Lattice basis reduction and integer programming

Karen Aardal*

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 [32] who proved that the integer 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 [31]. Later, Grötschel, Lovász, & Schrijver [17], Kannan [25], and Lovász & Scarf [35] 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 [10] 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 [21].

The purpose of this section is to review and explain how basis reduction can be used both theoretically and computationally in integer programming. To begin with, we define a lattice and a lattice basis. A reduced basis is a basis with relatively short and nearly orthogonal vectors. Two algorithms for finding a reduced basis are described in Section 1. First, the reduction algorithm by Lovász, as presented by Lenstra, Lenstra, & Lovász [31], is described in Section 1.1. Lovász' algorithm works with Euclidean norms, whereas the algorithm by Lovász & Scarf [35], presented in Section 1.2, works with a norm related to a given convex set. In Section 1 we also discuss some recent implementations.

The main ideas behind the integer programming algorithms by Lenstra [32], Grötschel, Lovász, & Schrijver [17], Kannan [25], and Lovász & Scarf [35] described in Section 2 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

*aardal@cs.uu.nl. Department of Computer Science, Utrecht University, P.O. Box 80089, 3508 TB Utrecht. Research partially supported by the ESPRIT Long Term Research Project nr. 20244 (Project ALCOM-IT: *Algorithms and Complexity in Information Technology*), by the project TMR-DONET nr. ERB FMRX-CT98-0202, both of the European Community, and by NSF through the Center for Research on Parallel Computation, Rice University, under Cooperative Agreement No. CCR-9120008.

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 Section 3 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 et al. [2], [3]. Their algorithm is outlined in Section 4. The basic idea behind the algorithms discussed in Sections 3 and 4 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.

It should be pointed out that basis reduction has been used in other fields such as computational algebra and number theory, but reviewing these other topics is outside the scope of our chapter.

For the reader wishing to study lattices and integer programming in more detail we refer to the articles mentioned in this introduction, to the survey article by Kannan [24], and to the textbooks by Lovász [34], Schrijver [44], Grötschel, Lovász, & Schrijver [18], Nemhauser & Wolsey [37], and Cohen [8]. In these references, and in the article by Lenstra, Lenstra, & Lovász [31], several applications of basis reduction, other than integer programming, are mentioned, 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 $\alpha_1, \alpha_1, \ldots, \alpha_n$.

A slightly different version of this manuscript will appear as part of the survey paper by Aardal, Weismantel and Wolsey [4].

1 Two basis reduction algorithms

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 *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 $b_1, \ldots, b_l$,*

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

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

Definition 2 *An integer nonsingular matrix U with $det(U) = \pm 1$ is called* unimodular.

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 *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 *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 [31] 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. Let $|| \; ||$ denote the Euclidean length in $\mathbb{R}^n$, and let x^T denote 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.

Definition 3 *The Gram-Schmidt process derives orthogonal vectors b_j^*, $1 \leq j \leq l$, from 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 Here we illustrate the Gram-Schmidt vectors obtained by applying the orthogonalization procedure given in Definition 3 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.
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. ■

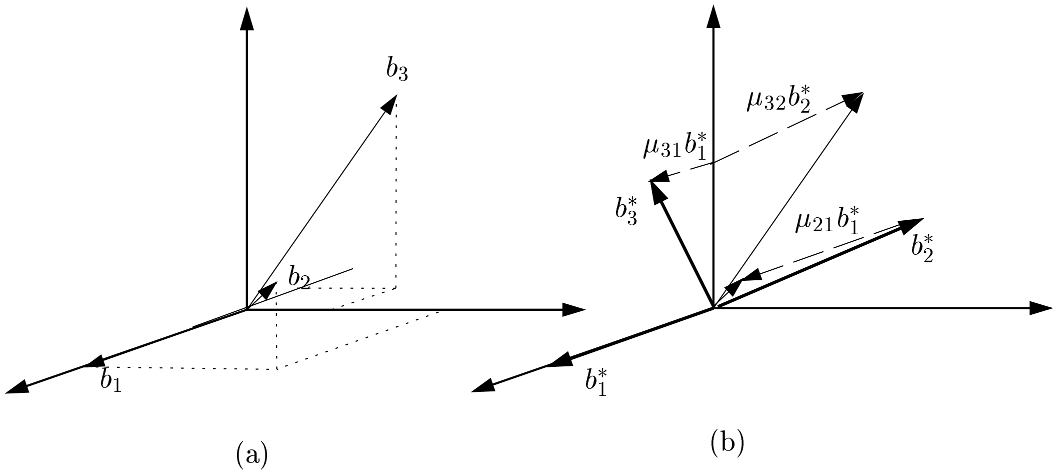

Figure 1:

