Fast Reduction of Ternary Quadratic Forms

Friedrich Eisenbrand[1] and Günter Rote[2]

[1] Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, eisen@mpi-sb.mpg.de

[2] Institut für Informatik, Freie Universität Berlin, Takustraße 9, 14195 Berlin, Germany, rote@inf.fu-berlin.de

Abstract. We show that a positive definite integral ternary form can be reduced with $O(M(s)\log^2 s)$ bit operations, where s is the binary encoding length of the form and $M(s)$ is the bit-complexity of s-bit integer multiplication.

This result is achieved in two steps. First we prove that the the classical Gaussian algorithm for ternary form reduction, in the variant of Lagarias, has this worst case running time. Then we show that, given a ternary form which is reduced in the Gaussian sense, it takes only a constant number of arithmetic operations and a constant number of binary-form reductions to fully reduce the form.

Finally we describe how this algorithm can be generalized to higher dimensions. Lattice basis reduction and shortest vector computation in fixed dimension d can be done with $O(M(s)\log^{d-1} s)$ bit-operations.

1 Introduction

A *positive definite integral quadratic form* F, or *form* for short, is a homogeneous polynomial

$$F(X_1,\dots,X_d) = (X_1,\dots,X_d)\,A\,(X_1,\dots,X_d)^{\mathrm{T}},$$

where $A \in \mathbb{Z}^{d\times d}$ is an integral positive definite matrix, i.e., $A = A^{\mathrm{T}}$ and $x^{\mathrm{T}}Ax > 0$ for all $x \neq 0$. The study of forms is a fundamental topic in the geometry of numbers (see, e.g., [2]). A basic question here is: Given a form F, what is the minimal nonzero value $\lambda(F) = \min\{\, F(x_1,\dots,x_d) \mid x \in \mathbb{Z}^d,\ x \neq 0\,\}$ of the form which is attained at an integral vector? This problem will be of central interest in this paper.

Problem 1. Given a form F, compute $\lambda(F)$.

At least since Lenstra's [9] polynomial algorithm for integer programming in fixed dimension, the study of quadratic forms has also become a major topic in theoretical computer science. Here, one is interested in the lattice variant of Problem 1, which is: Given a basis of an integral lattice, find a shortest nonzero vector of the lattice w.r.t. the ℓ_2-norm.

In fixed dimension, Problem 1 can be quickly solved if F is *reduced* (see Theorem 4 in Sect. 5). In our setting, this shall mean that the product of the diagonal elements of A satisfies

$$\prod_{i=1}^{d} a_{ii} \leq \gamma_d \, \Delta_F \tag{1}$$

for some constant γ_d depending on the dimension d only. Here $\Delta_F = \det A$ is the *determinant* of the form F. Algorithms which transform a form F into an equivalent reduced form are called *reduction algorithms*.

In algorithmic number theory, the cost measure that is widely used in the analysis of algorithms is the number of required *bit operations*. The famous *LLL algorithm* [8] is a reduction algorithm which has polynomial running time, even in varying dimension. In fixed dimension, the LLL reduction algorithm reduces a form F of binary encoding size s with $O(s)$ arithmetic operations on integers of size $O(s)$. This amounts to $O(M(s)\, s)$ bit-operations, where $M(s)$ is the bit-complexity of s-bit integer multiplication. If one plugs in the current record for $M(s) = O(s \log s \log\log s)$ [11], this shows that a form F can be reduced with a close to quadratic amount of bit-operations.

A form in two variables is called a *binary form*. Here one has asymptotically fast reduction algorithms. It was shown by Schönhage [10] and independently by Yap [16] that a binary quadratic form can be reduced with $O(M(s) \log s)$ bit-operations, see also Eisenbrand [3] for an easier approach.

In his famous *disquisitiones arithmeticae* [4], Gauß provided a "reduction algorithm" for forms in three variables, called *ternary forms*. He showed how to compute a ternary form, equivalent to a given form, such that the first diagonal element of the coefficient matrix is at most $\frac{4}{3}\sqrt[3]{\Delta_F}$. A form which is reduced in the Gaussian sense is not necessarily reduced in the sense of (1). The Gaussian notion of reduction was modified by Seeber [13] such that a reduced form satisfies (1) with $\gamma_3 = 3$. Gauß [5] showed later that $\gamma_3 = 2$.

The "reduction algorithm" of Gauß was modified by Lagarias [7] to produce so called *quasi-reduced* forms. They satisfy the slightly weaker condition that the first diagonal element is at most twice the cubic root of the determinant. Lagarias proved that his modified ternary form algorithm runs in polynomial time. However, a quasi-reduced form is not necessarily reduced in the sense of (1).

Results. We prove that ternary forms can be reduced with a close to linear amount of bit-operations, as it is the case for binary forms. More precisely, a ternary form F of binary encoding length s can be reduced in the sense of (1) with $\gamma_3 = \frac{16}{3}$ using $O(M(s) \log^2 s)$ bit-operations. Unfortunately, the complexity of the proposed reduction procedure has still an extra $(\log s)$-factor compared to the complexity of binary form reduction. However our result largely improves on the $O(M(s)\, s)$ complexity of algorithms for ternary form reduction which are based on the LLL algorithm.

