Short Vectors of Planar Lattices Via Continued Fractions

Friedrich Eisenbrand

Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
Germany
eisen@mpi-sb.mpg.de

May 16, 2000, revised: November 5, 2001

Abstract

We show that a shortest vector of a 2-dimensional integral lattice with respect to the ℓ_∞-norm can be computed with a constant number of extended-gcd computations, one common-convergent computation and a constant number of arithmetic operations. It follows that in two dimensions, a fast basis-reduction algorithm can be solely based on Schönhage's classical algorithm on the fast computation of continued fractions and the reduction algorithm of Gauß.

Keywords: Algorithms, computational geometry, number theoretic algorithms

1 Introduction

Lattice basis-reduction is an important technique in computer science. Well known applications are integer programming in fixed dimension [10], factorization of rational polynomials [9] or the development of strongly polynomial algorithms in combinatorial optimization [3], among others.

Gauß [4] invented an algorithm that finds a "short" or reduced basis of a 2-dimensional integral lattice. Such a basis consists of two integral vectors $b_1, b_2 \in \mathbf{Z}^2$ that generate the lattice, with the additional property that the enclosed angle between b_1 and b_2 is in the range $90^\circ \pm 30^\circ$. A shortest vector of a reduced basis is then a shortest vector of the lattice. The algorithm mimics the euclidean algorithm by subtracting integral multiples of the shorter vector from the larger vector thereby reducing its length. This *normalization step* is analogous to the division with remainder in the euclidean algorithm for integers.

The integer k in the repeat-loop of algorithm GAUSS is the nearest integer to the number $(b_1^T b_2)/(b_1^T b_1)$. Lagarias [7] showed that the Gaussian algorithm is polynomial. His analysis can be used to show that GAUSS requires $O(n^2)$ bit-operations for

Algorithm. $\text{GAUSS}(b_1, b_2)$

repeat

arrange that b_1 is the shorter vector of b_1 and b_2

find $k \in \mathbf{Z}$ such that $b_2 - kb_1$ is of minimal euclidean length

$b_2 \leftarrow (b_2 - kb_1)$ (*normalization step*)

until $k = 0$

return (b_1, b_2)

inputs with n bits, even if one uses the naive quadratic algorithms for multiplication and division with remainder [8, p. 682]. Rote [12] showed that the 2-dimensional $\bmod m$ shortest vector problem can be reduced to the classical case.

We show in this paper that a shortest vector of a 2-dimensional integral lattice corresponds to a best approximation of a rational number, which is uniquely defined by the lattice. This number can be obtained from the Hermite normal form of the lattice. The best approximation of this number that represents a shortest vector w.r.t. the ℓ_∞-norm can then be found with one common convergent computation and a constant number of arithmetic operations. This implies that 2-dimensional lattice reduction can be reduced to a constant number of extended-gcd computations, one common-convergent computation and a constant number of arithmetic operations. Hence it can be carried out in time $O(M(n)\log n)$ if the classical algorithm of Schönhage [13] on the fast computation of of continued fractions is used for the extended-gcd computations and the common-convergent computation. Here $M(n)$ denotes the time needed to multiply two n-bit integers. To achieve this running time, two previous methods [14, 16] attacked this problem directly.

2 Preliminaries

The letters $\mathbf{Z}, \mathbf{Q}$, and $\mathbf{R}$ denote the integers, rationals and reals respectively. The symbol $\mathbf{N}_+$ denotes the positive natural numbers whereas $\mathbf{N}_0$ denotes the natural numbers including 0. In this paper, the running times of algorithms are always given in terms of the binary encoding length n of the input data. The function $M(n)$ denotes the time needed to multiply two integers. All *basic arithmetic operations* `+, -, *, /` can be done in time $O(M(n))$ [1]. The ℓ_∞, ℓ_1, and ℓ_2-*norm* of a vector $c = (c_1, c_2)^T \in \mathbf{R}^2$ are the numbers $\|c\|_\infty = \max\{|c_1|, |c_2|\}$, $\|c\|_1 = |c_1| + |c_2|$, and $\|c\|_2 = (c_1^2 + c_2^2)^{1/2}$, respectively. One has $\|c\|_\infty \leqslant \|c\|_2 \leqslant \sqrt{2}\,\|c\|_\infty$.