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 4 *[31]. A basis $b_1, b_2, \ldots, b_l$ is called* reduced *if*

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

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

Before interpreting Conditions (5) and (6) we define $\lceil a \rfloor$ as $\lceil a \rfloor = \lceil a - \frac{1}{2} \rceil$, i.e., $\lceil a \rfloor$ is the nearest integer to a, where we round up if the fraction is equal to one half. 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 $||b_j^*|| \leq ||b_j||$. 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 [31]. First compute the Gram-Schmidt vectors b_j^*, $1 \leq j \leq l$ and the numbers μ_{jk}, $1 \leq k < j \leq l$. Initialize $i := 2$. Perform, if necessary, a size reduction to obtain $|\mu_{i,i-1}| \leq 1/2$. 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 $|\mu_{im}| \leq 1/2$ 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 2 *[31]. 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 [31] is $O(n^4 \log \beta)$, and the integers on which these operations are performed each have binary length $O(n \log \beta)$.*

In terms of bit operations, Theorem 2 implies that Lovász' basis reduction algorithm has a running time of $O(n^6 (\log \beta)^3)$ using classical algorithms for addition and multiplication. 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 [22], and Odlyzko [38].

Example 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.

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

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

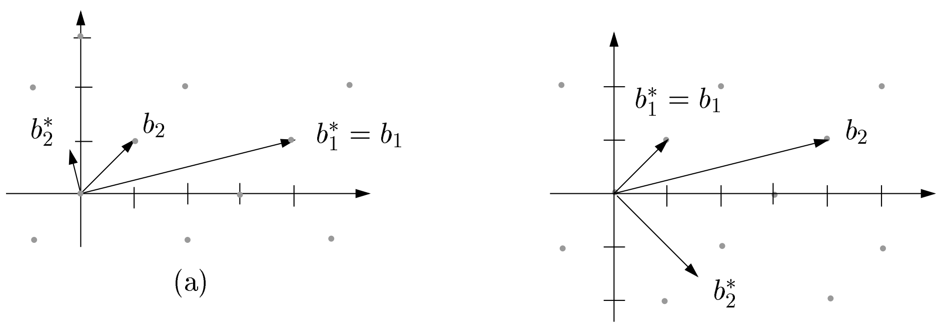

Figure 2:

and hence this basis is reduced, see Figure 3.

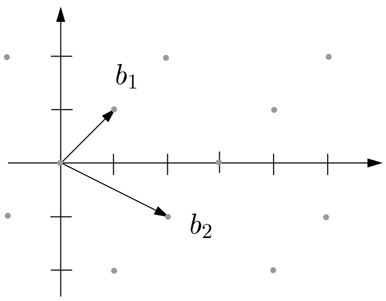

Figure 3: The reduced basis.

■

Let W be the vector space spanned by the lattice L, and let B_W be an orthonormal 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 $B = (b_1, \ldots, b_m)$ be a basis for the lattice $L \subset \mathbb{R}^n$, with $m \leq n$, and let $b_1^*, \ldots, b_m^*$ be the vectors obtained from applying the Gram-Schmidt orthogonalization procedure, see Definition 3, to $b_1, \ldots, b_m$.

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

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

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

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 $\det(L) = |\det(b_1, \ldots, b_n)|$. Note that the determinant of a lattice depends only on the lattice and not on the choice of basis (cf. Observation 1, and expression (9)). The determinant of $\mathbb{Z}^n$ is equal to one.

In Propositions 3 and 5 below we assume that the lattice L is full-dimensional.