We proceed as follows. First we show that the Gaussian ternary form algorithm, in the variant of Lagarias [7], requires $O(M(s)\log^2 s)$ bit-operations. This is achieved via a refinement of the analysis given by Lagarias. Then we prove that, given a quasi-reduced ternary form, it takes at most $O(M(s)\log s)$ bit-operations to compute an equivalent reduced form. Therefore, a ternary form can be reduced with $O(M(s)\log^2 s)$ bit-operations. This improves on the best previously known algorithms. It follows that, for ternary forms, Problem 1 can be solved with $O(M(s)\log^2 s)$ bit-operations.

Finally we generalize the described algorithm to any fixed dimension d. The resulting lattice basis reduction algorithm requires $O(M(s)\log^{d-1} s)$ bit-operations.

Related work. Apart from the already mentioned articles, three-dimensional lattice reduction was extensively studied by various authors. Vallée [15] invented a generalization of the two-dimensional Gaussian algorithm in three dimensions. Vallée's algorithm requires $O(M(s)\,s)$ bit-operations. Semaev [14] provides an algorithm for three-dimensional lattice basis reduction which is based on pair reduction. The running time of his algorithm is $O(s^2)$ bit-operations even if one uses the naive quadratic methods for integer multiplication and division. This matches the complexity of the Euclidean algorithm for the greatest common divisor.

2 Preliminaries and Notation

The letters $\mathbb{Z}$ and $\mathbb{Q}$ denote the integers and rationals respectively. The running times of algorithms are always given in terms of the binary encoding length of the input data. The cost measure is the amount of *bit operations*. The function $M(s)$ denotes the bit-complexity of s-bit integer multiplication. All basic arithmetic operations can be done in time $O(M(s))$ [1].

We will only consider positive definite integral quadratic forms. We identify a form F with its *coefficient matrix* $M_F \in \mathbb{Z}^{d\times d}$ such that

$$F(X_1,\dots,X_d) = (X_1,\dots,X_d)\; M_F\; (X_1,\dots,X_d)^{\mathrm{T}}.$$

The function $\operatorname{size}(F)$ denotes the binary encoding length of M_F. Two forms F and G are *equivalent* if there exists a unimodular matrix $U \in \mathbb{Z}^{d\times d}$ with $M_G = U^{\mathrm{T}} M_F U$. We say that U *transforms* F into G. The number $\Delta_F = \det M_F$ is the determinant of the form. The determinant is invariant under equivalence. See, e.g., [2] for more on the theory of quadratic forms. The coefficient matrix $M_F \in \mathbb{Z}^{d\times d}$ has a unique $R^{\mathrm{T}}DR$ factorization, i.e, a factorization $M_F = R^{\mathrm{T}}DR$, where $R \in \mathbb{Q}^{d\times d}$ is an upper triangular matrix with ones on the diagonal and D is a diagonal matrix. The matrix R has a unique *normalization* $R' = RU$, where U is unimodular and R' is upper triangular with ones on the diagonal and elements above the diagonal in the range $(-\frac{1}{2},\frac{1}{2}]$. The corresponding matrix $R'^{\mathrm{T}}DR'$ defines a form F' which is equivalent to F. The form F' is called the

Gram-Schmidt normalization of F. This is the normalization step of the LLL algorithm [8], translated into the language of quadratic forms. In fixed dimension, the Gram-Schmidt normalization of a form F of size s can be computed with a constant number of arithmetic operations, and hence with $O(M(s))$ bit-operations. We say that a form G is a *γ-reduction* of F, if G is equivalent to F and if the product of the diagonal elements of M_G is at most $\gamma\,\Delta_F$.

2.1 Binary Forms

A *binary form* is a form in two variables. We denote binary forms with lower case letters f or g. The binary form f is *reduced* if $M_f = \left(\begin{smallmatrix} a_{11} & a_{12} \\ a_{12} & a_{22}\end{smallmatrix}\right)$ satisfies

$$a_{11} \le a_{22} \tag{2}$$
$$|a_{12}| \le \tfrac{1}{2} a_{11}. \tag{3}$$

If f is reduced one has

$$\tfrac{3}{4}\, a_{11}\, a_{22} \le \Delta_f. \tag{4}$$

The unimodular matrix $\left(\begin{smallmatrix} 1 & -r \\ 0 & 1\end{smallmatrix}\right)$, where r is the nearest integer to $\frac{a_{12}}{a_{11}}$, transforms a binary form f to an equivalent form which is called the *normalization* of f. The normalization of f satisfies (3).

We have the following result of Schönhage [10] and Yap [16].

