible non-optimal point of any integer program in the family can be improved by a vector in the test set. They can be designed from various mathematical viewpoints. One of these approaches is based on Hilbert bases, another one comes from Gröbner bases associated with toric ideals. The latter object is the central topic in Section 3.3.

3.1 From augmentation to optimization

There are two elementary questions that arise in the analysis of an augmentation algorithm for a linear integer program: how can one solve the subproblem of detecting an improving direction and secondly what is a bound on the number of improvement steps required in order to reach an optimal point. This subsection is dedicated to the latter question.

To be more formal, let $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $u \in \mathbb{Z}_+^n$. Throughout this section we assume that

$$X = \{x \in \mathbb{Z}^n : \; Ax = b, \; 0 \le x \le u\}, \tag{71}$$

is the set of all feasible solutions of the integer program. Our goal is to solve the optimization problem (OPT) by an augmentation algorithm, i.e., by repeated calls to an oracle that solves the augmentation problem.

The Optimization Problem (OPT)
Given a vector $c \in \mathbb{Z}^n$ and a point $x \in X$, find a vector $x^* \in X$ that minimizes c over X.

The Augmentation Problem (AUG)
Given a vector $c \in \mathbb{Z}^n$ and a point $x \in X$, find a point $y \in X$ such that $c^T y < c^T x$, or assert that no such y exists.

A classical example of an augmentation algorithm for solving the minimum cost flow problem in digraphs is a cycle cancelling algorithm that improves feasible flow along negative cycles. Such negative cycles can be detected efficiently in an augmentation network that one constructs from a feasible solution. In this network each original arc, on which the value of a feasible flow can increase or decrease without exceeding the corresponding lower and upper bound requirement, is replaced by a forward arc and a backward arc. A generalization of this directed augmentation network to general integer programs is the directed augmentation problem.

The Directed Augmentation Problem (DIR–AUG)
Given vectors $c, d \in \mathbb{Z}^n$ and a point $x \in X$, find vectors $z^1, z^2 \in \mathbb{Z}^n_+$ such that $c^T z^1 - d^T z^2 < 0$ and $x + z^1 - z^2$ is feasible, or assert that no such vectors z^1, z^2 exist.

In the case of the minimum cost flow problem in digraphs it is well known that a cycle cancelling algorithm that augments along any negative cycle does not necessarily converge to an optimal solution in polynomial time in the encoding length of the input data. Indeed, a more sophisticated strategy for augmenting is required. In the min-cost-flow application it is for instance the augmentation of feasible flows along maximum mean ratio cycles that makes the primal algorithm work efficiently. Maximum mean ratio cycles are very special objects and there is no obvious counterpart in the case of general integer programs. Indeed, to show that a polynomial number of calls to the directed augmentation oracle suffices to solve the optimization problem we need a combination of an interior point philosophy by using a barrier function and a maximum mean ratio augmentation.

Definition 3.1.

(a) For $x \in X$ and $j \in \{1, \ldots, n\}$ let
$p(x)_j := 1/(u_j - x_j)$ if $x_j < u_j$ and $p(x)_j := 0$, otherwise.
$n(x)_j := 1/x_j$ if $x_j > 0$ and $n(x)_j := 0$, otherwise.

(b) A vector $z \in \mathbb{Z}^n$ is called *exhaustive* w.r.t. a point $x \in X$ if $x + z \in X$ and for all $\lambda \in \mathbb{Z}^+$, $\lambda \geq 2$ we have that $x + \lambda z \notin X$.

(c) For the integer program (IP) let $C := \max\{|c_i| : i = 1, \ldots, n\}$ and $U := \max\{|u_i| : i = 1, \ldots, n\}$.

The Maximum Ratio Augmentation Problem (MRA)
Given a vector $c \in \mathbb{Z}^n$ and a point $x \in X$, find vectors $z^1, z^2 \in \mathbb{Z}^n_+$ such that $c^T(z^1 - z^2) < 0$, $x + z^1 - z^2$ is feasible and the objective $\lvert c^T(z^1 - z^2)\rvert / (p(x)^T z^1 + n(x)^T z^2)$ is maximum.

An important relation between the oracles (MRA) and (DIR-AUG) is stated below.

Lemma 3.1. *[85] (MRA) can be solved with $O(nlog(nCU))$ calls to an oracle that solves (DIR–AUG).*

We are now ready for analyzing a specific augmentation algorithm that we call MMA-Augmentation-Algorithm. This algorithm has been invented for the minimum cost flow problem by Wallacher [98]. Later McCormick and Shioura extended one of Wallacher's algorithms to linear programming over unimodular spaces [69]. For its analysis we resort to

Lemma 3.2. *(Geometric Improvement [4])*

Let $x^1, x^2, \ldots$ be a sequence of feasible points in X produced by some algorithm $\mathcal{A}$ such that $c^T x^1 > c^T x^2 > \ldots$. Let x^* be a solution of $\min c^T x : x \in X$. If there exists a constant $0 < \alpha < 1$ such that for all k

$$|c^T(x^{k+1} - x^k)| \geq \alpha |c^T(x^* - x^k)|,$$

then $\mathcal{A}$ terminates after $O(\log(nCU)/\alpha)$ steps with an optimal solution.

Proof. Consider a consecutive sequence of $\beta := \frac{2}{\alpha}$ iterations starting with iteration k. If each of these iterations improves the objective function value by at least $\frac{\alpha}{2}|c^T(x^* - x^k)|$, then $x^{k+\beta}$ is an optimal solution. Otherwise, there exists q such that

$$\alpha(|c^T(x^* - x^q)|) \leq |c^T(x^{q+1} - x^q)| \leq \frac{\alpha}{2}|c^T(x^* - x^k)|$$

$$\iff \quad |c^T(x^* - x^q)| \leq \frac{1}{2}|c^T(x^* - x^k)|,$$

i.e., after β iterations we have halved the gap between $c^T x^*$ and $c^T x^k$. □

Algorithm 3.2. *Algorithm [MMA]*

1 Let $x \in X$.

2 Call (MRA) with input x and objective function c

3 If (MRA) does not return vectors z^1, z^2 then STOP. Otherwise let z^1, z^2 be the output of (MRA).

4 Using binary search determine a maximum step length, i.e., a number $\lambda \in \mathbb{Z}_+$ such that $\lambda(z^1 - z^2)$ is exhaustive.

5 Set $x := x + \lambda(z^1 - z^2)$ and return to Step 2.

Theorem 3.3. *[85] For any $x \in X$ and $c \in \mathbb{Z}^n$, Algorithm [MMA] solves (OPT) with at most $O(n \log(nCU))$ calls of (MRA).*

Proof. Let $x \in X$ and $c \in \mathbb{Z}^n$. Assuming that x is not minimal w.r.t. c, let z^1, z^2 be the output of (MRA) and $\lambda \in \mathbb{Z}_+$ such that $\lambda(z^1 - z^2)$ is exhaustive. Let $z := \lambda(z^1 - z^2)$ and x^* be an optimal solution. We set $z^* := x^* - x$. Since z is exhaustive, there exists $j \in \{1, \ldots, n\}$ such that $x_j + 2z_j > u_j$ or $x_j + 2z_j < 0$. This situation occurs if and only if $z_j^+ > (u_j - x_j)/2$ or $z_j^- > x_j/2$. Therefore, $p(x)^T z^+ + n(x)^T z^- \geq 1/2$. Moreover, $p(x)^T (z^*)^+ + n(x)^T (z^*)^- \leq n$. On account of the condition that

$$|c^T z|/(p(x)^T z^+ + n(x)^T z^-) \geq |c^T z^*|/(p(x)^T (z^*)^+ + n(x)^T (z^*)^-)$$

we obtain that $|c^T z| \geq |c^T z^*|/(2n)$. Applying Lemma 3.2 yields the result. □

A consequence of Theorem 3.3 is

Theorem 3.4. *[85] Let X be given by an oracle that solves (DIR-AUG). Then for every $c \in \mathbb{Z}^n$ the optimization problem can be solved in oracle polynomial time.*

We remark that one can also use the method of *bit-scaling* (see [28]) in order to show that for a class of 0/1-integer programming problems the optimization problem can be solved by a polynomial number of calls of the (directed) augmentation oracle. This is discussed in Schulz, Weismantel & Ziegler [84] and in Grötschel and Lovász [42]. For a thorough introduction to oracles and oracle–polynomial time algorithms we refer to Grötschel, Lovász, and Schrijver [43].

3.2 From augmentation to Hilbert bases

In this section we summarize elementary links between the study of test sets for families of integer programs and the augmentation problem. The augmentation problem is the task of determining an improving integral direction for a specific non-optimal point. Test sets for families of integer programs are collections of integral vectors with the property that every feasible non-optimal point of the integer program can be improved by a vector in the test set. We will see that test sets can be derived from Hilbert bases of rational polyhedral cones. The following definitions about cones are used.

Definition 3.2. *A subset C of $\mathbb{R}^n$ is called a* cone *if for all $x, y \in C$ and $\lambda, \mu \geq 0$ we have that $\lambda x + \mu y \in C$. A cone C is called* polyhedral *if it is finitely generated, i.e., there exists a finite set $V = \{v^1, \ldots, v^k\} \subseteq \mathbb{R}^n$ such that*

$$C = \{y : \ y = \sum_{i=1}^{k} \lambda_i v^i : \ \lambda_1, \ldots, \lambda_k \geq 0\}.$$

If C is generated by V, we write $C = C(V)$. If $V \subseteq \mathbb{Q}^n$, $C(V)$ is called rational. *A cone C is* pointed *if there exists a hyperplane $a^T x \leq 0$ such that $\{0\} = \{x \in C : \ a^T x \leq 0\}$.*

In the following we always consider rational polyhedral cones and call them cones for short. We are interested in a special subset of integral vectors in a cone, namely an integral generating set.

Definition 3.3. *[33] Let C be a rational polyhedral cone. A finite set $H \subseteq C \cap \mathbb{Z}^n$ is a Hilbert basis of C if every integral vector in C can be represented as a non-negative integral combination of the elements of H.*

Example 3.1. Let $C = \{y \in \mathbb{R}^2 : \ y_1 = \lambda_1 + \lambda_2, \ y_2 = 3\lambda_1 + \lambda_2 : \ \lambda_1, \lambda_2 \geq 0\}$. The set $\{(1,3)^T, (3,1)^T\}$ does not form a Hilbert basis of C because $(2,2)^T \in C$ cannot be represented as a non-negative integral combination of the vectors $(1,3)^T$ and $(3,1)^T$. However the set

$$\{(1,1)^T, (2,1)^T, (1,2)^T, (1,3)^T, (3,1)^T\}$$

is a Hilbert basis of C.

Theorem 3.6 tells us that Hilbert bases of rational cones exist. This result is fundamental. Its proof may be derived from the Gordan Lemma [39] but can also be given directly.

Theorem 3.5. (Gordan Lemma) *[39] Let $P \neq \emptyset \subseteq \mathbb{Z}_+^n$. There exists a unique minimal and finite subset $\{p_1, \dots, p_m\}$ of P such that $p \in P$ implies that $p^j \leq p$ for at least one $j \in \{1, \dots, m\}$.*

Theorem 3.6. *[39], see also [83] Every rational polyhedral cone possesses a Hilbert basis.*

Proof. Let $C(p^1, \dots, p^m)$ be a rational polyhedral cone generated by the vectors $p^1, \dots, p^m \in \mathbb{Z}^n$. A Hilbert basis H of C is always contained in the zonotope,

$$\begin{aligned} H \subseteq \mathcal{Z} \;=\; & \{p^1, \dots, p^m\} \cup \\ & \left\{ p \in C \backslash \{0\} : \; p = \sum_{i=1}^{m} \lambda_i p^i, \; 0 \leq \lambda_i < 1, \;\; 1 \leq i \leq m \right\}. \end{aligned} \tag{72}$$

Since $|\mathcal{Z} \cap \mathbb{Z}^n|$ is finite, the claim follows. □

Not every rational cone has however a unique Hilbert basis that is minimal with respect to inclusion.

Example 3.2. Let $C = \{x \in \mathbb{R}^2 \; : \; x_1 + 2x_2 = 0\}$. It may be checked that the set $(2, -1)^T, (-2, 1)^T$ is a Hilbert basis that is minimal with respect to taking subsets. The cone C also possesses a second Hilbert basis that is minimal with respect to inclusion consisting of the vectors $(4, -2)^T, (-2, 1)^T$.