Proposition 3 *[31]. Let $b_1, \ldots, b_n$ 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 [19] proved that each lattice $L \subset \mathbb{R}^n$ has a basis $b_1, \ldots, b_n$ 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 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 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 4 *[32]. Assume that $b_1, \ldots, b_n$ is a basis such that (10) holds, and that, after possible reordering, $||b_n|| = \max_{1 \leq j \leq n}\{||b_j||\}$. 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*

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

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

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

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 $h \geq c_1^{-1} ||b_n||$. From the definition of h we have $h \leq ||b_n||$, 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 4 plays a crucial role in the algorithm of H. W. Lenstra, Jr., that is described in Section 2.1.

Proposition 5 *[31]. 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*

$$||b_1||^2 \leq 2^{n-1} ||x||^2 \text{ for all } x \in L, x \neq 0, \tag{14}$$

$$||b_j||^2 \leq 2^{n-1} \max\{||x_1||^2, ||x_2||^2, \ldots, ||x_t||^2\} \text{ for } 1 \leq j \leq t. \tag{15}$$

Inequality (14) implies that the first reduced basis vector b_1 is an approximation of the shortest nonzero vector in L. Kannan [25] 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 [36] 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 [15] 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 6 *[31]. 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*

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

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 [42] for a more detailed overview. Schnorr [40] 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 [45] 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 [42] 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 [40], [41] 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 [43] 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 [33], developed at TH Darmstadt, and NTL - A Library for doing Number Theory [46], 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 $F(c) = \inf\{\lambda \geq 0 : c/\lambda \in C\}$. Lovász & Scarf [35] 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 $b_1, \dots, b_n$ be a basis for $\mathbb{Z}^n$, and let C_j be the projection of C on the orthogonal complement of $b_1, \dots, b_{j-1}$. We have that $c = \beta_j b_j + \cdots + \beta_n b_n \in C_j$ if and only if there exist $\alpha_1, \dots, \alpha_{j-1}$ 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:

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

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

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

In expression (18), note that only vectors z that are orthogonal to the basis vectors $b_1, \dots, b_{j-1}$ 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 5 *A basis $b_1, \dots, b_n$ is called* reduced *if*

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

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

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

Definition 6 *A basis $b_1, \ldots, b_n$, not necessarily reduced, is called* proper *if*

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

Remark: 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 [31] (cf. Definition 4).

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 7 *[35]. 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 $U = \max_j F_j(b_j)$, 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 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 8 *[35]. Let $0 < \varepsilon < \frac{1}{2}$, and let $b_1, \ldots, b_n$ be a Lovász-Scarf reduced basis. Then*

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

We can use this theorem to obtain a result analogous to (14) of Proposition 5.

Proposition 9 *[35]. Let $0 < \varepsilon < \frac{1}{2}$, and let $b_1, \ldots, b_n$ be a Lovász-Scarf reduced basis. Then*

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

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 6). $\nu_1, \ldots, \nu_n$ 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 10 *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*

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

The first reduced basis vector is an approximation of the shortest lattice vector (Proposition 9). 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 2.3. 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., $c = \alpha_1 b_1 + \cdots + \alpha_n b_n$, where $\alpha_j \in \mathbb{Z}$. We have

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

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 $|\alpha_n|$:

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

where the last inequality is obtained by applying Theorem 8 iteratively. Notice that the bound on α_n is polynomial for fixed n. In a similar fashion we can obtain a bound on α_j for $n-1 \geq j \geq 1$. Suppose that we have chosen multipliers $\alpha_n, \dots, \alpha_{j+1}$ and that we want to determine a bound on α_j. Let γ^* be the value of γ that minimizes $F_j(\alpha_n b_n + \cdots + \alpha_{j+1} b_{j+1} + \gamma b_j)$. If this minimum is greater than $F_1(b_1)$, then there does not exist a vector c, with $\alpha_n, \dots, \alpha_{j+1}$ fixed such that $F_1(c) \leq F_1(b_1)$, since in that case $F_1(b_1) < F_j(\alpha_n b_n + \cdots + \alpha_{j+1} b_{j+1} + \gamma^* b_j) \leq F_j(\alpha_n b_n + \cdots + \alpha_j b_j) = F_j(c) \leq F_1(c)$, which yields a contradiction. If the minimum is less than or equal to $F_1(b_1)$, then we can obtain the bound:

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

Hence, we obtain a search tree that has at most n levels, and, given the bounds on the multipliers α_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 [10], and by Wang [47]. Cook et al. used generalized basis reduction to derive a heuristic version of the integer programming algorithm by Lovász and Scarf (see Section 2.3) to solve difficult integer network design instances. Wang solved both linear and non-linear 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 2.3.