Theorem 1. *Given a positive definite integral binary quadratic form f of size s, one can compute with $O(M(s)\log s)$ bit-operations an equivalent reduced form g and a unimodular matrix $U \in \mathbb{Z}^{2\times 2}$ which transforms f into g.* □

2.2 Ternary Forms

Ternary forms will be denoted by capital letters F or G. Let F be given by its coefficient matrix

$$M_F = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix}.$$

The form F defines *associated binary forms* f_{ij}, $1 \le i, j \le 3$, $i \ne j$ which have coefficient matrix

$$M_{f_{ij}} = \begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix}.$$

By *reducing f_{ij} in F*, we mean that we compute the unimodular transformation which reduces f_{ij} and apply it to the whole coefficient matrix M_F. This changes only the i-th and j-th row and column of M_F and leaves the third diagonal element a_{kk} unchanged. It follows from Theorem 1 that such a reduction of f_{ij} in F can be done with $O(M(s)\log s)$ bit-operations on forms F of size s.

The *adjoint* F^* of F is defined by the coefficient matrix $M_{F^*} = \det M_F \cdot M_F^{-1}$ and we write

$$M_{F^*} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{23} & A_{33} \end{pmatrix}.$$

Clearly M_{F^*} is integral and positive definite. A unimodular matrix $S \in \mathbb{Z}^{3\times 3}$ transforms F into G if and only if $(S^{\mathrm{T}})^{-1}$ transforms F^* into G^*. The associated binary forms of F^* are denoted by f^*_{ij} and by *reducing* such an associated form *in* F we mean that we apply the corresponding reduction operations on F. Notice that $\mathrm{size}(F^*) = O(\mathrm{size}(F))$ and $\mathrm{size}(F) = O(\mathrm{size}(F^*))$ and that $\Delta_{F^*} = \Delta_F^2$.

The ternary form F is *quasi-reduced* (see [7, p. 162]) if

$$a_{11} \le 2\sqrt[3]{\Delta_F} \tag{5}$$

$$A_{33} \le 2\sqrt[3]{\Delta_F^2} \tag{6}$$

$$|a_{12}| \le \tfrac{1}{2}\, a_{11} \tag{7}$$

$$|A_{13}| \le \tfrac{1}{2}\, A_{33} \tag{8}$$

$$|A_{23}| \le \tfrac{1}{2}\, A_{33}. \tag{9}$$

This notion is a relaxation of Gauß' concept of reduction of ternary forms, which has the constant 4/3 instead of 2 in (5–6).

3 Computing a Quasi-Reduced Ternary Form

The Gaussian algorithm [4, Arts. 272–275] for ternary form "reduction" proceeds by iteratively reducing the associated binary forms f_{12} and f^*_{32} in F. Lagarias [7] modified the algorithm by keeping the entries above and below the diagonal of the intermediate forms small so that (7–9) are fulfilled after every iteration. So we only have to see that (5) and (6) are fulfilled. One iterates until

$$A_{33} < 2\sqrt[3]{\Delta_F^2}. \tag{10}$$

In the following we prove that the number of iterations until a ternary form F of size s satisfies (10) is $O(\log s)$. For F and its adjoint F^* one has

$$\begin{aligned} A_{33} &= \Delta_{f_{12}} \\ a_{11}\Delta_F &= \Delta_{f^*_{32}}. \end{aligned} \tag{11}$$

Thus reducing f_{12} in F leaves A_{33} unchanged and reducing f^*_{32} in F leaves a_{11} unchanged. Furthermore, after reducing f_{12} in F one has

$$a_{11} \le \sqrt{\tfrac{4}{3}\, A_{33}} \tag{12}$$

by (11), (2) and (4). Similarly, after reducing f^*_{32} in F one has

$$A_{33} \le \sqrt{\tfrac{4}{3}\, a_{11}\Delta_F}. \tag{13}$$

This shows that each iteration decreases the binary encoding length of A_{33} by roughly a factor of 4 as long as A_{33} exceeds $\sqrt[3]{\Delta_F^2}$ by a large amount. We make this observation more precise.

Let $A_{kl}^{(i)}$ denote the coefficients of F^* after the i-th iteration of this procedure. By combining (12) and (13) we get the following relation (see [7, p. 166, (4.65)])

$$A_{33}^{(i+1)} \leq (\tfrac{4}{3})^{(3/4)} \sqrt{\Delta_F}\, (A_{33}^{(i)})^{1/4}. \tag{14}$$

Lagarias then remarks that, if $A_{33}^{(i)} \geq 2\,\sqrt[3]{\Delta_F^2}$, then

$$A_{33}^{(i+1)} \leq (\tfrac{2}{3})^{3/4}\, A_{33}^{(i)} \tag{15}$$

and it follows that the number of iterations is bounded by $O(s)$. Lagarias does not take full advantage of (14). By rewriting (14) in the form

$$\frac{A_{33}^{(i+1)}}{\frac{4}{3}\Delta_F^{2/3}} \leq \sqrt[4]{\frac{A_{33}^{(i)}}{\frac{4}{3}\Delta_F^{2/3}}},$$

we see that we can achieve

$$\frac{A_{33}^{(i+1)}}{\frac{4}{3}\Delta_F^{2/3}} \leq 2$$

in at most

$$i = \log_4 \log_2 \left[A_{33}^{(0)}/(\tfrac{4}{3}\Delta_F^{2/3})\right] \leq \log_4 \log_2 A_{33}^{(0)} = O(\log s)$$

iterations. After we have achieved $A_{33}^{(i)} \leq \frac{8}{3}\sqrt[3]{\Delta_F^2}$, then, by (15), the modified ternary form algorithm requires at most one additional iteration to obtain an equivalent quasi-reduced form.