If $C = C(p^1, \dots, p^m) \subseteq \mathbb{R}^n$ is a pointed cone then a Hilbert basis H of C that is minimal with respect to inclusion is uniquely determined (cf. [23], [83]),

$$\begin{aligned} H = \Big\{ z \in C \cap \mathbb{Z}^n \backslash \{0\} : & \, z \text{ is not the sum of two other} \\ & \text{vectors in } C \cap \mathbb{Z}^n \backslash \{0\} \Big\}. \end{aligned} \tag{73}$$

Theorem 3.7. *[23] If a rational polyhedral cone C is pointed, then there exists a unique Hilbert basis that is minimal with respect to inclusion. This minimal Hilbert basis is denoted $H(C)$.*

Let O_j denote the j-th orthant in $\mathbb{R}^n$. We denote by

$$IP(b, c, u) \quad \min\{c^T x : \; Ax = b, \; 0 \leq x \leq u, \; x \in \mathbb{Z}^n\}$$

the family of integer programs associated with a fixed matrix A and varying $b \in \mathbb{Z}^m, u \in \mathbb{Z}_+^n, \; c \in \mathbb{R}^n$. Then $C_j := O_j \cap \{x \in \mathbb{R}^n \; : \; Ax = 0\}$ is a pointed polyhedral cone in $\mathbb{R}^n$. On account of Theorems 3.6 and 3.7, C_j possesses a unique and finite Hilbert basis $H_j := H(C_j)$. The set

$$\mathcal{H} := \bigcup_j H_j \setminus \{0\}$$

is called the *Graver test set* for the family of integer programs $IP(b, c, u)$. In particular, the Graver test set is a finite set. Moreover, it contains a test set for every member of the family of integer programs $IP(b, c, u)$.

Theorem 3.8. *[41] The Graver test set $\mathcal{H}$ contains a test set for all integer programs of the family $IP(b, c, u)$ with varying $b \in \mathbb{Z}^m, u \in \mathbb{Z}_+^n,\ c \in \mathbb{R}^n$.*

Proof. Let $b \in \mathbb{Z}^m$, $u \in \mathbb{Z}_+^n$ and $c \in \mathbb{R}^n$ and consider the integer program $\min c^T x :\ Ax = b,\ 0 \leq x \leq u,\ x \in \mathbb{Z}^n$. Let x be a feasible point for this program that is not minimal with respect to c and let y be an optimal solution. On account of $Ay = b = Ax$, it follows that $A(y - x) = 0$, $y - x \in \mathbb{Z}^n$ and $c^T(y - x) < 0$. Let O_j denote the orthant that contains $y - x$. As $y - x$ is an integral point in C_j, there exist multipliers $\epsilon_h \in \mathbb{Z}_+$ for all $h \in H_j$ such that $y - x = \sum_{h \in H_j} \epsilon_h h$. As $c^T(y - x) < 0$ and $\epsilon_h \geq 0$ for all $h \in H_j$, there exists a vector $h^* \in H_j$ such that $c^T h^* < 0$ and $\epsilon_{h^*} > 0$. h^* lies in the same orthant as $y - x$, i.e., if $y_j - x_j > 0$, then $y_j - x_j \geq h_j^* \geq 0$ and if $y_j - x_j < 0$, then $y_j - x_j \leq h_j^* \leq 0$. Since x and y are feasible for the same integer program we obtain that h^* is an "augmenting vector" and that $x + h^*$ is feasible. □

Example 3.3. Consider the family of equality knapsack problems,

$$\min\{c_1 x_1 + c_2 x_2 + c_3 x_3 :\ x_1 + 2x_2 + 3x_3 = b, 0 \leq x \leq u, x \in \mathbb{Z}^3\}$$

with varying $b \in \mathbb{Z}$ and c, u. Since the Graver test set is symmetric about the origin, it suffices to analyze four orthants:

$$O_1 = \{x_1 \geq 0, x_2 \geq 0, x_3 \geq 0\}, \quad O_2 = \{x_1 \geq 0, x_2 \geq 0, x_3 \leq 0\}$$

$$O_3 = \{x_1 \geq 0, x_2 \leq 0, x_3 \geq 0\}, \quad O_4 = \{x_1 \leq 0, x_2 \geq 0, x_3 \geq 0\}$$

For $j \in \{1, 2, 3, 4\}$ we need to determine a Hilbert basis H_j of the cone

$$C_j = \{x \in O_j :\ x_1 + 2x_2 + 3x_3 = 0\}.$$

We obtain

$$\begin{aligned} H_1 &= \emptyset, \\ H_2 &= \{(3, 0, -1), (0, 3, -2), (1, 1, -1)\}, \\ H_3 &= \{(2, -1, 0), (0, -3, 2), (1, -2, 1)\}, \\ H_4 &= \{(-3, 0, 1), (-2, 1, 0)\}. \end{aligned} \tag{74}$$

Therefore, the Graver test set for this family of integer programs is the set

$$\mathcal{H} = \{\pm(3, 0, -1), \pm(0, 3, -2), \pm(1, 1, -1), \pm(2, -1, 0), \pm(1, -2, 1).$$

Using the notion of irreducibility of vectors there is a second equivalent characterization of the Graver test set.

Definition 3.4. *Let $A \in \mathbb{Z}^{m \times n}$. We say that a vector $v \in \mathbb{Z}^n \setminus \{0\}$* reduces *$w \in \mathbb{Z}^n \setminus \{0, v\}$ w.r.t. A if the following properties hold:*

$$\begin{array}{ll} v^+ \leq w^+, & v^- \leq w^-, \\ (Av)^+ \leq (Aw)^+, & (Av)^- \leq (Aw)^-. \end{array} \tag{75}$$

If such a v exists, w is called reducible. *Otherwise, w is* irreducible.

If v reduces w, then also $w - v$ reduces w and we have that

$$\begin{array}{ll} v^+ + (w-v)^+ = w^+, & v^- + (w-v)^- = w^-, \\ (Av)^+ + (A(w-v))^+ = (Aw)^+, & (Av)^- + (A(w-v))^- = (Aw)^-. \end{array} \tag{76}$$

For a cone C_j of the form $\{x \in O_j : Ax = 0\}$, these conditions ensure that the set of all irreducible integral points that lie in C_j define a Hilbert basis of this cone. Moreover, $w \in C_j \cap \mathbb{Z}^n$ implies that w is an element of the lattice

$$L = \{x \in \mathbb{Z}^n : Ax = 0\}.$$

This yields

Remark 3.1. *The set of all irreducible lattice vectors $v \in L$ is the Graver test set for the family of integer programs $IP(b, c, u)$.*

Proof. Let $\mathcal{H}$ be the Graver test set for the family of integer programs $IP(b, c, u)$. Let H_j denote the unique Hilbert basis H_j of the pointed cone $C_j = \{x \in O_j : Ax = 0\}$. A vector $v \in L \cap C_j$ is irreducible if and only if v cannot be written as the sum of other lattice vectors in C_j. This is true if and only if v is contained in H_j. $\square$

We will see in the next section that the notion of reducibility provides a way to determine Hilbert bases algorithmically. In this context a question arises concerning the complexity of deciding whether a vector is reducible.

Theorem 3.9. *[86] Given a pointed cone $C \subset \mathbb{R}^n$ and a vector $z \in C \cap \mathbb{Z}^n$, it is co$\mathcal{NP}$-complete to decide whether z is contained in the minimal Hilbert basis of C.*

Theorem 3.9 asserts the difficulty of deciding whether an integral vector is reducible. On the other hand, every augmentation vector can be decomposed into irreducible ones. In fact, we can write every integral vector in a pointed cone as a non-negative integer combination of at most $2n - 2$ irreducible vectors.

Theorem 3.10. *[86] Let C be a pointed cone in $\mathbb{R}^n$ and $H(C)$ its minimal Hilbert basis. Every integral point in C can be written as the non-negative integral combination of at most $2n - 2$ elements from $H(C)$.*

Theorem 3.10 improves a result of Cook, Fonlupt & Schrijver [18] who showed that every integral vector in a pointed n-dimensional cone is the non-negative integral combination of at most $2n - 1$ vectors from the minimal Hilbert basis. From a result of Sebö [86] follows that in dimensions $n = 2$ and $n = 3$ every integral vector in a pointed n-dimensional cone is the non-negative integral combination of at most n vectors from the Hilbert basis. This also holds for cones arising from perfect graphs [18] and a class of cones described in [46]. However, in general at least $n + \lfloor 1/6 \cdot n \rfloor$ elements of the Hilbert basis are needed to represent any integral vector in the cone.

Theorem 3.11. *[12] Let C be a pointed cone in $\mathbb{R}^n$ and $H(C)$ its minimal Hilbert basis. In general at least $n + \lfloor 1/6 \cdot n \rfloor$ elements from $H(C)$ are needed in order to represent any vector in $C \cap \mathbb{Z}^n$ as a non-negative integral combination of elements of $H(C)$.*

We have seen that Hilbert bases of rational polyhedral cones are central in the design of a test set. In fact, Hilbert bases play a central role in the theory of integer programming in general. Of major importance is their link to the integrality of polyhedra, i.e., to totally dual integral systems of inequalities.

Definition 3.5. *Let $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$. The system of inequalities $Ax \leq b$ is called* totally dual integral *(TDI) if for every $c \in \mathbb{Z}^m$ such that $|\min\{b^T y : \ A^T y = c, \ y \geq 0\}| < \infty$, there exists an integral vector $y^* \in \mathbb{Z}^m$, $A^T y^* = c$, $y^* \geq 0$ with $b^T y^* = \min\{b^T y : \ A^T y = c, y \geq 0\}$.*

The TDIness of the system $Ax \leq b$ has an important consequence for polyhedra and a geometric meaning.

Theorem 3.12. *[29] If $Ax \leq b$ is TDI and b is integral, then $\{x \in \mathbb{R}^n : \ Ax \leq b\}$ is integral.*

Let c be an integral vector that lies in the cone generated by all the rows of A. Among all possible ways of writing c as a conic combination of the row vectors of A, let S be the set of shortest conic combinations with respect to the function b, i.e.,

$$S = \{y \geq 0 : \ A^T y = c \text{ such that } b^T y \text{ is minimal}\}.$$

Then $Ax \leq b$ is called TDI if there exists an integral vector in S.

This geometric property can be expressed using Hilbert bases.

Theorem 3.13. *[33] Let $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$. The system $Ax \leq b$ is TDI if and only if for every face F of $P = \{x \in \mathbb{R}^n : \ Ax \leq b\}$ the set of row vectors that determine F is a Hilbert basis of the cone generated by these row vectors.*

In fact, the converse of Theorem 3.12 is also true.

Theorem 3.14. *[33] If a rational polyhedron P is integral, then there exists a TDI system $Ax \leq b$ such that b is integral and $P = \{x \in \mathbb{R}^n : \ Ax \leq b\}$.*

Hilbert bases can also be used to estimate the distance between feasible solutions of an integer program.

Theorem 3.15. *[45] Let $A \in \mathbb{Z}^{m\times n}$ with all subdeterminants at most α in absolute value and $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$. If $\widetilde{x}$ is a feasible, non-optimal solution of the program $\min\{c^T x : Ax \leq b, \ x \in \mathbb{Z}^n\}$, then there exists a feasible solution $\widehat{x}$ such that $c^T\widehat{x} < c^T\widetilde{x}$ and $||\widehat{x} - \widetilde{x}||_\infty \leq (n-1)\alpha - (n-2)/(n^{n/2}\alpha^{n-2})$.*

The bound of Theorem 3.15 strengthens the bound of $n\alpha$ given in [19]. Its proof is based on an analysis of the *height* of a Hilbert basis, see [45] and also [66]. For further results about the structure and applications of Hilbert bases to combinatorial convexity, toric varieties and polynomial rings and ideals we refer the reader to the papers mentioned above and to Schrijver [83], Liu [65], Bruns & Gubeladze [10], Firla & Ziegler [31] Henk & Weismantel [47], Dais & Haus & Henk[26], Ewald [30], Oda [72], Sturmfels [89] and Bruns & Gubeladze & Trung [11]).

3.3 Hilbert bases versus Gröbner bases

We have seen that the Graver test set is naturally derived from a study of Hilbert bases of cones. There are two other ways of defining test sets that rely on a different mathematical approach. The *neighbors of the origin* define a test set that was introduced by Scarf [76], [78]. It is based on a study of lattice point free convex bodies and establishes a beautiful link between the area of test sets and the geometry of numbers that we do not discuss here. The *reduced Gröbner basis* of an integer program is a test set obtained from a study of generators of polynomial ideals. The latter topic is a classical field of algebra. The reduced Gröbner basis of a toric ideal that one associates with an integral matrix A and a term order induced by c yields a test set for the familiy of integer programs

$$IP(b) \quad \min\{c^T x : \ Ax = b, \ x \in \mathbb{Z}^n_+\}$$

associated with a fixed matrix $A \in \mathbb{Z}^{m\times n}$ and varying $b \in \mathbb{Z}^m$. The connection between test sets for integer programming and Gröbner bases of toric ideals was first established by Conti &Traverso [15]. This "algebraic view of test sets" is important from an algorithmic point of view. Reduced Gröbner bases of toric ideals can be computed by the Buchberger algorithm [13]. Reinterpreting the steps of this algorithm as operations on lattice vectors yields a combinatorial algorithm for computing test sets, see [92], [95]. We consider here a geometric interpretation of Gröbner bases for integer programs. We refer to Cox, Little & O'Shea [25] and Becker & Weispfenning [6] for basics on Gröbner bases and on Buchberger's algorithm for polynomial ideals that motivated these constructions. As in the previous section let L denote the lattice $\{x \in \mathbb{Z}^n : \ Ax = 0\}$.