2 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 [28] 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 [9] 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 [20], and by Kannan [23]. Scarf [39] 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. [32]. Below we first illustrate in Example 3 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 [32] and the algorithm of Grötschel, Lovász, & Schrijver [17], which are both based on Lovász' basis reduction algorithm [31], and, finally, the algorithm of Lovász & Scarf [35], 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 [14] 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 3 Consider the 2-dimensional polytope in Figure 4. If we use branch-and-bound

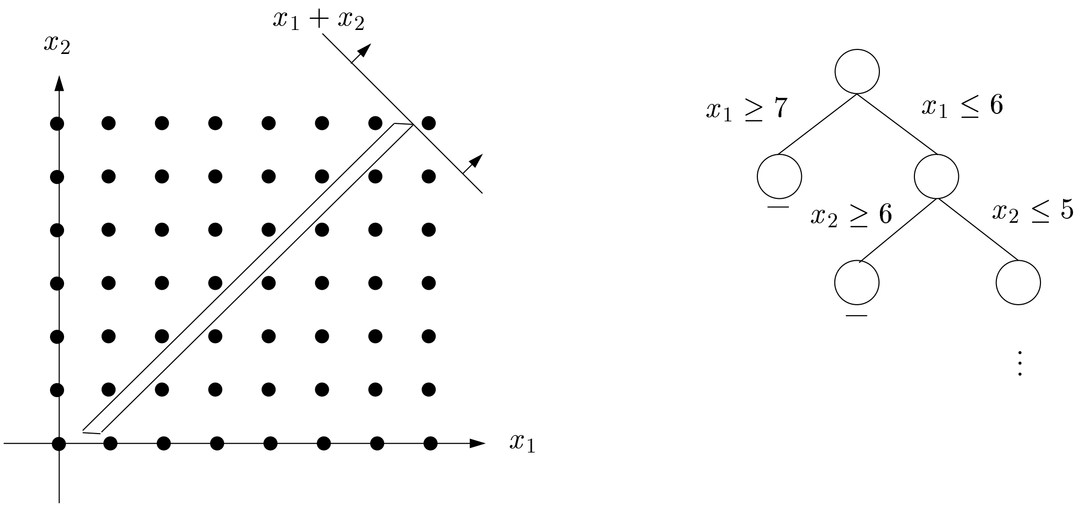

Figure 4: A difficult type of instance for branch-and-bound

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. ■

2.1 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 3 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

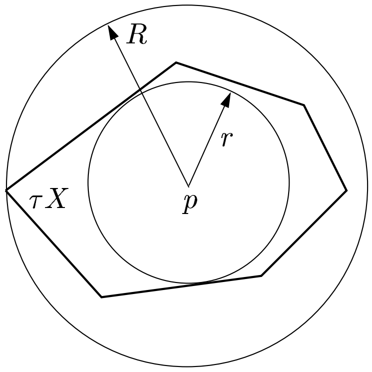

Figure 5:

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 11 *[32]. 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 11 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 4 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 4

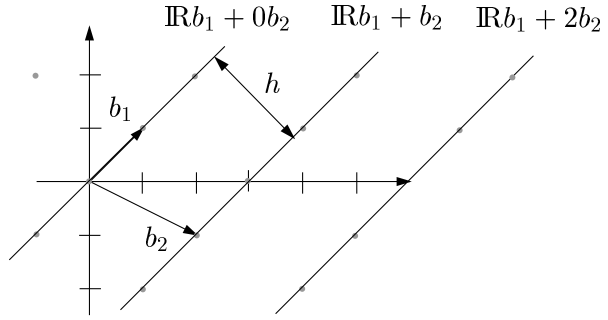

Figure 6:

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

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

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 4), 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 [32] 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 [32] 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 [32] 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 [25] 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.