A *2-dimensional* or *planar integral lattice* Λ is a set of the form $\Lambda(A) = \{Ax \mid x \in \mathbf{Z}^2\}$, where $A \in \mathbf{Z}^{2\times 2}$ is a nonsingular integral matrix. The matrix A is called *basis* of Λ. One has $\Lambda(A) = \Lambda(B)$ for $B \in \mathbf{Z}^{2\times 2}$ if and only if $B = AU$ with some *unimodular matrix* $U \in \mathbf{Z}^{2\times 2}$, i.e., $\det(U) = \pm 1$. Denote by $a^{(i)}$, $i = 1, 2$, the i-th column of A. The basis A of Λ is called *reduced* if

$$2\,|{a^{(1)}}^T a^{(2)}| \leqslant {a^{(1)}}^T a^{(1)} \leqslant {a^{(2)}}^T a^{(2)}. \tag{1}$$

A *shortest vector* of Λ w.r.t. $\|\cdot\|$ is a nonzero member $0 \neq v$ of Λ whose norm $\|v\|$ is minimal. Here $\|\cdot\|$ stands for the ℓ_∞, ℓ_1 or ℓ_2-norm. The first column of a reduced basis of Λ is a shortest vector of Λ w.r.t. the ℓ_2-norm.

2.1 The euclidean algorithm

The *extended euclidean algorithm* takes as input a pair of integers (a,b) and computes $d = \gcd(a,b)$ and a pair of integers (x,y) with $xa + yb = d$ (see, e.g., [2, p. 71]).

Algorithm. $\textsc{exgcd}(a,b)$

$M \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$n \leftarrow 0$
while $(b \neq 0)$ **do**
    $q \leftarrow \lfloor a/b \rfloor$
    $M \leftarrow M \begin{pmatrix} q & 1 \\ 1 & 0 \end{pmatrix}$
    $(a,b) \leftarrow (b, a - qb)$
    $n \leftarrow n+1$
return $(d = a, x = (-1)^n M_{2,2}, y = (-1)^{n+1} M_{1,2})$

Let $M^{(k)}$, $k \geqslant 0$, denote the matrix M after the $k+1$-st iteration of the while-loop in EXGCD. The running time of the extended euclidean algorithm is quadratic (see, e.g., [2]).

2.2 Continued fractions

Continued fractions are a classic in mathematics, see, e.g., the books [11, 6]. A very nice and short treatment can also be found in [5, p. 134-137]. Let $a_0, \ldots, a_t$ be integers, all positive, except perhaps a_0. The *continued fraction* $\langle a_0, \ldots, a_t \rangle$ is inductively defined as a_0, if $t = 0$ and as $a_0 + 1/\langle a_1, \ldots, a_t \rangle$ if $t > 0$. The function $f_k(x) = \langle a_0, \ldots, a_{k-1}, x \rangle$, $0 \leqslant k \leqslant t$ is increasing for $x > 0$ if k is even and decreasing for $x > 0$ if k is odd. Consider the two sequences g_k and h_k that are inductively defined as

$$\begin{pmatrix} g_{-1} & g_{-2} \\ h_{-1} & h_{-2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} g_k & g_{k-1} \\ h_k & h_{k-1} \end{pmatrix} = \begin{pmatrix} g_{k-1} & g_{k-2} \\ h_{k-1} & h_{k-2} \end{pmatrix} \begin{pmatrix} a_k & 1 \\ 1 & 0 \end{pmatrix}, k \geqslant 0. \qquad (2)$$

Let $\beta_k = g_k/h_k$, then one has $\langle a_0, \ldots, a_k \rangle = \beta_k$ for $0 \leqslant k \leqslant t$. Note that h_k is increasing in k.