In order to avoid technical difficulties we make the following two assumptions:

Assumption 3.1. *c is generic, i.e., $c^T x = 0$ for $x \in L$ if and only if $x = 0$. Moreover, A is a matrix in $\mathbb{Z}^{m\times n}$ such that $\{x \in \mathbb{Z}^n_+ : \ Ax = 0\} = \{0\}$. The latter assumption ensures that the integer program $IP(b)$ is bounded for every $b \in \mathbb{Z}^m$.*

Let P be the set of all non-negative integer points that are not optimal in $IP(b)$ for any value of b. More formally,

$$P = \{x \in \mathbb{Z}^n_+ : \ \exists y \in \mathbb{Z}^n_+ \text{ such that } Ay = Ax, \ c^T y < c^T x\}.$$

Note that the set P is well defined, because c is generic. The geometric structure of the set P can be nicely characterized, see Figure 3.3.

Lemma 3.16. *[92] There exists a unique minimal finite set of vectors $p^1, \dots, p^t$ in P such that*

$$P = \bigcup_{i=1}^{t} (p^i + \mathbb{Z}^n_+),$$

where $v + \mathbb{Z}^n_+ := \{w \in \mathbb{Z}^n_+ : \ w \geq v\}$. Moreover, for any $x \in \{p^1, \dots, p^t\}$ and $y \in \mathbb{Z}^n_+$ such that $c^T y < c^T x$ and $Ay = Ax$ we have that $\operatorname{supp}(y) \cap \operatorname{supp}(x) = \emptyset$.

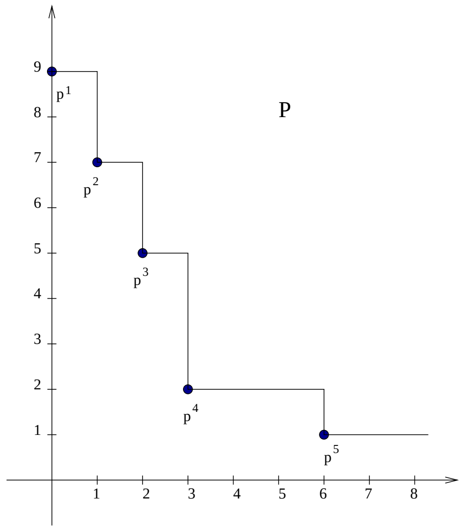

Figure 3.3: The geometric structure of the set P

Proof. Let $x \in P$. Then x does not attain the minimal objective function value with respect to c in the integer program with right hand side vector Ax. Let $y \in \mathbb{Z}^n_+$, be the unique minimal solution of $IP(Ax)$. Then for every $v \in \mathbb{Z}^n_+$ the vector $x + v$ is not minimal w.r.t. c because $c^T(y+v) < c^T(x+v)$ and $A(y+v) = Ay + Av = A(x+v)$. Therefore $x \in P$ implies that $(x + \mathbb{Z}^n_+) \subseteq P$. From Gordan's Lemma 3.5 we conclude that there exists a unique minimal and finite set of vectors $p^1, \dots, p^t \in P$ such that $P \subseteq \bigcup_{i=1}^t (p^i + \mathbb{Z}^n_+)$. This shows that $P = \bigcup_{i=1}^t (p^i + \mathbb{Z}^n_+)$.

Let $x \in \{p^1, \dots, p^t\}$ and $y \in \mathbb{Z}^n_+$ such that $c^T y < c^T x$ and $Ay = Ax$. Assuming that $k \in (\operatorname{supp}(y) \cap \operatorname{supp}(x))$ we have that $x - e^k \in \mathbb{Z}^n_+$ and $y - e^k \in \mathbb{Z}^n_+$. This implies that $x - e^k \in P$, a contradiction to the definition of $\{p^1, \dots, p^t\}$. $\square$

Taking into account the structure of the set P, we are ready to define a test set for the family of integer programs $IP(b)$ with varying $b \in \mathbb{Z}^m$.

Definition 3.6. *Let $p^1, \dots, p^t \in \mathbb{Z}^n_+$ be the set of vectors defined in Lemma 3.16 such that $P = \bigcup_{i=1}^t (p^i + \mathbb{Z}^n_+)$. For each $i \in \{1, \dots, t\}$ let y^i denote the optimal solution with respect to c of the program $IP(Ap^i)$. The set*

$$G_c := \{y^i - p^i : \ i = 1, \dots, t\}$$

is called the reduced Gröbner basis *of the family of integer programs $IP(b)$.*

Theorem 3.17. *[92] The reduced Gröbner basis G_c contains a test set for $IP(b)$ for every $b \in \mathbb{Z}^m$.*

Proof. Let $x \in P$. By Lemma 3.16 there exists $i \in \{1, \dots, t\}$ and $v \in \mathbb{Z}^n_+$ such that $x = p^i + v$. So, $x' := x + (y^i - p^i)$ satisfies $Ax' = Ax$ and $c^T x' < c^T x$ since $c^T y^i < c^T p^i$. Moreover, $x' = y^i + v \in \mathbb{Z}^n_+$. It follows that x is improved by a vector from the set G_c. $\square$

Next we show that the reduced Gröbner basis G_c is contained in the Graver test set $\mathcal{H}$.

Theorem 3.18. *[92] The reduced Gröbner basis G_c is contained in the Graver test set $\mathcal{H}$.*

Proof. Let $z \in G_c$ and $P = \bigcup_{i=1}^{t}(p^i + \mathbb{Z}^n_+)$. From Lemma 3.16 we have that $z = z^+ - z^-$ with $z^- \in \{p^1, \ldots, p^t\}$. Let O_j denote the orthant that contains z, $C_j = \{x \in O_j : \; Ax = 0\}$ and $H_j = H(C_j)$ be the minimal Hilbert basis of C_j. We conclude that $z \in C_j$. Suppose that $z \notin H_j$. Then $z = v + w$ where $v, w \in C_j \cap \mathbb{Z}^n$. As $c^T z < 0$, we can assume w.l.o.g. $c^T v < 0$. We obtain $v^+, v^- \in \mathbb{Z}^n_+$, $Av^+ = Av^-$ and $v^- \in P$. We have $z^- = v^- + w^-$. But $w^- \in Z^n_+$ and $z^- \in \{p^1, \ldots, p^t\}$. This contradicts the definition of the set $\{p^1, \ldots, p^t\}$. Therefore $z \in H_j$. □

Definition 3.7. *The set*

$$\mathcal{G} := \bigcup_{c \in \mathbb{R}^n, c \text{ generic}} G_c$$

is called the universal Gröbner basis *associated with a matrix A.*

Lemma 3.18 implies that $\mathcal{G}$ is contained in the Graver test set $\mathcal{H}$. In particular, $\mathcal{G}$ is finite. In fact, this containment relation is sometimes, not always strict. Note, however, that the Graver test set is designed for a family of integer programs $\min c^T x;\ Ax = b,\ 0 \le x \le u,\ x \in \mathbb{Z}^n$ with varying upper bounds on the variables, whereas the universal Gröbner basis applies to a family of integer programs with no upper bounds on the variables $\min c^T x :$ $Ax = b,\ 0 \le x,\ x \in \mathbb{Z}^n$. To make a comparison between the two objects possible, one may transform a program $\min c^T x :\ Ax = b,\ 0 \le x \le u,\ x \in \mathbb{Z}^n$ into the form $\min c^T x + 0^T y :$ $Ax + 0y = b,\ x + y = u,\ x, y \in \mathbb{Z}^n_+$. To the latter integer program in dimension $2n$ the universal Gröbner basis setting applies. We obtain

Theorem 3.19. *[90] Let $\tilde{G}$ be the universal Gröbner basis associated with the family of integer programs*

$$\min\{\tilde{c}^T(x, y) :\ Ax + 0y = b,\ x + y = u,\ (x, y) \in \mathbb{Z}^{2n}_+\}$$

with varying $u \in \mathbb{Z}^n$, $b \in \mathbb{Z}^n$ and generic $\tilde{c} \in \mathbb{R}^{2n}$. Let $\mathcal{H}$ be the Graver test set associated with the family of integer programs

$$\{\min c^T x :\ Ax = b,\ 0 \le x \le u,\ x \in \mathbb{Z}^n\}$$

with varying $u \in \mathbb{Z}^n$, $b \in \mathbb{Z}^n$ and $c \in \mathbb{R}^n$. Then $(x, -x) \in \tilde{G}$ if and only if $x \in \mathcal{H}$.

Example 3.4. Consider the family of integer programs with varying $c \in \mathbb{R}^3$, c generic and $b \in \mathbb{Z}$ of the form

$$\max\{c_1x_1 + c_2x_2 + c_3x_3 :\ x_1 + x_2 + 2x_3 = b, x \in \mathbb{Z}^3_+\}.$$

In this example the Graver test set $\mathcal{H}$ is the set

$$\mathcal{H} = \{\pm(1, -1, 0,), \pm(2, 0, -1), \pm(0, 2, -1), \pm(1, 1, -1)\}.$$

The universal Gröbner basis $\mathcal{G}$ is the set

$$\mathcal{G} = \{\pm(1,-1,0,), \pm(0,2,-1), \pm(2,0,-1)\}.$$

To see this note that $\pm(1,1,-1) = \frac{1}{2} \pm (0,2,-1) \pm \frac{1}{2}(2,0,-1)$. In fact, $(1,1,0) = \frac{1}{2}(0,2,0) + \frac{1}{2}(2,0,0)$, i.e.,$(1,1,0)$ cannot be a vertex of $\operatorname{conv}\{x \in \mathbb{Z}_+^3 : x_1 + x_2 + 2x_3 = 2\}$. Accordingly, $(-1,-1,1) \notin \mathcal{G}$, because for any objective function $c \in \mathbb{R}^3$, c generic such that $(1,1,0) \in P$, we have that $(1,0,0) \in P$ or $(0,1,0) \in P$. Therefore, $(1,1,0) \notin \{p^1, \ldots, p^t\}$, see Lemma 3.16. The six remaining vectors in $\mathcal{H}$ are also contained in $\mathcal{G}$, because there exist objective functions c for which these vectors define differences of a point $p^i \in P$ and the optimal solution y^i, see Definition 3.6. The example demonstrates that $\mathcal{G} \subseteq \mathcal{H}$, but $\mathcal{H}$ can be strictly bigger than $\mathcal{G}$.

The universal Gröbner basis $\mathcal{G}$ can be characterized geometrically. This is made precise in Theorem 3.20. For its proof in various versions we refer to the papers Sturmfels & Thomas [90], Thomas & Weismantel [94] , Sturmfels, Weismantel & Ziegler [91].

Theorem 3.20. *[90] Let $A \in \mathbb{Z}^{m \times n}$ of rank m.*

(i) Let $z = z^+ - z^- \in \mathcal{G}$. Then z^+ and z^- are vertices of the polyhedron $\operatorname{conv}\{x \in \mathbb{Z}_+^n : Ax = Az^+\}$. Moreover, the line with endpoints z^+ and z^- is an edge of the polyhedron $\operatorname{conv}\{x \in \mathbb{Z}_+^n : Ax = Az^+\}$.

(ii) Let z^1 and z^2 be two adjacent vertices of $\operatorname{conv}\{x \in \mathbb{Z}_+^n : Ax = Az^1\}$, then $(z^1 - z^2)/\gcd(z^1 - z^2) \in \mathcal{G}$.

We mentioned that Conti & Traverso [15] established the connection between test sets of integer programs and Gröbner bases of toric ideals. The latter objects can be computed by the Buchberger algorithm [13]. We discuss below a combinatorial variant of this procedure that allows us to determine a superset of the Graver test set and therefore a superset of the universal Gröbner basis for the family of integer programs $IP(b,c,u)$ and $IP(b)$, respectively. Starting with input $T := \{\pm e^i : i = 1, \ldots, n\}$ we repeatedly take all the sums of two vectors in T, reduce each of these vectors as much as possible by the elements of T and add all the reduced vectors that are different from the origin to the set T. On termination the set T contains the set of all irreducible vectors w.r.t. the matrix A.