2.2 The algorithm of Grötschel, Lovász, and Schrijver

Grötschel, Lovász, & Schrijver [17] 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 11, 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 12 *[17]. 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: 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 [44]. Notice that the algorithm referred to in Theorem 12 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 [16] 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 [44] 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 $//b_n// = \max_{1 \leq j \leq n}\{//b_j//\}$. After reordering, inequality (41) still holds. Suppose that the vector $y \in \mathbb{Z}^n$, which can be found by applying Lemma 11 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 $1/2\sqrt{n}//b_n// \geq //p-y// > 1/(n+1)$ 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 [44] pp 257–258, that if we consider a vector z such that $z^T D^{-1} z \leq 1$, then

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

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

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

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 12. If we apply the algorithm implied by Theorem 12, 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 12. The search tree has at most n levels and each level has polynomially many nodes if the dimension is fixed.

2.3 The algorithm of Lovász and Scarf

The integer programming algorithm of Lovász & Scarf [35] 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 7 *The* width *of the polytope X in the nonzero direction c is determined as*

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

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 13 and 14 below, and the definition of a so-called *generalized Korkine-Zolotarev basis*. Let b_j, $1 \leq j \leq n$ 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 $b_j, 1 \leq j \leq n$, see Definition 6 for the definition of a proper basis. The notion of a generalized KZ basis was introduced by Kannan & Lovász [26], [27]. Kannan & Lovász [26] gave an algorithm for computing a generalized KZ basis in polynomial time for fixed n.

Theorem 13 *[27]. 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, $1 \leq j \leq n$ is a generalized Korkine-Zolotarev basis. There exists a universal constant c_0 such that*

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

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 [7], is equal to $\frac{c_{\mathrm{BM}}^n}{n^n}$, where c_{BM} is a constant depending only on n. In Theorem 13 we have $c_0 = \frac{4}{c_{\mathrm{BM}}}$. See also the remark below.

Theorem 14 *[27]. 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 $\rho = \sum_{j=1}^{n} F_j(b_j) \leq 1$, 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 $F(c) \geq c_0 \cdot n \cdot (n+1)/2$, then $\rho_{\mathrm{KZ}} \leq 1$, which by Theorem 14 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 9, $F(b_1) \leq (\frac{1}{2} - \varepsilon)^{1-n} F(c)$. 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 [35], proof of Theorem 10).

Remark: Lagarias, Lenstra, & Schnorr [29] 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 $L^* = \{w \in W : w^T v \text{ is an integer for all } v \in L\}$. The bounds are given in terms of 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 [26], [27] 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 13 above. There are many interesting results on properties of various lattice constants. Many of them are described in the survey by Kannan [24], and will not be discussed further here.

Example 4 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.

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. 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.

■

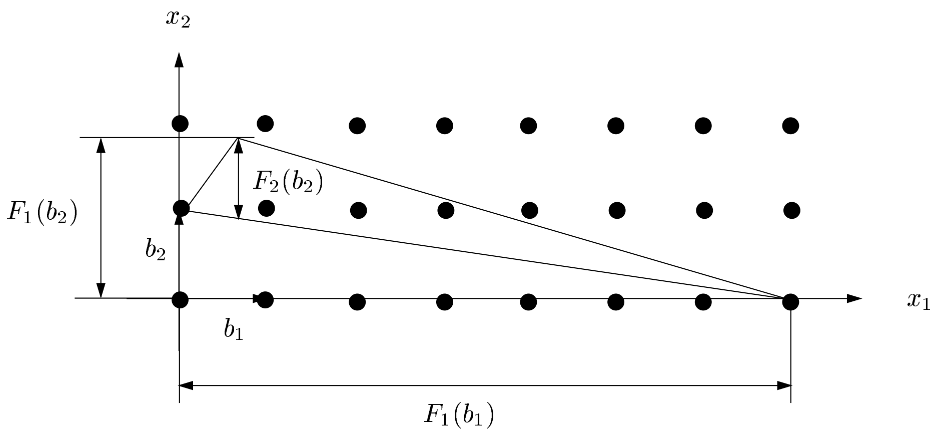

Figure 7:

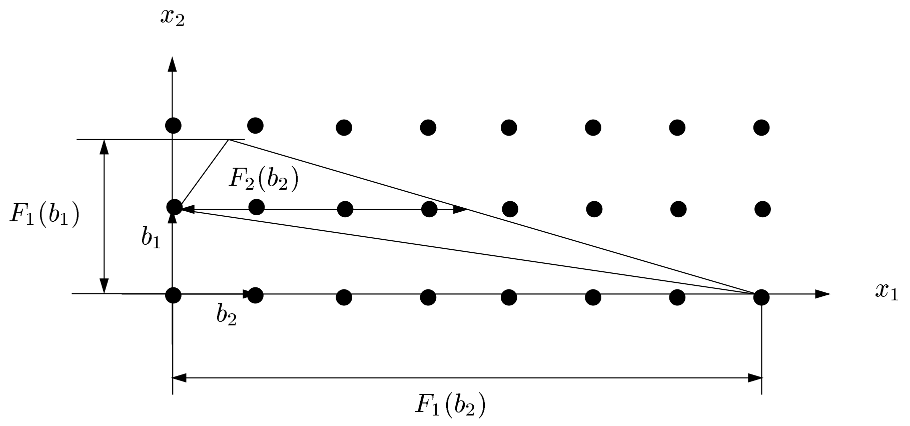

Figure 8:

3 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 [21].

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

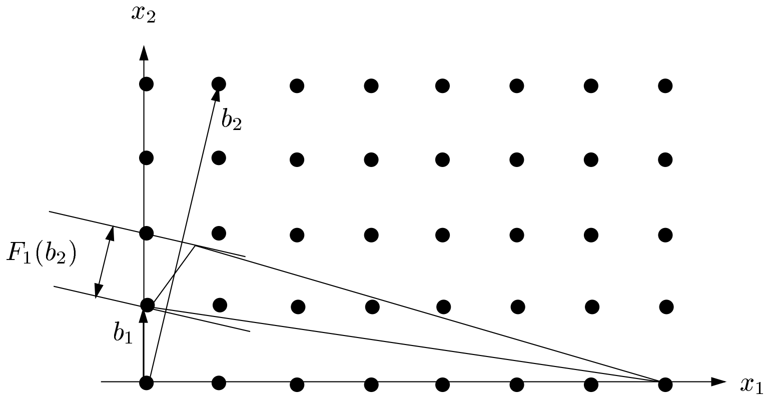

Figure 9:

subset sum problem should be hard to solve in order to obtain a secure transmission.

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

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

The density, as defined above, is an approximation of the information rate at which bits are transmitted. The interesting case is $d(a) \leq 1$, 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 [30] 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:

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

where y 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 $y = 1$ and $ax - a_0 y = 0$. 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 $y \in \mathbb{Z}$. We obtain

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

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 λ for some fixed constant λ, for $1 \le j \le n$. If such a reduced basis vector is found, check if the vector $x_j = b'_{jk}/\lambda$ 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 15 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 $\sum_{j=1}^n x_j \le \frac{n}{2}$. The reason for this assumption is that either $\sum_{j=1}^n x_j \le \frac{n}{2}$, or $\sum_{j=1}^n x'_j \le \frac{n}{2}$, where $x'_j = (1 - x_j)$, and since algorithm SV is run for both cases, one can perform the analysis for the vector that does satisfy the assumption. Let $x^e = (x_1, \ldots, x_n, 0)$. Let the sample space $\Lambda(A, x^e)$ of lattices be defined to consist of all lattices L_{a,a_0} generated by the basis (52) such that

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

and

$$a_0 = \sum_{j=1}^n a_j x_j^e. \tag{55}$$

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 15 *[30]. Let x^e be a 0-1 vector for which $\sum_{j=1}^n x_j^e \le \frac{n}{2}$. If $A = 2^{\beta n}$ for any constant $\beta > 1.54725$, then the number of lattices L_{a,a_0} in $\Lambda(A, x^e)$ that contain a vector v such that $v \neq kx^e$ for all $k \in \mathbb{Z}$, and such that $||v||^2 \le \frac{n}{2}$ 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 15 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 $\Lambda(A, x^e)$ for which x^e 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

$\Lambda(A, x^e)$ have x^e 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 [13] proposed two ways of improving Theorem 15. 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

$$L'_{a,a_0} = \{(x_1 - \frac{1}{2}y, \ldots, x_n - \frac{1}{2}y,\ N(ax - a_0y))^T\}, \tag{57}$$

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 $x^e = (x_1, \ldots, x_n, 0)$. The vector x^e does not belong to the lattice L', but the vector $w = (w_1, \ldots, w_n, 0)$, where $w_j = x_j - \frac{1}{2}$, $1 \le j \le n$ does. So, if the reduced basis $\bar{B}'$, obtained by applying Lovász' basis reduction algorithm to $\bar{B}$, contains a vector $(w_1, \ldots, w_n, 0)$ with $w_j = \{-\frac{1}{2}, \frac{1}{2}\}$, $1 \le j \le n$, then the vector $(w_j + \frac{1}{2})$, $1 \le j \le n$ 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 15.