The *continued-fraction expansion* of a number $\alpha \in \mathbf{Q}$ is inductively defined as the sequence α if $\alpha \in \mathbf{Z}$, and as $\lfloor \alpha \rfloor, a_1, \ldots, a_t$ if $\alpha \notin \mathbf{Z}$ and where $a_1, \ldots, a_t$ is the continued fraction expansion of $1/(\alpha - \lfloor \alpha \rfloor)$. If k is even, then a_k is maximal with $\langle a_0, \ldots, a_k \rangle \leqslant \alpha$ and if k is odd, then a_k is maximal with $\alpha \leqslant \langle a_0, \ldots, a_k \rangle$. For $0 \leqslant k \leqslant t$, the number $\langle a_0, \ldots, a_k \rangle = \beta_k$ is called the *k-th convergent* of α, and we have $\beta_0 < \beta_2 < \cdots < \beta_t = \alpha < \cdots < \beta_3 < \beta_1$. It is easy to see that the continued fraction expansion of a rational number $\alpha = u/v \neq 0$ is the sequence of q's which are computed in the while-loop of the algorithm EXGCD on input (u,v). Let $R^{(k)}$ denote the matrix

$$R^{(k)} = \begin{pmatrix} a_0 & 1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} a_k & 1 \\ 1 & 0 \end{pmatrix}.$$

Then $R^{(k)} = M^{(k)}$, when EXGCD is run on (u,v) and $u/v = \alpha$.

A *fraction* is a representation $x/y, y > 0$ of a rational number, where x and y are integers. The fraction is *reduced* if $\gcd(x,y) = 1$. A fraction x/y is a *good approximation* to the number $\alpha \in \mathbf{Q}$, if one has $|\alpha - x/y| \leqslant |\alpha - x'/y'|$ for all other fractions x'/y' with $0 < y' \leqslant y$. Each convergent β_k, $0 \leqslant k \leqslant t$, of $\alpha \in \mathbf{Q}$ is a good approximation to α. A fraction x/y is a *best approximation of the second kind* to the number $\alpha \in \mathbf{Q}$, if one has $|y\alpha - x| < |y'\alpha - x'|$ for all other fractions x'/y' with $0 < y' \leqslant y$, see [6, p. 28]. A best approximation of the second kind to $\alpha \in \mathbf{Q}$ is a convergent of α.

The *common convergent* of two rational numbers $\alpha_1, \alpha_2 \in \mathbf{Q}$ is the convergent $\langle a_0, \ldots, a_k \rangle$ of α_1 and α_2 that corresponds to the longest common prefix of the continued fraction expansions of α_1 and α_2. Thus k is maximal such that the k-th convergent of α_1 and the k-th convergent of α_2 are equal. If $\alpha_1 \leqslant \alpha_2$, then this is the common convergent of all rationals in the interval $[\alpha_1, \alpha_2]$. Schönhage [13] showed how to compute the common convergent β_k and the corresponding matrix $R^{(k)}$ of two rationals $\alpha_1, \alpha_2 \in \mathbf{Q}$ in time $O(M(n)\log n)$. Schönhage's result yields an algorithm that computes in time $O(M(n)\log n)$ the greatest common divisor, $\gcd(a,b)$, of two n-bit integers a and b as well as two n-bit integers x and y that represent it, i.e., $\gcd(a,b) = xa + yb$.

3 The Hermite normal form

Before we establish the connection between best approximations and shortest vectors of planar lattices we perform some preprocessing on the lattice basis $A \in \mathbf{Z}^{2\times 2}$. Let A be of the form $A = \left(\begin{smallmatrix} a_1 & a_2 \\ a_3 & a_4 \end{smallmatrix}\right) \in \mathbf{Z}^{2\times 2}$. First we compute integers x and y that represent the greatest common divisor d of a_3 and a_4, i.e., $d = xa_3 + ya_4$. By multiplying the basis A with the unimodular matrix $\left(\begin{smallmatrix} a_4/d & x \\ -a_3/d & y \end{smallmatrix}\right)$ one obtains an upper triangular matrix

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} a_4/d & x \\ -a_3/d & y \end{pmatrix} = \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} \in \mathbf{Z}^{2\times 2}.$$

After some unimodular column operations, i.e., multiplying the first and second column with ± 1 and adding integral multiples of the first column to the second column, we can assure that $c > 0$ and $a > b \geqslant 0$ holds. This is the *Hermite normal form*, or *HNF*, of A (see, e.g., [15, p. 45]). The HNF of an integral lattice is unique and its computation requires one extended-gcd computation and a constant number of arithmetic operations. The computation of the HNF can be carried out in time $O(M(n)\log n)$ if the extended-gcd is computed with the algorithm of Schönhage [13] on the fast computation of continued fractions.