Algorithm 3.3.

Input: $A \in \mathbb{Z}^{m \times n}$.
Output: A finite set T containing all the irreducible vectors w.r.t. A.

(1) Set $T_{old} := \emptyset$ and $T := \{\pm e^i : i = 1, \ldots, n\}$.

(2) While $T_{old} \neq T$ repeat the following steps:

(2.1) Set $T_{old} := T$.

(2.2) For all pairs of vectors $v, w \in T_{old}$, set $z := v + w$:

(2.2.1) While there exists $y \in T \setminus \{z\}$ reducing z, set $z := z - y$.

(2.2.2) If $z \neq 0$, update $T := T \cup \{z\}$.

Algorithm 3.3 is a simple combinatorial variant of a Buchberger type algorithm. We refer to [95] and [22] for earlier versions of this algorithm as well as other proofs of their correctness. We first illustrate the performance of Algorithm 3.3 on a small example.

Example 3.5. Consider the family of integer programs in 3 variables

$$\min\{c_1x_1 + c_2x_2 + c_3x_3 :\ x_1 + 2x_2 + 3x_3 = b,\ 0 \le x \le u,\ x \in {\mathbb{Z}_+}^3\},$$

with varying $b, u \in \mathbb{Z}_+$, $c \in \mathbb{R}^3$. Algorithm 3.3 starts with all the unit vectors,

$$T = \{\pm e^1, \pm e^2, \pm e^3\}.$$

Taking all sums of vectors of G gives rise after reduction to a new set

$$T \ = \ \{\pm e^1, \pm e^2, \pm e^3, \pm(e^1 - e^2), \pm(e^1 - e^3), \pm(e^2 - e^3)\}.$$

Note that for $i = 1, 2, 3$ the vectors $2e^i$ reduce to 0. Accordingly $(e^1 + e^2)$ can be reduced by $\pm e^1$ and a vector of the form $(e^i + e^3)$ is reducible by both e^i and by $\pm e^3$. With the updated set T we again perform the operation of taking all the sums of vectors of T and checking for reducibility. This yields after reduction an updated set

$$T \ = \ T_{old} \cup \{\pm(2e^1 - e^2), \pm(2e^1 - e^3), \pm(2e^2 - e^3), \pm(e^1 + e^2 - e^3)\}.$$

Denoting this set T by T_{old}, taking the sums of vectors $\pm(e^1 + [2e^1 - e^3])$, $\pm(e^1 + [-2e^1 + e^3])$ and $\pm([e^2 - e^3] + [2e^2 - e^3])$ yields three additional vectors that are irreducible and added to T. All other sums of vectors of T_{old} can be reduced by T to 0. Algorithm 3.3 terminates with the following set

$$\begin{array}{lll} T = \{ & \pm e^1, \pm e^2, \pm e^3, & \\ & \pm(e^1 - e^2), \pm(e^1 - e^3), \pm(e^2 - e^3), & \\ & \pm(2e^1 - e^2), \pm(2e^1 - e^3), \pm(2e^2 - e^3), \pm(e^1 + e^2 - e^3), & \\ & \pm(3e^1 - e^3), \pm(3e^2 - 2e^3), \pm(e^1 + e^3 - 2e^2) & \}. \end{array} \tag{77}$$

The Graver test set for this family of knapsack problems is

$$\{\pm(2e^1 - e^2), \pm(3e^1 - e^3), \pm(e^1 + e^3 - 2e^2), \pm(3e^2 - 2e^3), \pm(e^1 + e^2 - e^3)\}.$$

Theorem 3.21. *Algorithm 3.3 is finite. The set T that is returned by the algorithm contains the set of all irreducible vectors w.r.t. the matrix A.*

Proof. Let G denote the set of all irreducible elements w.r.t. A. Let $t \in G$. Let T^u denote the current set T of Algorithm 3.3 before the u-th. performance of Step (2). We remark that $\{\pm e^i :\ i = 1, \ldots, n\} \subseteq T^u$. Therefore, there exists a multiset $S = \{t^1, \ldots, t^k\} \subseteq T^u$ such that

$$t = t^1 + \ldots + t^k.$$

For every multiset $S_t = \{t^1, \ldots, t^k\} \subseteq T^u$ with $t = \sum_{i=1}^k t^i$, let

$$\phi(S_t) := \sum_{i=1}^k (||At^i||_1 + ||t^i||_1).$$

Let S_t^* denote a multiset such that $\phi(S_t^*)$ is minimal. Note that $t \in G$ if and only if t is irreducible. On account of Definition 3.4 t is irreducible if and only if for all decompositions of the form $t = \sum_{i=1}^k t^i$ the following condition holds,

$$||At||_1 + ||t||_1 < \sum_{i=1}^{k}(||At^i||_1 + ||t^i||_1).$$

We conclude,

$$\phi(S_t^*) > ||At||_1 + ||t||_1 \iff t \notin T^u.$$

However, if $t \notin T^u$, then there exist indices $i, j \in \{1, \ldots, k\}$ such that the vectors (t^i, At^i) and (t^j, At^j) lie in different orthants of $\mathbb{R}^{n+m}$. On account of the minimality of $\phi(S_t^*)$, $t^i + t^j$ is neither in T^u nor can $t^i + t^j$ be written as the sum of elements from T^u all of which reduce $t^i + t^j$. However, $z = t^i + t^j$ will be considered in the subsequent performance of Step (2.2.1) of the algorithm. Then z will be added to T^u and the value $\phi(S_t^*)$ will decrease by at least one. Since $\phi(S_t^*) > ||At||_1 + ||t||_1$ for all iterations of Step (2) in which $t \notin T^u$, the algorithm will detect t in a finite number of steps. These arguments apply to any irreducible vector. There is only a finite number of irreducible vectors, and hence, the algorithm is finite. $\square$

4 Group Relaxations, Corner Polyhedra and Subadditivity

Here we look at relaxations in which the nonnegativity constraints are dropped on a subset of the variables. We then present the Gomory group relaxation [35]. Though the subject of groups may be new to some, all the reader needs to understand are linear equations in integers under addition modulo integers. After discussing briefly algorithms based on the group relaxation, we study the corner polyhedron [36, 37], which is the convex hull of solutions to the group problem. The study of the structure of valid inequalities for the corner polyhedron indicates the importance of subadditivity. Finally we briefly consider solving (IP) for all possible vectors b, and the question of choosing on which sets of variables nonnegativity needs to be relaxed.

4.1 A family of relaxations and a canonical form

Given the integer program

$$IP(b) \qquad z = \min\{c^T x : Ax = b, x \in \mathbb{Z}_+^{m+n}\},$$

where A is an $m \times (m+n)$ integer matrix and b is an integer m vector, a natural idea is to drop the nonnegativity constraints on a subset $S \subseteq V$ of the variables, which leads to the relaxation

$$IP_S(b) \qquad z_S = \min\{c^T x : Ax = b, x_j \in \mathbb{Z}_+^1 \text{ for } j \in V \setminus S, x_j \in \mathbb{Z}^1 \text{ for } j \in S\}.$$

This can be viewed as a special case of a more general family of relaxations

$$IP_{K,\beta}(b) \qquad z_{K,\beta} = \min\{c^T x + \beta^T w : Ax + Kw = b, x \in \mathbb{Z}_+^{m+n}, w \in \mathbb{Z}^p\}$$

where K is an integer $m \times p$ matrix. In particular if $K = A_S$ and $\beta = c_S$, $IP_{K,\beta}(b)$ reduces to $IP_S(b)$, and when $K = \begin{pmatrix} 0 \\ I^{(m-1)} \end{pmatrix}$ and $\beta = 0$, $IP_{K,\beta}(b)$ reduces to a knapsack relaxation

$$z^{KN} = \min\{c^T x : a^1 x = b_1, x \in \mathbb{Z}_+^{m+n}\},$$

where a^1 denotes the first row of A.

Instead of working with $IP_{K,\beta}(b)$, we may consider its projection $X_K(b)$ on the space of x variables. Suppose that $p \leq m$ and there exists a dual feasible vector $u \in \mathbb{R}^m$ with $u^T A \leq c^T$ and $u^T K = \beta^T$. Now $IP_{K,\beta}(b)$ can be rewritten as

$$z_{K,\beta}(b) = u^T b + \min\{(c^T - u^T A)x : x \in X_K(b)\} \text{ where}$$

$$X_K(b) = \{x \in Z_+^{m+n} : Ax + Kw = b \text{ for some } w \in \mathbb{Z}^p\}.$$

To see the structure of $X_K(b)$ we make use of the Smith Normal Form of a matrix. Remember that a square integer matrix C is *unimodular* if $|\det C| = 1$, and if $x, y \in \mathbb{Z}_+^1 \setminus \{0\}$, and $x|y$ means that y is an integer multiple of x.

Theorem 4.1. *Given an $m \times p$ integer matrix K of rank $p \leq m$, there exist unimodular integer matrices R and C with R an $m \times m$ matrix, C a $p \times p$ matrix such that $RKC = \begin{pmatrix} \Delta \\ 0 \end{pmatrix}$ where the diagonal matrix Δ has diagonal elements $\delta_i \in \mathbb{Z}^1_+$ for $i = 1, \ldots, p$ with $\delta_1 \mid \delta_2 \mid \cdots \mid \delta_p$, and Δ is unique.*

Now we can derive a canonical representation of $X_K(b)$. Here a^j denotes the j^{th} column of A and $(\rho)_i$ is the i^{th} coordinate of the vector ρ.

Theorem 4.2.

$$X_K(b) = \{x \in \mathbb{Z}^{m+n}_+ : \textstyle\sum_{j=1}^{m+n} (Ra^j)_i x_j \equiv (Rb)_i \ (mod\ \delta_i) \ for\ i = 1, \ldots, p,$$

$$\textstyle\sum_{j=1}^{m+n} (Ra^j)_i x_j = (Rb)_i \ for\ i = p+1, \ldots, m\}$$

with $Ra^j, Rb \in \mathbb{Z}^m$ for $j = 1, \ldots, m+n$.

Proof. Observe that

$$RAx + RKw = Rb, x \in \mathbb{Z}^{m+n}_+, \ w \in \mathbb{Z}^p$$

can be rewritten as

$$RAx + RKCC^{-1}w = Rb, x \in \mathbb{Z}^{m+n}_+, C^{-1}w \in \mathbb{Z}^p$$

as C is unimodular. Now setting $v = -C^{-1}w$ and $\Delta' = \begin{pmatrix} \Delta \\ 0 \end{pmatrix}$, this becomes

$$RAx = Rb + \Delta' v, x \in \mathbb{Z}^{m+n}_+, v \in \mathbb{Z}^p.$$

where RA and Rb are integer as R is unimodular. □

Note that when $K = A_S$, it is more natural to look at the feasible region in the space of the variables $x_{V \setminus S}$. Now

$$X_{V\setminus S} = \{x_{V\setminus S} \in \mathbb{Z}^{|V\setminus S|}_+ : A_{V\setminus S} x_{V\setminus S} + A_S x_S = b \text{ for some } x_S \in \mathbb{Z}^{|S|}\}$$

$$= \{x_{V\setminus S} \in \mathbb{Z}^{|V\setminus S|}_+ : RA_{V\setminus S} x_{V\setminus S} \equiv Rb \ (\text{mod } \Delta')\},$$

where (mod Δ') means (mod δ_i) for rows $i = 1, \ldots, p$, and equality in rows $p+1, \ldots, m$.