Theorem 16 *[13]. Let A be a natural number, and let $a_1, \ldots, a_n$ be random integers such that $1 \le a_j \le A$, for $1 \le j \le n$. 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 16 by showing that the probability that the lattice L'_{a,a_0} contains a vector $v = (v_1, \ldots, v_{n+1})$ satisfying

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

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 $||w||^2 \le \frac{n}{4}$. 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 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 $d(a) = n/\log_2(\max_j a_j) < 0.9408$, which implies that $\max_j a_j > 2^{n/0.9408}$, 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 16. The lattice $L''(a, a_0)$ consists of vectors

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

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

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

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

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

4 Solving diophantine equations using basis reduction

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

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

Here A is an $m \times n$-matrix, with $m \le n$, 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_d be an integral vector satisfying $Ax_d = d$, and let X_0 be an $n \times (n-m)$-matrix consisting of integer, linearly independent column vectors x_0^j, $1 \leq j \leq n-m$, such that $Ax_0^j = 0$ for $1 \leq j \leq n-m$. We can now rewrite X_{IR} as

$$X_{\mathrm{IR}} = \{x \in \mathbb{Z}^n : x = x_d + X_0\lambda,\ \lambda \in \mathbb{Z}^{n-m}\}, \tag{64}$$

that is, we express any vector x that satisfies $Ax = d$ as a vector x_d, satisfying $Ax_d = d$, 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_d that is short with respect to the bounds. Then, we may hope that x_d satisfies $l \leq x_d \leq u$, in which case we are done. If x_d does not satisfy the bounds, then we observe that $A(x_d + \lambda x_0) = d$ for any integer multiplier λ and any vector x_0 satisfying $Ax_0 = 0$. Hence, we can derive an enumeration scheme in which we branch on integer linear combinations of vectors x_0 satisfying $Ax_0 = 0$, 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:

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

where a_i is the i^{th} row of the matrix A, where N_1 and N_2 are natural numbers, and where y, as in Section 3, 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 17 *The integer vector x_d satisfies $Ax_d = d$ if and only if the vector*

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

belongs to the lattice L, and the integer vector x_0 satisfies $Ax_0 = 0$ if and only if the vector

$$(x_0^T, 0, 0^{(1\times m)})^T = B\begin{pmatrix} x_0 \\ 0 \end{pmatrix} \tag{68}$$

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:

$$\begin{pmatrix} X_0^{(n\times(n-m))} & x_d \\ 0^{(1\times(n-m))} & N_1 \\ 0^{(m\times(n-m))} & 0 \end{pmatrix}. \tag{69}$$

This result is stated in the following theorem.

Theorem 18 *[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 $1 \le j \le n-m$,*
2. *$\hat{b}_{ij} = 0$ for $n+2 \le i \le n+m+1$ and $1 \le j \le n-m+1$,*
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 18, Aardal et al. make use of inequality (15) of Proposition 5.

Once we have obtained the matrix X_0 and the vector x_d, we can derive the following equivalent formulation of problem (63):

$$\text{Does there exist a vector } \lambda \in \mathbb{Z}^{n-m} \text{ such that } l \le x_d + X_0\lambda \le u? \tag{70}$$

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 [12], and the market split instances of Cornuéjols & Dawande [11]. 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 $P = \{\lambda \in \mathbb{Z}^{n-m} : \ l \le x_d + X_0\lambda \le u\}$. At node k of the enumeration tree they choose a unit vector e_j, $1 \le j \le n-m$ 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. 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 5 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 an integral vector? To use branch-and-bound we need to introduce an objective function.