4 Best approximations and shortest vectors

Here we establish the connection between shortest vectors and best approximations. Throughout this section, assume that the norm $\|\cdot\|$ is invariant under the replacement of components by their absolute value. The ℓ_1, ℓ_2 and ℓ_∞-norms have this property.

Let Λ be given by its HNF $\left(\begin{smallmatrix} a & b \\ 0 & c \end{smallmatrix}\right) \in \mathbf{Z}^{2\times 2}$, where $c > 0$ and $a > b \geqslant 0$. Assume that $\left(\begin{smallmatrix} a \\ 0 \end{smallmatrix}\right)$ is not a shortest vector of Λ. Then, if a shortest vector has a negative second component, it yields a shortest vector with a positive second component by multiplying it with -1. Thus, if $\binom{a}{0}$ is not a shortest vector of Λ, there exists a shortest vector of the form $\binom{-xa+yb}{yc}$, where $x \in \mathbf{N}_0, y \in \mathbf{N}_+$.

Lemma 1. *If neither* $\binom{a}{0}$ *nor* $\binom{b}{c}$ *are shortest vectors of* Λ*, then there exists a shortest vector* $\binom{-xa+yb}{yc}$, $x \in \mathbf{N}_0, y \in \mathbf{N}_+$ *of* Λ *such that the fraction* x/y *is a best approximation of the second kind to the number* b/a*.*

Proof. Let $\binom{-xa+yb}{yc}$, $x \in \mathbf{N}_0, y \in \mathbf{N}_+$ be a shortest vector of Λ with minimal ℓ_1-norm among all shortest vectors and suppose that x/y is not a best approximation of the second kind of b/a. Then there exists a fraction $x'/y' \neq x/y$ with $0 < y' \leqslant y$ and $|by' - ax'| \leqslant |by - ax|$. If $y' < y$ or $|by' - ax'| < |by - ax|$, then $\binom{-xa+yb}{yc}$ does not have minimal ℓ_1-norm among the shortest vectors. So we have $y' = y$ and $|by - ax'| = |by - ax|$. Assume without loss of generality that $x < x'$ holds. The numbers x and x' have been chosen such that

$$|by - ax| = |by - ax'| = \min_{z \in \mathbf{N}_0} |by - az|$$

holds. Thus we conclude that $x' = x + 1$ and that $by - ax = a/2$.

If $y > 1$, then since $a > b \geqslant 0$, one has $|b(y-1) - ax| = |a/2 - b| \leqslant a/2$, implying that $\binom{-xa+yb}{yc}$ does not have minimal ℓ_1-norm among the shortest vectors. Thus $y = 1$ and since $a > b$ and $b - ax = a/2$ one has $x = 0$ which implies that $\binom{-xa+yb}{yc} = \binom{b}{c}$, a contradiction. ☐

Lemma 1 reveals that one can find a shortest vector with the classical extended euclidean algorithm.

A naive method would work as follows. First compute the vectors $(a,0)^T$ and $(b,c)^T$. Then compute the convergents g_k/h_k of b/a with EXGCD(b,a) and the corresponding vectors $(-g_k a + h_k b, h_k c)^T$. The shortest of the so computed vectors is a shortest vector of Λ. This algorithm would require a linear search through all convergents of b/a. In the next section we show a substantial improvement.