4.2 Gomory's asymptotic group relaxation

Taking the integer program $IP(b)$, let $A = (A_B, A_N)$ with A_B an optimal LP basis. Now $IP(b)$ can be rewritten as

$$\begin{aligned} z = c_B^T A_B^{-1} b + \min \textstyle\sum_{j \in N} (c_j - c_B^T A_B^{-1} a_j) x_j \\ x_B + A_B^{-1} A_N x_N = A_B^{-1} b \\ x_B \in \mathbb{Z}^m_+, x_N \in \mathbb{Z}^n_+. \end{aligned}$$

where $N = \{1, \ldots, n\}$ is the set of nonbasic variables. The Gomory group relaxation [35] is obtained by dropping the nonnegativity constraint on x_B. It can be written as

$$\begin{array}{l} z^G = c_B^T A_B^{-1} b + \min \sum_{j \in N} \bar{c}_j x_j \\ \sum_{j \in N} (A_B^{-1} a^j) x_j \equiv A_B^{-1} b \text{ (mod 1)} \\ \qquad\qquad x \in \mathbb{Z}_+^n \end{array}$$

where $\bar{c}_j = (c_j - c_B^T A_B^{-1} a^j) \geq 0$ for $j \in N$. Equivalently using the description following Theorem 4.2, the feasible region can be written in canonical form

$$X_N = \{x_N \in \mathbb{Z}_+^n : RA_N x_N \equiv Rb \text{ (mod } \Delta)\}.$$

Example 4.1. Consider the IP

$$\begin{array}{lllllllllllll} z & = \min & - & 6x_5 & - & 4x_6 & - & x_1 & + & 4x_2 & & & + & 2x_4 & & \\ & & & 5x_5 & + & 3x_6 & + & 2x_1 & - & 4x_2 & + & x_3 & & & = & 5 \\ & & & x_5 & + & 2x_6 & - & 3x_1 & + & 5x_2 & & & + & x_4 & = & 2 \\ & & & & & & & x \in & & \mathbb{Z}_+^6. & & & & & & \end{array}$$

$x_B = (x_5, x_6)$ are optimal basic variables, so we obtain

$$\begin{array}{lllllllllllll} z & = \min & -\frac{44}{7} & & & + & \frac{3}{7}x_1 & + & \frac{6}{7}x_2 & + & \frac{8}{7}x_3 & + & \frac{16}{7}x_4 & & \\ & & x_5 & & & + & \frac{13}{7}x_1 & - & \frac{23}{7}x_2 & + & \frac{2}{7}x_3 & - & \frac{3}{7}x_4 & = & \frac{4}{7} \\ & & & x_6 & - & \frac{17}{7}x_1 & + & \frac{29}{7}x_1 & - & \frac{1}{7}x_3 & + & \frac{5}{7}x_4 & = & \frac{5}{7} & \\ & & & & & x \in & & \mathbb{Z}_+^6. & & & & & & & \end{array}$$

The Gomory group relaxation is thus

$$\begin{array}{llllllllllll} z^G & = -\frac{44}{7} + \min & & \frac{3}{7}x_1 & + & \frac{6}{7}x_2 & + & \frac{8}{7}x_3 & + & \frac{16}{7}x_4 & & \\ & & & \frac{13}{7}x_1 & - & \frac{23}{7}x_2 & & \frac{2}{7}x_3 & - & \frac{3}{7}x_4 & = & \frac{4}{7} \text{ (mod 1)} \\ & & - & \frac{17}{7}x_1 & + & \frac{29}{7}x_2 & - & \frac{1}{7}x_3 & + & \frac{5}{7}x_4 & = & \frac{5}{7} \text{ (mod 1)} \\ & & & & & & x \in & \mathbb{Z}_+^6, & & & & \end{array}$$

which simplifies to:

$$\begin{array}{lllllllllll} z^G & = -\frac{44}{7} + \min & \frac{3}{7}x_1 & + & \frac{6}{7}x_2 & + & \frac{8}{7}x_3 & + & \frac{16}{7}x_4 & & \\ & & \frac{6}{7}x_1 & + & \frac{5}{7}x_2 & + & \frac{2}{7}x_3 & + & \frac{4}{7}x_4 & = & \frac{4}{7}\text{(mod 1)} \\ & & \frac{4}{7}x_1 & + & \frac{1}{7}x_2 & + & \frac{6}{7}x_3 & + & \frac{5}{7}x_4 & = & \frac{5}{7}\text{(mod 1)} \\ & & & & & x \in & \mathbb{Z}_+^6. & & & & \end{array}$$

Now consider the Gomory group relaxation in canonical form. With $R = \begin{pmatrix} 1 & 0 \\ 3 & -1 \end{pmatrix}$ and $C = \begin{pmatrix} -1 & 3 \\ 2 & -5 \end{pmatrix}$, $\Delta = RA_B C = \begin{pmatrix} 1 & \\ & 7 \end{pmatrix}$, so we obtain

$$\begin{array}{lllllllllll} z^G & = -\frac{44}{7} + \min & \frac{3}{7}x_1 & + & \frac{6}{7}x_2 & + & \frac{8}{7}x_3 & + & \frac{16}{7}x_4 & & \\ & & 2x_1 & - & 4x_2 & + & 1x_3 & + & 0x_4 & = & 5 \text{ (mod 1)} \\ & & 9x_1 & - & 17x_2 & + & 3x_3 & - & x_4 & = & 13 \text{ (mod 7)} \\ & & & & & x \in & \mathbb{Z}_+^6, & & & & \end{array}$$

or

$$\begin{array}{rcccccccccl} z^G & = -\frac{44}{7} + \min & \frac{3}{7}x_1 & + & \frac{6}{7}x_2 & + & \frac{8}{7}x_3 & + & \frac{16}{7}x_4 & & \\ & & 2x_1 & + & 4x_2 & + & 3x_3 & + & 6x_4 & = & 6 \pmod 7 \\ & & & & & & x \in & \mathbb{Z}^6_+. & & & \end{array}$$

The canonical form that we have derived for the group problem is not surprising given that all finite commutative groups reduce to sets of integer vectors under addition modulo some given integer vector.

Theorem 4.3. *Every finite abelian (commutative) group G is isomorphic to the group consisting of integer p vectors under addition modulo $(\delta_1, \delta_2, \ldots, \delta_p)$ for some p with $\delta_i \in \mathbb{Z}^1_+ \setminus \{0, 1\}$ and $\delta_1 \mid \delta_2 \mid \cdots \mid \delta_p$. Such a group is denoted by $\mathbb{Z}_{\delta_1} \times \cdots \times \mathbb{Z}_{\delta_p}$.*

Thus we see that the canonical form provides an explicit representation of the group, and hence Gomory chose to speak of the "group relaxation" which has the general form

$$IP_G(g_0) \qquad \begin{array}{c} z^G = \min \sum_{j \in N} \bar{c}_j x_j \\ \sum_{j \in N} g_j x_j = g_0 \text{ in } G \\ x \in \mathbb{Z}^n_+. \end{array}$$

where $g_j \in G$ for $j \in N$ and $g_j x_j$ denotes $g_j + \cdots + g_j$ added x_j times with addition in the group.

4.3 Algorithms based on group relaxations

The group relaxation $IP_G(g_0)$ can be viewed as a shortest path problem in a graph with $|G| = \Pi_{i=1}^p \delta_i = |\det A_B|$ nodes. For each $g \in G$, there is an arc $(g, g + g_j)$ with cost (length) $\bar{c}_j$, and the problem is to find a shortest path from $0 \in G$ to $g_0 \in G$. In Figure 13, the shortest path from 0 to 2 gives an optimal solution to the group problem:

$$\min\{3x_1 + 7x_2 : 1x_1 + 3x_2 \equiv 2 \pmod 5,\ x \in \mathbb{Z}^2_+\}.$$

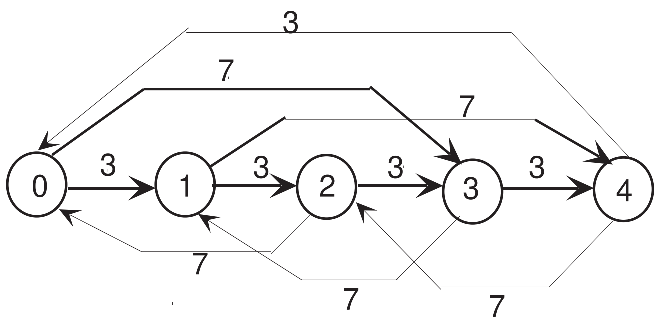

Figure 13: Shortest Path Group Problem

Proposition 4.4. *Using a shortest path algorithm, $IP_G(g_0)$ can be solved for all $g_0 \in G$ with $O(n|det\ A_B|)$ operations.*

Note that the original problem $IP(b)$ and the relaxations $IP_{K,\beta}(b)$ can also be viewed a priori as shortest path problems on an infinite graph whose nodes are the vectors $d \in \mathbb{Z}^m$. For $IP(b)$, there is an arc $(d, d + a^j)$ with cost c_j for all $d \in \mathbb{Z}^m$ and $j = 1, \ldots, m+n$, and the problem is to find a shortest path from 0 to b. To obtain $IP_{K,\beta}$, the original problem $IP(b)$ is relaxed by adding additional arcs $(d, d \pm k^i)$ with cost $\pm\beta_i$ for each column k^i of K and for all $d \in \mathbb{Z}^m$.

Something can also be said about the magnitude of solutions to the group problems.

Observation 4.1. *There exists an optimal solution $\overline{x}$ to the group problem $IP_G(g_0)$ with $\Pi_{j \in N}(1 + \overline{x}_j) \leq |det\ A_B|$.*

Obviously one hopes that a solution to the group problem $IP_G(g_0)$ provides a solution of the original problem $IP(b)$.

Observation 4.2. *If $\overline{x}_N$ solves the group problem $IP_G(g_0)$, then by construction $\overline{x}_B = A_B^{-1}b - A_B^{-1}A_N\overline{x}_N \in \mathbb{Z}^m$. If in addition $\overline{x}_B \geq 0$, then $(\overline{x}_B, \overline{x}_N)$ solves $IP(b)$.*

From this, we see that a solution $\overline{x}_N$ to $IP_G(g_0)$ leads to a solution of the original problem $IP(b)$ for an infinity of values of b.

Observation 4.3. *Let $D = \{d \in \mathbb{Z}^m : A_B^{-1}d \geq A_B^{-1}N\overline{x}_N\}$. Then for all $b' \in D$ for which $Rb'\ (mod\ \Delta) \equiv Rb\ (mod\ \Delta)$, $(x_B, x_N) = (A_B^{-1}b' - A_B^{-1}A_N\overline{x}_N, \overline{x}_N)$ is optimal in $IP(b')$.*

When the original problem is not solved, $IP(b)$ has been reduced to finding the least cost group solution x^* with $A_B^{-1}b - A_B^{-1}A_Nx^* \geq 0$. Based on this, and using the optimal value of the group problem for all right hand sides to obtain bounds, a branch-and-bound algorithm for $IP(b)$ was developed in [40]. A best bound variant of this approach based on finding the k^{th} best solution to the group problem appears in [100].

Another natural approach is to systematically tighten the relaxation every time that the solution to the relaxed problem is infeasible in $IP(b)$. Specifically if $(\overline{x}_B)_u = (A_B^{-1}b)_u - (A_B^{-1}A_N\overline{x})_u < 0$, one can drop column u from A_B, and create a new relaxation in which the nonnegativity of x_{B_u} is taken into account. However repeating this, we may be unlucky and return to the original problem $IP(b)$. An alternative that maintains finite group relaxations of the form $P_{\overline{G}}$ is just to increase the size of the group $\overline{G}$ so that $\overline{x}$ is no longer a feasible solution. One way to do this systematically is described in [7], see also [71].

Example 4.1 (cont.) An optimal solution of the group relaxation is

$$x_1 = x_2 = 1, x_3 = x_4 = 0.$$

Note that

$$\begin{pmatrix} x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} \frac{4}{7} \\ \frac{5}{7} \end{pmatrix} - A_B^{-1}A_Nx_N = \begin{pmatrix} \frac{4}{7} \\ \frac{5}{7} \end{pmatrix} - \begin{pmatrix} \frac{10}{7} \\ \frac{12}{7} \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

Thus the solution of the group relaxation is not feasible in the original problem, and thus $z_G = -\frac{44}{7} + \frac{9}{7} = -5$ is only a lower bound on $z(b)$.

4.4 The corner polyhedron

Given the basis A_B, the feasible region of the original problem $IP(b)$ can be viewed in the space of nonbasic variables as

$$\tilde{X}(b) = \{x_N \in \mathbb{Z}_+^n : x_B = A_B^{-1}b - A_B^{-1}A_N x_N \in \mathbb{Z}_+^m\}$$

whereas the feasible region of the group problem is

$$\begin{aligned}\tilde{X}_G(g_0) &= \{x_N \in \mathbb{Z}_+^n : x_B = A_B^{-1}b - A_B^{-1}A_N x_N \in \mathbb{Z}^m\} \\ &= \{x_N \in \mathbb{Z}_+^n : \sum_{j\in N} g_j x_j = g_0 \text{ in } G\}\end{aligned}$$

with $\tilde{X}(b) \subseteq \tilde{X}_G(g_0)$, and thus $\operatorname{conv}(\tilde{X}(b)) \subseteq \operatorname{conv}(\tilde{X}_G(g_0))$. This suggests the study of the *corner* polyhedron $\tilde{P}_G(g_0) = \operatorname{conv}(\tilde{X}_G(g_0))$, first introduced in [36]. Continuing in a remarkable paper [37], Gomory introduced several of the ideas that have now become standard in polyhedral combinatorics, projection onto faces, subadditivity, master polytopes, using automorphisms to generate one facet from another, some form of lifting, etc.