This shows that the modified ternary form algorithm requires $O(\log s)$ iterations to quasi-reduce a ternary form of size s. If one iteration of the reduction algorithm is performed with the fast reduction algorithm for binary forms one obtains the following result.

Theorem 2. *The modified ternary form reduction method reduces a ternary form of size s in $O(M(s)\log^2 s)$ bit-operations.*

Proof. Lagarias proves that the sizes of the intermediate ternary forms are $O(s)$. We have seen that the number of iterations is $O(\log s)$. One iteration requires $O(M(s)\log s)$ bit-operations if one uses the fast reduction for binary forms. □

4 From Quasi-Reduced to Reduced

A quasi-reduced form (or a form which is reduced in the sense of Gauß) is not necessarily reduced. For example, the form F given by

$$M_F = \begin{pmatrix} 4x & 2x & 0 \\ 2x & x+1 & 0 \\ 0 & 0 & 2x^2 \end{pmatrix}, \quad M_{F^*} = \begin{pmatrix} 2x^3+2x^2 & -4x^2 & 0 \\ -4x^2 & 8x^3 & 0 \\ 0 & 0 & 4x \end{pmatrix}$$

with $\Delta_F = 8x^3$ is quasi-reduced, but it is far from being reduced, for $x \to \infty$.

In this section we show that we can compute a $\frac{16}{3}$-reduction of a quasi-reduced ternary form F with $O(M(s) \log s)$ bit-operations.

The following lemma states that, if F has two small entries on the diagonal which belong to an associated reduced binary form, then the Gram-Schmidt normalization of F is reduced.

Lemma 1. *Let F be a ternary form such that f_{12} is reduced and $a_{11}, a_{22} \leq \kappa \sqrt[3]{\Delta_F}$ for some κ. Then one has*

$$a'_{11}a'_{22}a'_{33} \leq \left(\tfrac{4}{3} + \tfrac{1}{2}\kappa^3\right) \Delta_F,$$

for the Gram-Schmidt normalization F' of F.

Proof. Let

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ r_{12} & 1 & 0 \\ r_{13} & r_{23} & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & r_{12} & r_{13} \\ 0 & 1 & r_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

be the $R^{\mathrm{T}}DR$ factorization of the coefficient matrix of F. Since $\Delta_{f_{12}} = d_1 d_2$, f_{12} is reduced, and $d_1 = a_{11}$, it follows that

$$d_2 \geq \tfrac{3}{4}\, a_{22}. \tag{16}$$

Now $\Delta_F = d_1 d_2 d_3$ and (16) imply

$$d_3 \leq \frac{4}{3}\, \frac{\Delta_F}{a_{11}\, a_{22}}. \tag{17}$$

Let $F' = {R'}^{\mathrm{T}} D R'$ be the Gram-Schmidt normalization of F, then

$$\begin{aligned} a'_{33} &= d_3 + (r'_{23})^2\, d_2 + (r'_{13})^2\, d_1 \leq d_3 + (r'_{23})^2\, a_{22} + (r'_{13})^2\, a_{11} \\ &\leq d_3 + \tfrac{1}{2}\kappa \sqrt[3]{\Delta_F}. \end{aligned} \tag{18}$$

Since f_{12} is reduced we have not only $a'_{11} = a_{11}$ but also $a'_{22} = a_{22}$ since $|r_{12}| \leq \frac{1}{2}$. By combining (17) and (18) and the assumption that a_{11}, $a_{22} \leq \kappa \sqrt[3]{\Delta_F}$, one obtains

$$\begin{aligned} a'_{11}\, a'_{22}\, a'_{33} &= a_{11}\, a_{22}\, a'_{33} \\ &\leq a_{11}\, a_{22}\, (d_3 + \tfrac{1}{2}\kappa \sqrt[3]{\Delta_F}) \\ &\leq \tfrac{4}{3}\Delta_F + \tfrac{1}{2}\kappa^3\, \Delta_F = \left(\tfrac{4}{3} + \tfrac{1}{2}\kappa^3\right) \Delta_F. \end{aligned}$$

Now we are ready to prove that, given a quasi-reduced ternary form F, an equivalent γ-reduction is readily available, for $\gamma = \frac{16}{3}$.