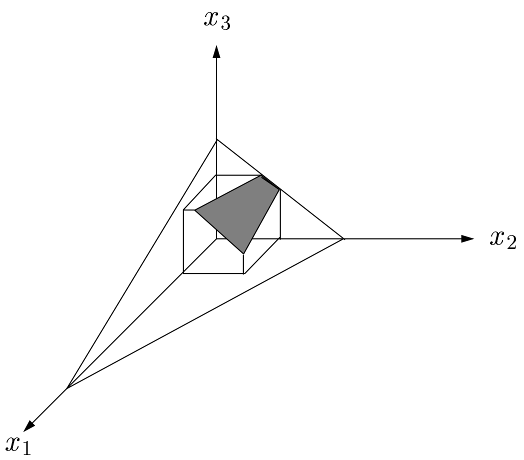

Figure 10:

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.

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

$$X_0 = \begin{pmatrix} -2 & 1 \\ 1 & 2 \\ 0 & -2 \end{pmatrix}.$$

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

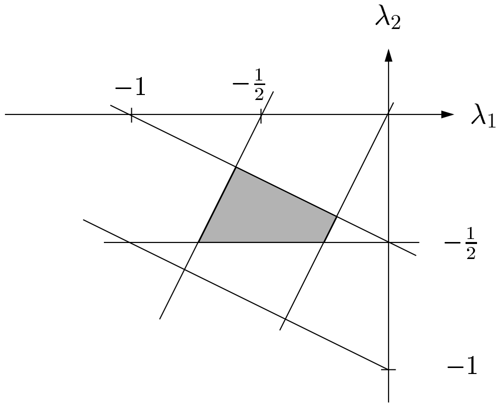

Figure 11:

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_d, i.e., the origin of the λ-coordinate system is located at the vector x_d. 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. ■

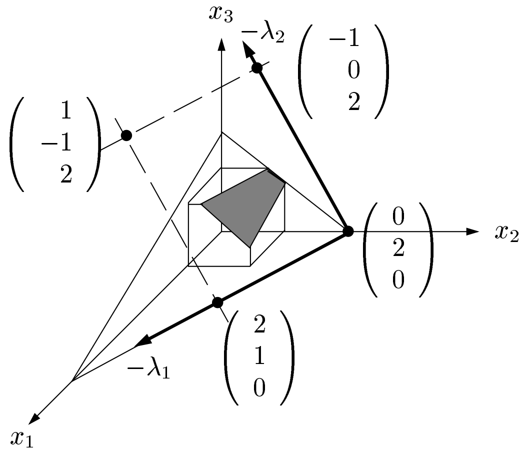

Figure 12:

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 $j = n - m, \ldots, 1$ was quite effective, and in general much better than the order $1, \ldots, n-m$. This can be explained as follows. Due to Lovász' algorithm, the vectors of X_0 are more or less in order of increasing length, so typically, the $(n-m)^{\text{th}}$ vector of X_0 is the longest one. Branching on this vector first should generate relatively few hyperplanes intersecting the linear relaxation of X, if this set has a regular shape. Note, that to branch on the j^{th} vector of X_0 corresponds to branching on the j^{th} unit vector in the space of the λ-variables.

5 Discussion

One important question is whether there exist versions of the integer programming algorithms presented in Section 2 that can be used with good results in practice. It should be noted that the main purpose of the algorithms by Lenstra [32], and by Lovász & Scarf [35], 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. [10], 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. [10], 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 sized 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.

Acknowledgements

The author 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, 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, 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, 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] K. Aardal, R. Weismantel, L. A. Wolsey (1999). Non-standard approaches to integer programming. Manuscript. To be submitted.

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

[7] J. Bourgain, 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.

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

[9] 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.

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

[11] G. Cornuéjols, 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.

[12] G. Cornuéjols, R. Urbaniak, R. Weismantel, L. 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.

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

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

[15] 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.

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

[17] 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.

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

[19] 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.]

[20] 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.

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

[22] 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.

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

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

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

[26] R. Kannan, 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.

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

[28] 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.

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

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

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

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

[33] 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`

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

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

[36] 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.

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

[38] 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.

[39] 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.

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

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

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

[43] C. P. Schnorr, 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.

[44] A. Schrijver (1986). *Theory of Linear and Integer Programming.* Wiley, Chichester.

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

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

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