Observation 4.4. *Except for nonnegativity constraints $x_j \geq 0$ for $j \in N$, all facet-defining inequalities of $\tilde{P}_G(g_0)$ are of the form $\sum_{j\in N} \pi_j x_j \geq \pi_0$ with $\pi_j \geq 0$ for $j \in N$ and $\pi_0 > 0$.*

Gomory also introduced the idea of Master Polytopes for a given group G. Specifically let

$$X_G(g_0) = \{y \in \mathbb{Z}_+^{|G|} : \sum_{g\in G} g y_g = g_0 \text{ in } G\}$$

be a group problem in which every group element appears. Its convex hull

$$P_G(g_0) = \operatorname{conv}(X_G(g_0))$$

is called the *Master Polytope*. The following theorem says that the Master Polytope for G with right hand side g_0 provides the convex hull $\tilde{P}_G(g_0)$ for all instances of a group problem over the group G with right hand side g_0. Specifically this follows because $\tilde{X}_G(g_0) = X_G(g_0) \cap \{y : y_g = 0 \text{ for } g \notin \{g_1, \dots, g_n\}\}$, $y_g \geq 0$ defines faces of $P_G(g_0)$, and every face of an integral polyhedron is integral.

Theorem 4.5. *If $P_G(g_0) = \{y \in \mathbb{R}_+^{|G|} : \sum_{g\in G} \pi_g^k y_g \geq \pi_0^k \text{ for } k = 1, \dots, K\}$, then $\tilde{P}_G(g_0) = \{x \in \mathbb{R}_+^n : \sum_{j\in N} \pi_{g_j}^k x_j \geq \pi_0^k \text{ for } k = 1, \dots, K\}$.*

He also derived a characterization for the facets of the Master Polytope.

Theorem 4.6. *Let $\{t^q\}_{q=1}^Q$ be the vertices of $P_G(g_0)$, then $\sum_{g\in G} \pi_g y_g \geq 1$ is facet-defining if and only if $\pi \in \mathbb{R}^{|G|}$ is a basic feasible solution (vertex) of the polyhedron*

$$\begin{aligned}&\textstyle\sum_{g\in G} \pi_g t_g^q \geq 1, \quad q = 1, \dots, Q \\ &\pi_g \geq 0 \text{ for } g \in G \setminus \{0\}, \pi_0 = 0.\end{aligned}$$

Theorem 4.7. *The inequality $\sum_{g\in G}\pi_g y_g \geq 1$ is facet defining for $P_G(g_0)$ with $g_0 \neq 0$ if and only if $\pi \in \mathbb{R}^{|G|}$ is an extreme point of the polyhedron:*

$$\begin{aligned}
\pi_{g_1} + \pi_{g_2} &\geq \pi_{g_1+g_2} \text{ for } g_1, g_2 \in G \setminus \{0, g_0\}\\
\pi_g + \pi_{g_0-g} &= 1 \text{ for } g \in G \setminus \{0\}\\
\pi_g \geq 0 \quad &\text{for } g \in G \setminus \{0\}, \pi_0 = 0, \pi_{g_0} = 1.
\end{aligned}$$

Example 4.2. Take $G = \mathbb{Z}_6$ and the master set $X_G(3)$:

$$0y_0 + 1y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 \equiv 3 \pmod 6), y \in \mathbb{Z}^6_+.$$

The polyhedron of Theorem 4.7 takes the form

$$\begin{array}{lllll}
2\pi_1 & & & & \geq \pi_2\\
\pi_1 & +\pi_2 & & & = 1\\
\pi_1 & & \pi_4 & & \geq \pi_5\\
 & 2\pi_2 & & & \geq \pi_4\\
 & \pi_2 & & +\pi_5 & \geq \pi_1\\
 & & 2\pi_4 & & \geq \pi_2\\
 & & \pi_4 & +\pi_5 & = 1\\
 & \pi & \geq 0, & \pi_0 = 0, & \pi_3 = 1.
\end{array}$$

One extreme point is $(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5) = (0, \frac{1}{3}, \frac{2}{3}, 1, \frac{1}{3}, \frac{2}{3})$ giving the facet-defining inequality

$$0y_0 + \frac{1}{3}y_1 + \frac{2}{3}y + 1y_3 + \frac{1}{3}y_4 + \frac{2}{3}y_5 \geq 1.$$

Gomory also showed that because of the group structure, one vertex/facet could be used to obtain several vertices/facets for the same, or related group polyhedra. An automorphism is a one-to-one transformation from a group to itself preserving the addition structure of the group.

Proposition 4.8. *Suppose ϕ is an automorphism of G, then*

i) *if* $\sum_{g\in G}\pi_g y_g \geq 1$ *defines a facet of* $P_G(g_0)$,
then $\sum_{g\in G}\pi_{\phi^{-1}(g)} y_g \geq 1$ *defines a facet for* $P_G(\phi(g_0))$.

ii) *if* $t = (t_g)$ *is a vertex of* $P_G(g_0)$,
then $\bar{t} = (t_{\phi^{-1}(g)})$ *is a vertex of* $P_G(\phi(g_0))$.

Example 4.3. Consider the group $G = \mathbb{Z}_5$ of order 5 with $g_0 = 4$, and the associated corner polyhedron

$$P_{\mathbb{Z}_5}(4) = \text{conv}\{y \in \mathbb{Z}^5_+ : 0y_0 + 1y_1 + 2y_2 + 3y_3 + 4y_4 \equiv 4 \pmod 5)\}.$$

It is not difficult to show that $0y_0 + \frac{1}{4}y_1 + \frac{2}{4}y_2 + \frac{3}{4}y_3 + 1y_4 \geq 1$ is a facet-defining inequality, and that $t = (0, 1, 0, 1, 0)$ is a vertex.

Consider the automorphism (one-to-one transformation from G to G preserving the group addition) $\phi : G \to G$ with $\phi(g) = 2g \pmod 5$. Then the inverse $\phi^{-1} : G \to G$ is given by $\phi^{-1}(g) = 3g \pmod 5$, so

$$(\phi^{-1}(0), \phi^{-1}(1), \phi^{-1}(2), \phi^{-1}(3), \phi^{-1}(4)) = (0, 3, 1, 4, 2), \text{ and } \phi(g_0) = 3.$$

Now by i) of Proposition 4.8, the inequality $\pi_0 y_0 + \pi_3 y_1 + \pi_1 y_2 + \pi_4 y_3 + \pi_2 y_4 = 0y_0 + \frac{3}{4}y_1 + \frac{1}{4}y_2 + 1y_3 + \frac{1}{2}y_4 \geq 1$ defines a facet of $P_{\mathbb{Z}_5}(3)$.

Also by ii), $\bar{t} = (t_0, t_3, t_1, t_4, t_2) = (0, 1, 1, 0, 0)$ is a vertex of $P_{\mathbb{Z}_5}(3)$.

Facet defining inequalities for subgroups can also be lifted into facets for larger groups. Specifically a subgroup H of a group G is a subset of the elements of G that is closed under addition within the group (if $h_1, h_2 \in H$, then $h_1 + h_2 (\text{mod } \Delta) \in H$, and a homeomorphism ϕ from a group G to a subgroup H is a transformation preserving the addition structure of the group ($\phi(h_1 + h_2)(\text{mod } \Delta) \equiv \phi(h_1) + \phi(h_2)(\text{mod } \Delta)$).

Proposition 4.9. *If $\phi : G \to H$ is a homeomorphism into a subgroup H of G, and $\sum_{h \in H} \pi_h y_h \geq 1$ defines a facet of* $\text{conv}(X_H(\beta_0))$*, then if $\phi(g_0) = \beta_0 \neq 0$, $\sum_{g \in G} \pi_{\phi(g)} y_g \geq 1$ defines a facet of* $\text{conv}(X_G(g_0))$.

Example 4.4. Consider a group $G = \mathbb{Z}_2 \times \mathbb{Z}_4$ with subgroup $H = \mathbb{Z}_4$, and homeomorphism $\phi : G \to H$ given by $\phi(\alpha, \beta) = \beta$. If $(\alpha, \beta) \in G$, $\phi(\alpha, \beta) = \beta \in H$. Now as $y_1 + y_3 \geq 1$ defines a facet of $P_{\mathbb{Z}_4}(3) = \text{conv}(X_H(g_0))$, we obtain from Proposition 4.9 that $y_{(0,1)} + y_{(1,1)} + y_{(0,3)} + y_{(1,3)} \geq 1$ defines a facet of both $P_{\mathbb{Z}_2 \times \mathbb{Z}_4}(1, 3)$ and of $P_{\mathbb{Z}_2 \times \mathbb{Z}_4}(0, 3)$.

4.5 Subadditivity and duality

Here we limit our attention to cyclic (one-dimensional) groups $G = \mathbb{Z}_\delta$ for simplicity. We are still interested in the Master set

$$X_{\mathbb{Z}_\delta}(g_0) = \{y \in \mathbb{Z}^\delta_+ : \sum_{g=0}^{\delta-1} g y_g \equiv g_0 \ (\text{mod } \delta)\}$$

and its convex hull $P_{\mathbb{Z}_\delta}(g_0)$.

Definition 4.1. *$\pi : \{0, 1, \ldots, \delta - 1\} \to \mathbb{R}$ is subadditive on $\mathbb{Z}_\delta$ if $\pi(0) = 0$ and $\pi(u) + \pi(v) \geq \pi(u +_\delta v)$ for all $u, v \in \mathbb{Z}_\delta$, where $u +_\delta v$ denotes $u + v$ (mod δ).*

Theorem 4.7 can now be interpreted as saying that all the facet-defining inequalities arise from such a subadditive function. More generally, every subadditive function on $\mathbb{Z}_\delta$ leads to a valid inequality for $X_{\mathbb{Z}_\delta}(g_0)$.

Proposition 4.10. *If π is subadditive on $\mathbb{Z}_\delta$,*

$$\sum_{g=0}^{\delta-1} \pi(g) y_g \geq \pi(g_0)$$

is a valid inequality for $P_{\mathbb{Z}_\delta}(g_0)$.

Such functions also provide duals for the group problem.

Theorem 4.11. *Consider the (primal) group problem*

$$z_G(g_0) = \min\{\sum_{j\in N} \bar{c}_j x_j : \sum_{j\in N} g_j x_j = g_0 \text{ in } \mathbb{Z}_\delta, x \in \mathbb{Z}^n_+\}.$$

The problem

$$\begin{array}{rcl} w & = & \max \ \pi(g_0) \\ & & \pi(g_j) \le \bar{c}_j \text{ for } j \in N \\ & & \pi \text{ subadditive on } \mathbb{Z}_\delta \end{array}$$

is a strong dual with $z_G(g_0) = w$.

Proof. Let $\sum_{j\in N} \pi^k(g_j)x_j \ge \pi^k(g_0)$ for $k = 1, \ldots, K$ be the facet-defining inequalities of the corner polyhedron $P_G(g_0)$ with π^k subadditive on $\mathbb{Z}_\delta$. Now

$$z_G(g_0) = \min\{\sum_{j\in N} \bar{c}_j x_j : \sum_{j\in N} \pi^k(g_j)x_j \ge \pi^k(g_0) \text{ for } k = 1, \ldots, K, x \in \mathbb{R}^n_+\}.$$

Let $(u^1, \ldots, u^K)$ be a vector of optimal dual variables. Then clearly $\pi = \sum_{k=1}^K u^k \pi^k$ is a subadditive function on $\mathbb{Z}_\delta$ and is dual feasible with $\pi(g_0) = z_G(g_0)$. □

Now we examine how to generate a valid inequality off any constraint, and not just for groups corresponding to the integers modulo δ for some fixed δ. The results are from Gomory and Johnson [38].

Let I denote the unit interval $[0, 1)$ with addition modulo 1. Thinking of $\mathbb{Z}_\delta$ as a group with elements $\{0, \frac{1}{\delta}, \frac{2}{\delta}, \ldots, \frac{\delta-1}{\delta}\}$ under addition modulo 1, we can let δ tend to $+\infty$, and then we obtain an infinite group I whose elements lie in $[0, 1)$ with addition modulo 1.

Here we are interested in generating valid inequalities for the set

$$X_I(u_0) = \{x : \sum_{u\in I} ux(u) \equiv u_0 \ (\text{mod } 1), x(u) \ge 0 \text{ and integer, } x \text{ has finite support}\}.$$

We first extend our definition of a subadditive function.