5 Finding a shortest vector with respect to ℓ_∞

Let Λ be given by its HNF $\begin{pmatrix} a & b \\ 0 & c \end{pmatrix} \in \mathbf{Z}^{2\times 2}$, where $c > 0$ and $a > b \geqslant 0$. In this section, we identify two candidate convergents of b/a to form a shortest vector and we apply the result of Schönhage [13] on the fast computation of continued fractions to find them. Throughout this section, we consider only shortest vectors w.r.t. the ℓ_∞-norm.

Consider the set of vectors

$$\left\{ \begin{pmatrix} -g_k a + h_k b \\ h_k c \end{pmatrix} \mid k = 0, \ldots, t \right\}, \tag{3}$$

where $\beta_k = g_k/h_k$, $0 \leqslant k \leqslant t$ are the convergents of b/a.

Proposition 2. *The shortest vector in* (3) *w.r.t.* ℓ_∞ *is represented by the last convergent of* b/a *that lies outside the interval* $[(b-c)/a, (b+c)/a]$ *or the first convergent of* b/a *that lies inside* $[(b-c)/a, (b+c)/a]$.

Proof. The absolute value of the first component of the vectors $\binom{-g_k a + h_k b}{h_k c}$, $k = 0, \ldots, t$ is decreasing, since each convergent of b/a is a good approximation of b/a. The absolute value of the second components is increasing for growing k. We have to determine the first k, for which the absolute value of the second component of $\binom{-g_k a + h_k b}{h_k c}$ is larger than the absolute value of the first component. Either this, or the previous k, is the k of the shortest vector. But $|-g_k a + h_k b| \leqslant h_k c$ if and only if $|b/a - g_k/h_k| \leqslant c/a$. □

In the next proposition we show that the common convergent of the interval $[(b-c)/a, (b+c)/a]$ is a good starting point for the convergent of b/a which is "shortest" in (3).

Proposition 3. *Let $\beta_k = g_k/h_k$ be the common convergent of $(b-c)/a$ and $(b+c)/a$. Then the k-th, $k+1$-st or the $k+2$-nd convergent of b/a represents a shortest vector in* (3) *w.r.t. the ℓ_∞-norm.*

Proof. Assume that k is even, the proof is analogous for odd k. Then $\beta_k \leqslant (b-c)/a$. If $\beta_k = (b-c)/a$, then $\binom{-g_k a + h_k b}{h_k c}$ is a shortest vector in (3) since the absolute values of the first and second components are equal. So assume that $\beta_k < (b-c)/a$.

Let $\beta_{k+1}^{(i)} = g_{k+1}^{(i)}/h_{k+1}^{(i)}$, $i = 1,2,3$ be the $k+1$-st convergent of the numbers $(b-c)/a$, b/a and $(b+c)/a$ respectively. We show now that β_k or $\beta_{k+1}^{(2)}$ is the last convergent of b/a which is not in $[(b-c)/a, (b+c)/a]$. The claim follows then from Proposition 2.

Suppose $\beta_{k+1}^{(2)}$ is not in $[(b-c)/a, (b+c)/a]$. Then one has $(b-c)/a \leqslant \beta_{k+1}^{(1)} < b/a$ and $(b+c)/a \leqslant \beta_{k+1}^{(2)} = \beta_{k+1}^{(3)}$. Let $a_1 > a_2 \in \mathbf{N}_+$ be the numbers in $\mathbf{N}_+$ with

$$h_{k+1}^{(1)} = h_{k-1} + a_1 h_k \text{ and } h_{k+1}^{(2)} = h_{k-1} + a_2 h_k.$$

Since the sequence $\beta(x) = (g_{k-1} + x g_k)/(h_{k-1} + x h_k)$, $x \in \mathbf{N}_+$ is decreasing and since a_2 is maximal with $b/a \leqslant \beta(a_2)$ and since $(b-c)/a \leqslant \beta(a_1) < b/a$ we see that $\beta(a_2 + 1) \in [(b-c)/a, b/a]$. Let $h_{k+2}^{(2)}$ be the denominator of the $k+2$-nd convergent of b/a. One has

$$h_{k+2}^{(2)} \geqslant h_k + h_{k-1} + a_2 h_k = h_{k-1} + (a_2 + 1) h_k.$$

Since each convergent of b/a is a good approximation to b/a, the $k+2$-nd convergent of b/a has to lie in $[(b-c)/a, (b+c)/a]$. □

These observations show that the classical result of Schönhage [13] on the fast computation of continued fractions can be used to compute a shortest vector of a lattice.