Proposition 1. *Given a quasi-reduced ternary form F of size s, one can compute with $O(M(s) \log s)$ bit-operations a $\frac{16}{3}$-reduction G of F.*

Proof. Let F be quasi-reduced and let F^* be the adjoint of F. First reduce f_{32}^* in F. This leaves a_{11} unchanged and maybe decreases A_{33}. Recall that $a_{11} \le 2\sqrt[3]{\Delta_F}$. It follows from (4) that

$$\tfrac{3}{4} A_{33}\, A_{22} \le \det f_{32}^* = a_{11} \Delta_F. \tag{19}$$

We normalize f_{12} in F. This leaves the form f_{13} unchanged. Also normalizing f_{13} in F leaves f_{12} unchanged. Therefore normalizing f_{12} and f_{13} in F leaves $A_{33} = \Delta_{f_{12}}$ and $A_{22} = \Delta_{f_{13}}$ unchanged. If, after these normalizations, f_{12} or f_{13} is not reduced, (2) must be violated and we have two diagonal elements of value at most $2\sqrt[3]{\Delta}$. By one more binary form reduction step performed on f_{12} or f_{13} in F, we are in the situation of Lemma 1 with $\kappa = 2$ after swapping the second and third row and column if necessary. It is clear that the computations in the proof of Lemma 1 can be carried out in $O(M(s))$ bit operations. In this case we compute a γ-reduction of F with $\gamma \le \frac{4}{3} + 4 = \frac{16}{3}$.

If f_{12} and f_{13} are reduced then (4) implies

$$\begin{aligned} A_{33} &= \det f_{12} \ge \tfrac{3}{4} a_{11} a_{22} \\ A_{22} &= \det f_{13} \ge \tfrac{3}{4} a_{11} a_{33} \end{aligned}$$

We conclude from (19) that

$$a_{11} \Delta_F \ge (\tfrac{3}{4})^3\, a_{11}^2\, a_{22}\, a_{33}$$

and thus that

$$\Delta_F \ge (\tfrac{3}{4})^3\, a_{11}\, a_{22}\, a_{33} \ge \tfrac{3}{16}\, a_{11}\, a_{22}\, a_{33},$$

and we have a $\frac{16}{3}$-reduction of F. The overall amount of bit operations is $O(M(s) \log s)$, where the factor $\log s$ is required for the binary reduction steps that may be necessary. □

By combining Theorem 2 and Proposition 1 we have our main result.

Theorem 3. *Given an integral positive definite ternary form F of size s, one can compute with $O(M(s) \log^2 s)$ bit-operations a $\frac{16}{3}$-reduction of F.* □

5 Finding the Minimum of a Ternary Form

The following theorem is well known.

Theorem 4. *If F is a form in d variables with coefficient matrix $M_F = (a_{ij})$ such that $\prod_{i=1}^{d} a_{ii} \le \gamma\, \Delta_F$, then*

$$\lambda(F) = \min\bigl\{\, F(x_1, \ldots, x_d) \bigm| |x_i| \le \sqrt{\gamma},\, x_i \in \mathbb{Z},\, i = 1, \ldots d \,\bigr\}. \qquad \square$$

If the dimension is fixed and F is reduced, then Theorem 4 states that $\lambda(F)$ can be quickly computed from a constant number of candidates. This gives rise to the next theorem.

Theorem 5. *The minimum $\lambda(F)$ of a positive definite integral ternary form F of binary encoding length s can be computed with $O(M(s)\log^2 s)$ bit-operations, where $M(s)$ is the bit-complexity of s-bit integer multiplication.*

Proof. Given a ternary form F of size s, we first compute a $\frac{16}{3}$-reduction G of F. Now $\lambda(F) = \lambda(G)$ and by Theorem 4, the minimum of G is attained at an integral vector $x \in \mathbb{Z}^3$ with $|x_i| \leq \sqrt{\frac{16}{3}}$, $i = 1, \ldots, 3$. By Theorem 3, all this can be done with $O(M(s)\log^2 s)$ bit-operations. □

6 Fast Reduction in any Fixed Dimension

In this section we sketch how the previous technique can be generalized to any fixed dimension. It is more convenient to describe this in the language of lattices. For this we review some terminology. A *(rational) lattice* $\Lambda \subseteq \mathbb{Q}^d$ is a set of the form $\Lambda = \Lambda(A) = \{Ax \mid x \in \mathbb{Z}^k\}$, where $A \in \mathbb{Q}^{d\times k}$ is a rational matrix of full column rank. The matrix A is a *basis* of the lattice Λ and its columns are the *basis vectors*. The lattice Λ is *integral* if $A \in \mathbb{Z}^{d\times k}$. The number k is the *dimension* of the lattice. If $k = d$, then Λ is *full-dimensional*. Let F be the quadratic form with coefficient matrix $A^{\mathrm{T}}A$. The *lattice determinant* of Λ is the number $\det\Lambda = \sqrt{\Delta_F}$ and the lattice basis $A = (x_1, \ldots, x_k)$ is *reduced* if the form F is reduced. More explicitly, this means that

$$\prod_{i=1}^{k} \|x_i\| \leq \gamma \det\Lambda \tag{20}$$

for some constant γ. The *Lattice Reduction Problem* is the problem of computing a reduced basis for a given lattice.