Definition 4.2. $\pi : [0, 1) \to \mathbb{R}$ *is subadditive on I if $\pi(0) = 0$ and*

$$\pi(u) + \pi(v) \ge \pi(u +_1 v) \text{ for all } u, v \in [0, 1).$$

Now we will use subadditive functions for finite groups $\mathbb{Z}_\delta$ to obtain subadditive functions on I. In analogy with Proposition 4.10, we have

Proposition 4.12. *If π is subadditive on I,*

$$\sum_{u\in I} \pi(u)x(u) \ge \pi(u_0)$$

is a valid inequality for $X_I(u_0)$.

Proposition 4.13. Direct Fill In. *Let π be a subadditive function on $\mathbb{Z}_\delta$. Define $\pi(u)$ for $u \in [0,1) \setminus \{0, \frac{1}{\delta}, \ldots, \frac{\delta-1}{\delta}\}$ by*

$$\pi(u) = \delta[(u - L(u)\pi(R(u)) + (R(u) - u)\pi(L(u))]$$

where $L(u) = \frac{1}{\delta}\lfloor \delta u \rfloor$ and $R(u) = \frac{1}{\delta}\lceil \delta u \rceil$. Then π is subadditive on I.

Example 4.5. Take $\delta = 6$, and consider the subadditive function π derived for $\mathbb{Z}_6$ in Example 4.2 with $(\pi(0), \pi(\frac{1}{6}), \pi(\frac{2}{6}), \pi(\frac{3}{6}), \pi(\frac{4}{6}), \pi(\frac{5}{6})) = (0, \frac{1}{3}, \frac{2}{3}, 1, \frac{1}{3}, \frac{2}{3})$. Direct fill in immediately gives the function π where

$$\begin{array}{rcllcccc}
\pi(u) & = & 2u & \text{for } 0 & \leq & u & \leq & \frac{1}{2} \\
\pi(u) & = & 3 - 4u & \frac{1}{2} & \leq & u & < & \frac{2}{3} \\
\pi(u) & = & -1 + 2u & \frac{2}{3} & \leq & u & < & \frac{5}{6} \\
\pi(u) & = & 4 - 4u & \frac{5}{6} & \leq & u & < & 1.
\end{array}$$

In Figure 14 we show the values of the original function π on $\mathbb{Z}_5$, and the function π o I obtained by fill-in.

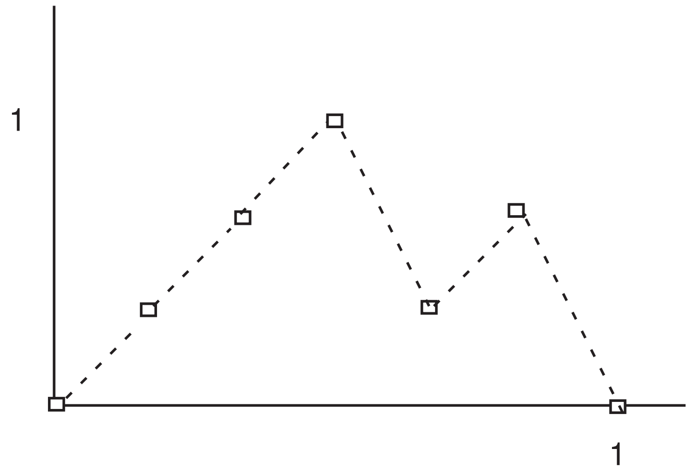

Figure 14: Constructing a subadditive function by fill-in

Now consider the constraint set

$$x_0 + 0.76x_1 - 0.35x_2 + 2.41x_3 = 4.49, \;\; x_0, x_1, x_2, x_3 \in \mathbb{Z}^1_+$$

coming from some arbitrary integer program. Relaxing the nonnegativity on x_0 leads to the set

$$X_I(0.49) = \{(x_1, x_2, x_3) \in \mathbb{Z}^3_+ : 0.76x_1 + 0.65x_2 + 0.41x_3 \equiv 0.49 \pmod 1)\}$$

and the subadditive function π just constructed immediately gives the valid inequality

$$\begin{array}{ccccccc}
\pi\,(0.76)\,x_1 & + & \pi\,(0.65)\,x_2 & + & \pi\,(0.41)\,x_3 & \geq & \pi\,(0.49)\,, \text{ or} \\
0.52x_1 & + & 0.30x_2 & + & 0.82x_3 & \geq & 0.98.
\end{array}$$

4.6 Solving $IP(b)$ for all values of $b \in \mathbb{Z}^m$

Consider again the integer program $IP(b)$. For a given $b \in \mathbb{Z}^m$, let A_B be an optimal LP basis. We have seen in Observation 4.3 that the solution x_N^* of the group relaxation

$$IP_B(b) \qquad \min\{c_B^T A_B^{-1} b + \bar{c}_N^T x_N : A_B^{-1} A_N x_N \equiv A_B^{-1} b \pmod 1, x_N \in \mathbb{Z}_+^n\}$$

solves $IP(d)$ for an infinity of values of $d \in \mathbb{Z}^m$, i.e. for all d such that $A_B^{-1} d = A_B^{-1} b$ (mod 1) and $x_B^* = A_B^{-1} d - A_B^{-1} A_N x_N^* \geq 0$. More generally, it is natural to ask what is the largest subset of columns $A_S \subset A_B$ for which the relaxation

$$IP_S(b) \quad z_S(b) = \min\{c^T x : Ax = b, x_j \in \mathbb{Z}_+^1 \text{ for } j \in V \setminus S, x_j \in \mathbb{Z}^1 \text{ for } j \in S\}$$

solves $IP(b)$. Solving $IP_S(b)$ will in turn provide a correction vector $x_{V\setminus S}^*$ that solves $IP(b)$ for many values of b.

In fact it can be shown that it suffices to solve a finite number of such problems $IP_{S_1}(b^1), \ldots, IP_{S_t}(b^t)$ in order to have a solution to $IP(b)$ for all b. Furthermore if for $u = 1, \ldots, t$, we take the subadditive functions $\{\pi^{k,u}\}_{k=1}^{K_u}$ describing the facets of the associated convex hulls for $u = 1, \ldots, t$, these suffice to describe the convex hulls of $IP(b)$ for all b, specifically

$$\sum_{j=1}^{m+n} \pi^{k,u}(a^j) x_j \geq \pi^{k,u}(b) \text{ for } k = 1, \ldots, K_u, \ u = 1, \ldots, t, \ x \in \mathbb{R}_+^{m+n}.$$

In other words

Theorem 4.14. *([101], [17]). For each integer $m \times (m+n)$ matrix A, there exists an integer $m' \times (m+n)$ matrix Q such that for any $b \in \mathbb{Z}^m$, there exists a function $q : \mathbb{Z}^m \to \mathbb{R}^{m'}$ such that*

$$\mathrm{conv}(\{x \in \mathbb{Z}_+^m : Ax = b\}) = \{x \in \mathbb{R}^{m+n} : Qx \geq q(b)\}.$$

In addition the size of the coefficients in Q is bounded by $(m+n)^{2(m+n)} f(A)$ where $f(A)$ is the maximum absolute value of the subdeterminants of A.

In [17], it is also shown that the difference between the optimal values $z(b) - z(b')$ cannot be too large.

Finally one might ask, given A and c fixed, but all possible b, for which sets S, it is necessary to solve $IP_S(b)$. Following Hosten and Thomas, a set $S \subset V$ is called *minimal* if for some b, the relaxation $IP_S(b)$ solves $IP(b)$, but $IP_{S'}(b)$ does not for any $S' \supset S$. Note that different vectors b and b' may imply that S is minimal, and alternatively different sets S and S' may be minimal due to the same vector b.

Theorem 4.15. *[50] Given A, c with rank$(A) = m$, if $S \subset V$ is a minimal set with $|S| < m$ for the family of problems $IP(b)$, there exists $v \in V \setminus S$ such that $S \cup \{v\}$ is minimal.*

4.7 Discussion

One important question is whether there exist versions of the integer programming algorithms presented in Section 2.1 that can be used with good results in practice. It should be noted that the main purpose of the algorithms by Lenstra [63], and by Lovász & Scarf [68], was to prove a theorem. No particular care was taken to ensure good performance in practice. We do believe, however, that some of the concepts discussed in this chapter can be used to design effective practical integer programming algorithms, and the studies by Cook et al. [20], and by Aardal et al. [1], [3] support this belief. We want to emphasize two such concepts here; branching on hyperplanes, and considering sublattices.

Branching on hyperplanes, or "integer branching", in directions in which the polytope is thin may reduce the number of nodes that one needs to evaluate in an enumeration tree quite drastically. One problem that needs to be dealt with is the amount of effort spent in each node. To compute search directions that are provably thin is quite time consuming, so heuristic algorithms are needed.

One of the features of the approach by Aardal et al. [3] is to consider a sublattice of $\mathbb{Z}^n$. Combining this idea with integer branching led to a decrease in the number of enumeration nodes of up to a factor of 10^4, compared to the number of nodes needed using branch-and-bound on the original formulation, [1].

The instances tackled by Cook et al. [20] and by Aardal et al. [1], were relatively small. If one applies Lovász' algorithm to such instances to obtain a reformulation such as (70), then the reduction only takes a couple of seconds. Therefore, the branching phase is the bottleneck. If one wants to solve medium size instances, then the reduction phase will be time consuming using the current versions of Lovász' algorithm. A faster basis reduction algorithm that can give similar guarantees as Lovász' algorithm would be extremely useful.

The practical use of test sets within an augmentation algorithm for general integer programming remains a challenging task for the future. It is certain that for medium sized integer programs an entire test set is far too big to be computed. This raises the question to design exponential subfamilies of a test set in combination with an efficient algorithm that checks whether at a given point one of the vectors of the subfamily is applicable and yields an improvement. If this is not the case, a certificate must be given that the current point is optimal. One way of deriving such a certificate might be to use the integer primal simplex algorithm of Young [102] or a variant of it.

The use of subadditive functions to generate valid inequalities for an arbitrary row of an integer program can be extended to mixed integer progams [38]. Given the recent computational interest in using Gomory's fractional cuts, mixed integer rounding inequalities and Gomory's mixed integer cuts, this reopens questions about the possible use of alternative subadditive functions to generate practically effective cutting planes. It is also natural to ask whether interesting higher dimensional functions can be found and put to use, see [71] for a natural 2-dimensional function.

The subadditive duality on $\mathbb{Z}_\delta$ has been generalized to a subadditive duality theory for general integer and mixed integer programs [51]. This raises the question of the conversion of subadditive functions on I into nondecreasing subadditive functions on $\mathbb{R}^n$ for use in arbitrary integer programs.

There are other non-standard approaches than those considered here. The area of constraint programming [97] has proved effective in tackling various problems including certain scheduling problems that are very difficult to solve as integer programs. The integration of CP and IP approaches is an area attracting considerable interest, though there is little to report in terms of mathematical results.

Acknowledgements

The authors would like to thank Bill Cook, Ravi Kannan, Arjen Lenstra, Hendrik Lenstra, Herb Scarf, Alexander Schrijver, and Rekha Thomas for enlightening discussions, comments on various versions of the manuscript, and for providing references.

References

[1] K. Aardal, R. E. Bixby, C. A. J. Hurkens, A. K. Lenstra and J. W. Smeltink (1999), Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances, in *Integer Programming and Combinatorial Optimization, 7th International IPCO Conference* (G. Cornuéjols, R. E. Burkard, G. J. Woeginger (eds.)), Lecture Notes in Computer Science 1610, pp 1–16, Springer-Verlag, Berlin Heidelberg.

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

[3] K. Aardal, C. Hurkens and A. K. Lenstra (1998), Solving a system of diophantine equations with lower and upper bounds on the variables, Research report UU-CS-1998-36, Department of Computer Science, Utrecht University, to appear in *Mathematics of Operations Research.*

[4] R. K. Ahuja, T. L. Magnanti and J. B. Orlin (1993), *Network flows: theory, algorithms, and applications*, Prentice Hall, Englewood Cliffs NJ.

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

[6] T. Becker, V. Weispfenning (1993), *Gröbner bases: A computational approach to commutative algebra*, Springer Verlag, New York.

[7] D.E. Bell and J.F. Shapiro (1977), A convergent duality theory for integer programming, *Operations Research* **25**, 419-434.

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

[9] J. Bourgain and V. D. Milman (1985). Sections euclidiennes et volume des corps symétriques convexes dans $\mathbb{R}^n$, *C. R. Acad. Sc. Paris* t. **300**, Série I, no 13, 435–438.

[10] W. Bruns and J. Gubeladze (1998), Normality and covering properties, preprint, University of Osnabrück.

[11] W. Bruns, J. Gubeladze and N.V. Trung (1997), Normal polytopes, triangulations, and Koszul algebras, *J. Reine Angew. Math.* **485**, 123 – 160.