Corollary 4. *There exists an algorithm that computes in time $O(M(n) \log n)$ a basis B of a 2-dimensional integral lattice Λ defined by $A \in \mathbf{Z}^{2\times 2}$, with the property that the first column of B is a shortest vector of Λ w.r.t. the ℓ_∞-norm.*

Proof. First compute the HNF $\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}$ of A. Next compute the common convergent β_k of $[(b-c)/a, (b+c)/a]$ and the corresponding matrix $R^{(k)}$. The next two convergents of b/a can then be computed as follows. Perform two runs through the while-loop of

EXGCD on input $R^{(k)^{-1}}\binom{b}{a}$ and store the matrix $M^{(2)}$. The next two convergents β_{k+1} and β_{k+2} of b/a are then obtained from the matrix $R^{(k)}M^{(2)}$ according to (2). Lemma 1 and Proposition 3 show that one of the vectors represented by β_k, β_{k+1} and β_{k+2} or one of the vectors $(a,0)^T$ and $(b,c)^T$ is shortest w.r.t. ℓ_∞.

If one has a shortest vector $\binom{-xa+yb}{yc}$, then one computes two integers u and v with $\gcd(x,y) = 1 = uy - vx$. The matrix $\left(\begin{smallmatrix}-x & -u\\ y & v\end{smallmatrix}\right)$ is unimodular. Thus

$$\begin{pmatrix} a & b \\ 0 & c \end{pmatrix} \begin{pmatrix} -x & -u \\ y & v \end{pmatrix}$$

is a basis of Λ whose first column vector consists of a shortest vector of Λ w.r.t. the ℓ_∞-norm.

It is easy to see that the described method runs in time $O(M(n)\log n)$ if the algorithm of Schönhage [13] is used for the common-convergent computation and extended-gcd computations. □

6 Finding a reduced basis

In this section, $\|\cdot\|$ denotes the ℓ_2-norm. Let $B \in \mathbf{Z}^{2\times 2}$ be a basis of Λ whose first column $b^{(1)}$ is a shortest vector of Λ w.r.t. the ℓ_∞-norm. Let c be a shortest vector w.r.t. the ℓ_2-norm. It follows that $\sqrt{2}\,\|c\| \geqslant \|b^{(1)}\|$ holds, and thus that the basis B is "almost reduced".

Lagarias [7, proof of Theorem 4.2] has shown that in this case the algorithm GAUSS requires at most 3 runs through the repeat-loop to reduce B. We thus have the following consequence.

Corollary 5. *There exists an algorithm that reduces a* 2*-dimensional lattice basis* $A \in \mathbf{Z}^{2\times 2}$*, described by n-bit integers, in time* $O(M(n)\log n)$*, where* $M(n)$ *is the time required for n-bit integer multiplication.*

Acknowledgements

I am grateful to Günter Rote and Volker Priebe for many helpful remarks that greatly improved the presentation of this material. Thanks also to Alexander Bockmayr and Johannes Buchmann for interesting discussions on the subject.

References

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

[2] E. Bach and J. Shallit. *Algorithmic number theory*, volume 1: efficient algorithms. MIT Press, 1996.

[3] A. Frank and É. Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7:49–65, 1987.

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

[5] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[6] A. Ya. Khintchine. *Continued Fractions*. Noordhoff, Groningen, 1963.

[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 and H. W. Lenstra. Algorithms in number theory. In L. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 12, pages 673–715. Elsevier, 1990.

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

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

[11] O. Perron. *Die Lehre von den Kettenbrüchen*. Teubner, 3-rd edition, 1954.

[12] G. Rote. Finding a shortest vector in a two-dimensional lattice modulo m. *Theoretical Computer Science*, 172(1–2):303–308, 1997.

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

[14] 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.

[15] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley, 1986.

[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.