The *dual lattice* of a full-dimensional lattice Λ is the lattice $\Lambda^* = \{\, y \in \mathbb{Q}^d \mid y^{\mathrm{T}}x \in \mathbb{Z}, \forall x \in \Lambda \,\}$. Clearly $\Lambda^* = \Lambda(A^{\mathrm{T}^{-1}})$ and $\det\Lambda^* = 1/\det\Lambda$.

6.1 Lattice Reduction, Shortest Vectors, and Short Vectors

The *Shortest Vector Problem* is the problem of finding a shortest nonzero vector of a given lattice. This is just the translation of Problem 1 into lattice terminology. Hermite [6] proved that a d-dimensional lattice Λ always contains a (shortest) vector x with $\|x\| \leq (4/3)^{(d-1)/4}(\det\Lambda)^{1/d}$. We call the problem of computing a vector x with

$$\|x\| \leq \kappa \cdot (\det\Lambda)^{1/d},$$

where κ is an arbitrary constant, the *SHORT Vector Problem.*

Clearly, every shortest vector is also a short vector. If a reduced lattice basis is available, a shortest vector can be computed fast, as mentioned above in Sect. 5 (Theorem 4). The availability of a reduced lattice bases also implies an easy solution of the Short Vector Problem, either directly by (20) or via the Shortest Vector Problem.

So, the Short Vector Problem is apparently the easiest problem among the three problems Lattice Reduction, Shortest Vector, and Short Vector. We will show in Sect. 6.3 that Lattice Reduction (and hence the Shortest Vector Problem) can be reduced to Short Vector. In Sect. 6.2, we will first describe a solution of the Short Vector Problem which proceeds by induction on the dimension, analogously to the procedure of Sect. 3.

6.2 Finding a Short Vector

First we describe how one can find a lattice vector $x \in \Lambda$ of a d-dimensional integral lattice $\Lambda \subseteq \mathbb{Z}^d$ with $\|x\| \leq \alpha\,(4/3)^{(d-1)/4}\sqrt[d]{\det\Lambda}$, for any constant $\alpha > 1$. The procedure mimics the proof of Hermite [6] who showed that such a vector (with $\alpha = 1$) exists, see also [12, p. 79].

The idea is to compute a sequence of lattice vectors $x_0, x_1, x_2, \dots$ which satisfy the relation

$$\|x_{i+1}\| \leq (\kappa_{d-1})^{d/(d-1)}(\det\Lambda)^{(d-2)/(d-1)^2}\|x_i\|^{1/(d-1)^2}, \tag{21}$$

for a certain constant κ_{d-1}. This is the generalization of (14) to higher dimensions. We rewrite (21) as

$$\frac{\|x_{i+1}\|}{(\kappa_{d-1})^{(d-1)/(d-2)}\sqrt[d]{\det\Lambda}} \leq \left[\frac{\|x_i\|}{(\kappa_{d-1})^{(d-1)/(d-2)}\sqrt[d]{\det\Lambda}}\right]^{1/(d-1)^2}.$$

Arguing as in Sect. 3, we can obtain $\|x_i\| \leq \kappa_d \cdot (\det\Lambda)^{1/d}$ in $i = O(\log\log\|x_0\|)$ steps, if we choose the constant $\kappa_d > (\kappa_{d-1})^{(d-1)/(d-2)}$.

We now describe how the successor of x_i is computed. Let x_i be given. Consider the $(d-1)$-dimensional sub-lattice Ω^* of Λ^* defined by

$$\Omega^* = \{\, y \in \Lambda^* \mid y^{\mathrm{T}} x_i = 0 \,\}.$$

The lattice Ω^* has determinant

$$\det\Omega^* \leq \|x_i\| \, \det\Lambda^* = \|x_i\| \, (\det\Lambda)^{-1}.$$

We find a short vector $\tilde{y}$ in Ω^* with

$$\|\tilde{y}\| \leq \kappa_{d-1}(\|x_i\| \, (\det\Lambda)^{-1})^{1/(d-1)}.$$

This is a Short Vector Problem in $d-1$ dimensions, which is solved inductively. Now we repeat the same procedure, going from the dual lattice back to the original lattice: consider the $(d-1)$-dimensional sub-lattice Γ of Λ defined by

$$\Gamma = \{\, x \in \Lambda \mid \tilde{y}^{\mathrm{T}} x = 0 \,\},$$

whose determinant satisfies

$$\det \Gamma \leq \|\tilde{y}\| \cdot \det \Lambda \leq \kappa_{d-1} (\det \Lambda)^{(d-2)/(d-1)} \|x_i\|^{1/(d-1)}.$$

We find a short vector x_{i+1} of Γ with $\|x_{i+1}\| \leq \kappa_{d-1} \, (\det \Gamma)^{1/(d-1)}$, which immediately yields (21). □

As a consequence one obtains the following proposition which generalizes Theorem 2.