[12] W. Bruns, J. Gubeladze, M. Henk, A. Martin and R. Weismantel (1998), A counterexample to an integer analogue of Carathéodory's Theorem, Preprint No. 32, Universität Magdeburg.

[13] B. Buchberger (1985), Gröbner bases: an algorithmic method in polynomial ideal theory, in N.K. Bose ed., *Multidimensional Systems Theory*, D. Reidel Publications, 184 - 232.

[14] H. Cohen (1996), *A Course in Computational Algebraic Number Theory*, Springer-Verlag, Berlin, Heidelberg.

[15] P. Conti and C. Traverso (1991), Buchberger algorithm and integer programming, *Proceedings AAECC-9* (New Orleans), Springer LNCS 539, 130 - 139.

[16] S. A. Cook (1971). The complexity of theorem-proving procedures, in *Proceedings of Third Annual ACM Symposium on Theory of Computing*, pp 151–158, ACM, New York.

[17] W. Cook, A.M.H. Gerards, A. Schrijver and E. Tardos (1986), Sensitivity Results in Integer Programming, *Mathematical Programming* **34**, 251-264.

[18] W. Cook, J. Fonlupt and A. Schrijver (1986), An integer analogue of Carathéodory's theorem, *J. Comb. Theory (B)* **40**, 63 - 70.

[19] W. Cook, A.M.H. Gerards, A. Schrijver, and É. Tardos (1986), Sensitivity theorems in integer programming problems, *Mathematical Programming* **34**, 63 - 70.

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

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

[22] G. Cornuéjols, R. Urbaniak, R. Weismantel and L.A. Wolsey (1997), Decomposition of integer programs and of generating sets, in *Algorithms – ESA '97* (R. Burkard, G. Woeginger (eds.)), Lecture Notes in Computer Science 1284, pp 92–103, Springer-Verlag, Berlin Heidelberg.

[23] J.G. van der Corput (1931), Über Systeme von linear-homogenen Gleichungen und Ungleichungen, *Proceedings Koninklijke Akademie van Wetenschappen te Amsterdam* **34**, 368 - 371.

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

[25] D. A. Cox, J. B. Little and D. O'Shea (1992), *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*, Springer-Verlag, New York.

[26] D. Dais, U.U. Haus and M. Henk (1998), On crepant resolutions of 2-parameter series of Gorenstein cyclic quotient singularities, *Results in Mathematics* **33**, 208 - 265.

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

[28] J. Edmonds and R. M. Karp (1972), Theoretical improvements in algorithmic efficiency for network flow problems, *Journal of the Association for Computing Machinery* **19**, 248 - 264.

[29] J. Edmonds and R. Giles (1977), A min-max relation for submodular functions on graphs, in Studies in Integer Programming, P.L. Hammer et al. eds, *Annals of Discrete Mathematics* **1**, 185 - 204.

[30] G. Ewald (1996), *Combinatorial Convexity and Algebraic Geometry*, Graduate Texts in Mathematics, Vol. **168**, Springer-Verlag.

[31] R.T. Firla and G.M. Ziegler (1999), Hilbert bases, unimodular triangulations, and binary covers of rational polyhedral cones, *Discrete Computational Geometry* **21**, 205–216.

[32] H. N. Gabow (1985), Scaling algorithms for network problems, *Journal of Computer and System Sciences* **31**, 148 - 168.

[33] F.R. Giles and W.R. Pulleyblank (1979), Total dual integrality and integer polyhedra, *Linear Algebra and Applications* **25**, 191 - 196.

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

[35] R.E. Gomory (1965)., On the relation between integer and non-integer solutions to linear programs, *Proceedings of the National Academy of Sciences* **53**, 260-265

[36] R.E. Gomory (1967), Faces of an integer polyhedron, *Proceedings of the National Academy of Sciences* **57**, 16-18 .

[37] R.E. Gomory (1969), Some polyhedra related to combinatorial problems, *Linear Algebra and its Applications* **2**, 451-558.

[38] R.E. Gomory and E.L. Johnson (1972), Some continuous functions related to corner polyhedra, *Mathematical Programming* **3**, 23-85.

[39] P. Gordan (1873), Über die Auflösung linearer Gleichungen mit reellen Coefficienten, *Math. Ann.* **6**, 23 - 28.

[40] G.A. Gorry, W.D. Northup and J.F. Shapiro (1973), Computational experience with a group theoretic integer programming algorithm, *Mathematical Programming* **4**, 171-192. .

[41] J. E. Graver (1975), On the foundations of linear and integer programming I, *Mathematical Programming* **8**, 207 - 226.

[42] M. Grötschel and L. Lovász (1995), Combinatorial optimization, in *Handbook of Combinatorics*, R. Graham, M. Grötschel and L. Lovász eds., North-Holland, Amsterdam.

[43] M. Grötschel, L. Lovász and A. Schrijver (1988), *Geometric algorithms and combinatorial optimization*, Springer-Verlag, Berlin.

[44] M. Grötschel, L. Lovász, A. Schrijver (1984), Geometric methods in combinatorial optimization, in: *Progress in Combinatorial Optimization* (W. R. Pulleyblank (ed.)), Academic Press, Toronto, pp 167–183.

[45] M. Henk and R. Weismantel (1997), The height of minimal Hilbert bases, *Results in Mathematics* **32**, 298 - 303.

[46] M. Henk and R. Weismantel (1997), Hilbert bases of cones related to simultaneous Diophantine approximations and linear Diophantine equations, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Preprint SC 97-29.

[47] M. Henk and R. Weismantel (1998), A theorem about minimal solutions of linear Diophantine equations, *Contributions to Algebra and Geometry*, to appear.

[48] Ch. Hermite (1850), Extraits de lettres de M. Ch. Hermite à M. Jacobii sur différents objets de la théorie des nombres. *Journal für die reine und angewandte Mathematik* 40, 261–278, 279–290, 291–307, 308–315, [reprinted in: Oevres de Charles Hermite, Tome I (É. Picard, ed.), Gauthier-Villars, Paris, 1905, pp 100–121, 122–135, 136–155, 155–163.]

[49] D. S. Hirschberg, C. K. Wong (1976), A polynomial-time algorithm for the knapsack problem with two variables, *Journal of the ACM* **23**, 147–154.

[50] S. Hosten and R.R. Thomas (1998), Standard Pairs and Group Relaxations in Integer Programming, Technical Report, Department of Mathematics, Texas A&M University, February 1998.

[51] E. L. Johnson (1980), *Integer Programming – Facets, Subadditivity and Duality for Group and Semi-Group Problems*, SIAM Publications.

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

[53] E. Kaltofen (1983), On the complexity of finding short vectors in integer lattices, in: *Computer Algebra: Proceedings of EUROCAL '83, European Computer Algebra Conference* (J. A. VanHulzen (ed.)), Lecture Notes in Computer Science 162, pp 236–244, Springer-Verlag, New York.

[54] R. Kannan (1980), A polynomial algorithm for the two-variable integer programming problem, *Journal of the ACM* **27**, 118–122.

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

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

[57] R. Kannan and L. Lovász (1986), Covering minima and lattice point free convex bodies, in: *Foundations of Software Technology and Theoretical Computer Science* (K. V. Nori (ed.)), Lecture Notes in Computer Science 241, pp 193–213.

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

[59] R. M. Karp (1972), Reducibility among combinatorial problems, in *Complexity of Computer Computations* (R. E. Miller and J. W. Thatcher, (eds.)), pp 85–103, Plenum Press, New York.

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

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

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

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

[64] 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`

[65] J. Liu (1991), Hilbert bases with the Carathéodory property, PhD. Thesis, Cornell University.

[66] J. Liu, L.E. Trotter, Jr., and G.M. Ziegler (1993), On the height of the minimal Hilbert basis, *Results in Mathematics* **23**, 374–376.

[67] L. Lovász (1986), *An Algorithmic Theory of Numbers, Graphs and Convexity*, CBMS-NSF Regional Conference Series in Applied Mathematics Vol 50. SIAM, Philadelphia.

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

[69] T. McCormick and A. Shioura (1996), A minimum ratio cycle canceling algorithm for linear programming problems with application to network optimization, Manuscript.

[70] D. Micciancio (1998). The shortest vector in a lattice is hard to approximate to within some constant (preliminary version), *Electronic Colloquium on Computational Complexity*, Report No. 16. ???

[71] G. L. Nemhauser and L. A. Wolsey (1988), *Integer and Combinatorial Optimization*, Wiley, New York.

[72] T. Oda (1988), Convex bodies and algebraic geometry. An Introduction to the theory of toric varieties, *Ergebnisse der Mathematik und ihrer Grenzgebiete* 3. Folge, Bd. 15, Springer-Verlag.

[73] A. M. Odlyzko (1984), Cryptanalytic attacks on the multivariate knapsack cryptosystem and on Shamir's fast signature scheme, *IEEE Transactions on Information Theory IT-30* **4**, 584–601.

[74] L. Pottier (1991), Minimal solutions of linear diophantine systems: bounds and algorithms, Proceedings RTA (Como), LNCS 488, Springer.

[75] H. Röck (1980), Scaling techniques for minimal cost network flows, in *Discrete Structures and Algorithms*, Carl Hanser, München, 181 - 191.

[76] H. E. Scarf (1981), Production sets with indivisibilities, Part I: Generalities, *Econometrica* **49**, 1 - 32.

[77] H. E. Scarf (1981). Production sets with indivisibilities – Part I: Generalities, *Econometrica* 49, 1–32. Part II: The case of two activities, ibid., 395–423.

[78] H. E. Scarf (1986), Neighborhood systems for production sets with indivisibilities, *Econometrica* **54**, 507 - 532.

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

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

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

[82] C. P. Schnorr and H. H. Hörner (1995), Attacking the Chor-Rivest Cryptosystem by improved lattice reduction, in *Advances in Cryptology – EUROCRYPT '95* (L.C. Guillou, J.-J Quisquater (eds.)), Lecture Notes in Computer Science 921, pp 1–12.

[83] A. Schrijver (1986), *Theory of linear and integer programming*, Wiley, Chichester.

[84] A. Schulz, R. Weismantel and G. Ziegler (1995), 0/1 integer programming: optimization and augmentation are equivalent, in *Lecture Notes in Computer Science 979*, Springer, 473–483.

[85] A. Schulz and R. Weismantel (1999), An oracle-polynomial time augmentation algorithm for integer programming, in *Proc. of the 10th ACM-SIAM Symposium on Discrete Algorithms, Baltimore, USA*, 967–968.

[86] A. Sebö (1990), Hilbert bases, Carathéodory's theorem and combinatorial optimization, in Proc. of the IPCO conference, Waterloo, Canada , 431- 455.

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

[88] V. Shoup, NTL: A Library for doing Number Theory, Department of Computer Science, University of Wisconsin-Madison.
`http://www.shoup.net/`

[89] B. Sturmfels (1996), *Gröbner bases and convex polytopes*, University Lecture Series, Vol. 8, AMS.

[90] B. Sturmfels and R. Thomas (1997), Variation of cost functions in integer programming, *Mathematical Programming* **77**, 357 - 388.

[91] B. Sturmfels, R. Weismantel and G. Ziegler (1995), Gröbner bases of lattices, corner polyhedra and integer programming, *Beiträge zur Geometrie und Algebra* **36**, 281 - 298.

[92] R. Thomas (1995), A geometric Buchberger algorithm for integer programming, *Mathematics of Operations Research* **20**, 864 - 884.

[93] R. Thomas (1994), Gröbner basis methods for integer programming, PhD. Thesis, Cornell University.

[94] R. Thomas and R. Weismantel (1997), Truncated Gröbner bases for integer programming, *Applicable Algebra in Engineering, Communication and Computing* **8**, 241 - 257.

[95] R. Urbaniak, R. Weismantel and G. Ziegler (1997), A variant of Buchberger's algorithm for integer programming, *SIAM Journal on Discrete Mathematics* **1**, 96 - 108.

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

[97] P. van Hentenryck (1989), *Constraint Satisfaction in Logic Programming*, MIT Press, Cambridge, Mass.

[98] C. Wallacher (1992), Kombinatorische Algorithmen für Flußprobleme und submodulare Flußprobleme, PhD. Thesis, Technische Universität zu Braunschweig.

[99] X. Wang (1997), A new implementation of the generalized basis reduction algorithm for convex integer programming, PhD Thesis, Yale University.

[100] L.A. Wolsey(1973), Generalized dynamic programming methods in integer programming, *Mathematical Programming* **4**, 222-232.

[101] L.A. Wolsey (1981), On the b-hull of an integer program, *Discrete Applied Mathematics* **3**, 193-201.

[102] R.D. Young (1965), A primal (all integer), integer programming algorithm, *Journal of Research of the National Bureau of Standards* **69b**, 213-250.