Proposition 2. *Let $d \in \mathbb{N}$, $d \geq 3$, and let κ_{d-1} be some constant. Suppose that, in an integral lattice Γ of dimension $d-1$ with binary encoding length s, a short vector x with*

$$\|x\| \leq \kappa_{d-1} \, (\det \Gamma)^{1/(d-1)}$$

can be found in $T_{d-1}(s)$ bit-operations. Then, for an integral lattice basis $A \in \mathbb{Z}^{d \times d}$ with binary encoding length s, we can compute a basis $B \in \mathbb{Z}^{d \times d}$ of the generated lattice Λ such that the first column vector x of B satisfies

$$\|x\| \leq \kappa_d \, (\det \Lambda)^{1/d},$$

in $T_d(s) = O(T_{d-1}(s) \log s + M(s) \log s)$ bit-operations, for any constant κ_d with $\kappa_d > (\kappa_{d-1})^{(d-1)/(d-2)}$.

Proof. We start the sequence $x_0, \ldots, x_k$ with an arbitrary vector x_0 out of the basis A. The successors are computed as described above. The computation of $\tilde{y}$ can be done with $O(T_{d-1}(s) + M(s))$ bit-operations, since this involves only one $(d-1)$-dimensional shortest vector problem and basic linear algebra. The same time bound holds for the computation of x_{i+1}. These computations have to be repeated at most $O(\log \log \|x_0\|)$ times and we arrive at a lattice vector x with $\|x\| \leq \kappa_d \, (\det \Lambda)^{1/d}$. Now we determine an integral vector $y \in \mathbb{Z}^d$ with $Ay = x$. With the extended Euclidean algorithm one can find a unimodular matrix $U \in \mathbb{Z}^{d \times d}$ with first column $y / \gcd(y_1, \ldots, y_d)$. The matrix $B = AU$ is as claimed. □

We can use this proposition inductively, starting with $\kappa_2 = \sqrt[4]{4/3}$ and $T_2(s) = O(M(s) \log s)$. We see that we can choose κ_d as close to $(4/3)^{(d-1)/4}$ as we like. So we obtain:

Corollary 1. *In a d-dimensional integral lattice $\Lambda \subseteq \mathbb{Z}^d$, a lattice vector x with $\|x\| \leq \kappa \sqrt[d]{\det \Lambda}$ can be found in $O(M(s) \log^{d-1} s)$ time, for any constant $\kappa > (\frac{4}{3})^{(d-1)/4}$.* □

6.3 Augmenting the Number of Short Vectors in the Basis

Now we generalize the approach of Sect. 4 to get a reduced basis. Suppose we have a basis $v_1, \ldots, v_d$ of the d-dimensional lattice Λ which is not reduced and such that the first $k \geq 1$ basis vectors satisfy $\|v_i\| \leq \alpha \sqrt[d]{\det \Lambda}$, $1 \leq i \leq k$ for some constant α depending on d and k only. We describe a procedure that computes a new basis $v'_1, \ldots, v'_d$ which satisfies one of the following.

(a) $v_1', \ldots, v_d'$ is reduced, or
(b) for all $1 \le j \le k+1$ one has $v_j' \le \alpha^* \sqrt[d]{\det \Lambda}$ for some constant α^* depending on d and $k+1$ only.

Let L be the subspace of $\mathbb{R}^d$ which is generated by the vectors $v_1, \ldots, v_k$ and denote its orthogonal complement by $L^\perp$. Let $\bar{v}_j$ denote the projection of v_j into $L^\perp$. Let $\Lambda^{(1)}$ be the k-dimensional lattice generated by $v_1, \ldots, v_k$ and let $\Lambda^{(2)}$ be the $(d-k)$-dimensional lattice generated by the vectors $\bar{v}_{k+1}, \ldots, \bar{v}_d$. Clearly $\det \Lambda^{(1)} \, \det \Lambda^{(2)} = \det \Lambda$. Let

$$\bar{u}_{k+1}, \ldots, \bar{u}_d$$

be a reduced basis of $\Lambda^{(2)}$ and suppose that $\bar{u}_{k+1}$ is the shortest among these basis vectors. Let $U \in \mathbb{Z}^{(d-k)\times(d-k)}$ denote the unimodular matrix which transforms $(\bar{v}_{k+1}, \ldots, \bar{v}_d)$ into $(\bar{u}_{k+1}, \ldots, \bar{u}_d)$. The vectors $v_j^* \in \Lambda$ defined by $(v_{k+1}^*, \ldots, v_d^*) = (v_{k+1}, \ldots, v_d)\, U$ are of the form

$$v_j^* = \bar{u}_j + \sum_{i=1}^{k} \mu_{ij} \, v_i,$$

with some real coefficients μ_{ij}. It follows that

$$v_j' = \bar{u}_{k+1} + \sum_{i=1}^{k} \{\mu_{ij}\} \, v_i \in \Lambda,$$

where $\{x\}$ denotes the fractional part of x. Clearly

$$v_1, \ldots, v_k, v_{k+1}', \ldots, v_d'$$

is a basis of Λ and

$$\|v_j'\| \le \|\bar{u}_j\| + k\alpha \sqrt[d]{\det \Lambda}.$$

There are two cases. If $\|\bar{u}_{k+1}\| > \sqrt[d]{\det \Lambda}$, then for all $j = k+1, \ldots, d$,

$$\|v_j'\| \le (k\alpha + 1)\, \|\bar{u}_j\|.$$

Thus we get $\|v_{k+1}'\| \cdots \|v_d'\| \le \alpha_2 \det \Lambda^{(2)}$ for some constant α_2 since $\bar{u}_{k+1}, \ldots, \bar{u}_d$ is reduced. Now let $v_1', \ldots, v_k'$ be a reduced basis of $\Lambda^{(1)}$. Then

$$\|v_1'\| \cdots \|v_d'\| \le \alpha_1 \, \det \Lambda^{(1)} \, \alpha_2 \, \det \Lambda^{(2)} = \alpha_1 \, \alpha_2 \, \det \Lambda,$$

which means that $v_1', \ldots, v_d'$ is reduced and thus (a) holds.

If, on the other hand, $\|\bar{u}_{k+1}\| \le \sqrt[d]{\det \Lambda}$, then the basis $v_1, \ldots, v_k, v_{k+1}', \ldots, v_d'$ satisfies (b). $\square$

Now it is clear how to proceed. We find the first short basis vector by Proposition 2, and we iterate the above procedure as long as case (b) prevails, increasing k. We must eventually end up with a reduced basis, because as soon as k reaches d, we have $\|v_i\| \le \alpha \sqrt[d]{\det \Lambda}$ for *all* basis vectors v_i, and this implies that the basis is reduced.

In this way, we have reduced the Lattice Reduction Problem in dimension d to one d-dimensional Short Vector Problem and a constant number (fewer than $2d$) of lower-dimensional lattice reduction problems, plus some linear algebra which can be done in $O(M(n))$ time. Thus we obtain the following theorem by induction on the dimension.

Theorem 6. *Let $d \in \mathbb{N}$, $d \geq 2$, $A \in \mathbb{Z}^{d \times d}$ be a lattice basis generating Λ and suppose that the binary encoding length of A is s. Then one can compute with $O(M(s) \log^{d-1} s)$ bit-operations a reduced basis of Λ or a shortest vector of Λ.*

□

References

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, 1974.
2. J. W. S. Cassels. *Rational quadratic forms.* Academic Press, 1978.
3. F. Eisenbrand. Short vectors of planar lattices via continued fractions. *Information Processing Letters*, 2001, to appear. `http://www.mpi-sb.mpg.de/~eisen/report_lattice.ps.gz`
4. C. F. Gauß. *Disquisitiones arithmeticae.* Gerh. Fleischer Iun., 1801.
5. C. F. Gauß. Recension der "'Untersuchungen über die Eigenschaften der positiven ternären quadratischen Formen von Ludwig August Seeber"'. Reprinted in *Journal für die reine und angewandte Mathematik*, 20:312–320, 1840.
6. Ch. Hermite. Extraits de lettres de M. Ch. Hermite à M. Jacobi sur différents objets de la théorie des nombres. *Journal für die reine und angewandte Mathematik*, 40, 1850.
7. J. C. Lagarias. Worst-case complexity bounds for algorithms in the theory of integral quadratic forms. *Journal of Algorithms*, 1:142–186, 1980.
8. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Annalen*, 261:515–534, 1982.
9. H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
10. A. Schönhage. Fast reduction and composition of binary quadratic forms. In *International Symposium on Symbolic and Algebraic Computation, ISSAC 91*, pages 128–133. ACM Press, 1991.
11. A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen (Fast multiplication of large numbers). *Computing*, 7:281–292, 1971.
12. A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley, 1986.
13. L. A. Seeber. *Untersuchung über die Eigenschaften der positiven ternären quadratischen Formen.* Loeffler, Mannheim, 1831.
14. I. Semaev. A 3-dimensional lattice reduction algorithm. In: J. H. Silverman (ed.), *CaLC 2001, Cryptography and Lattices Conference*, Lecture Notes in Computer Science, vol. 2146 (this volume), Springer-Verlag, 2001, pp. 181–193.
15. B. Vallée. An affine point of view on minima finding in integer lattices of lower dimensions. In *Proceedings of the European Conference on Computer Algebra, EUROCAL '87*, volume 378 of *Lecture Notes in Computer Science*, pp. 376–378. Springer, Berlin, 1989.
16. C. K. Yap. Fast unimodular reduction: Planar integer lattices. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 437–446, Pittsburgh, 1992. IEEE Computer Society Press.