COMPLEXITY ANALYSIS OF A PARALLEL LATTICE BASIS REDUCTION ALGORITHM

C. HECKLER* AND L. THIELE†

Abstract. Lattice basis reduction is an important problem in geometry of numbers with applications in combinatorial optimization, computer algebra and cryptography. The well known sequential LLL-algorithm finds a short vector in $O(n^4 \log B)$ arithmetic operations on integers having binary length $O(n \log B)$, where n denotes the dimension of the lattice and B denotes the maximum L_2 norm of the initial basis vectors. In this paper a new analysis of the parallel algorithm of Roch and Villard is presented. It is shown that on an $n \times n$ mesh it needs $O(n^2 \log B)$ arithmetic operations on integers having binary length $O(n \log B)$. This improves the previous analysis and shows that an asymptotical speed up of n^2 is possible using n^2 processors.

Key words. lattice basis reduction, parallel algorithms, parallel computational complexity, size of the numbers

AMS subject classifications. 11H06, 11Y16, 68Q22

1. Introduction. Lattice basis reduction, the problem of finding a basis of a lattice with 'short' vectors, is an important problem in geometry of numbers with applications in combinatorial optimization, computer algebra and cryptography (see for example [8, 7, 6]). In the field of basis reduction a major advance was reached in 1982, when A.K. Lenstra, H.W. Lenstra Jr. and L. Lovasz [7] developed the so called LLL-algorithm, which is a polynomial time algorithm finding a 'good' basis ("reduced basis") in $O(n^4 \log B)$ arithmetic operations on integers having binary length $O(n \log B)$, where n denotes the dimension of the lattice and B denotes the maximum L_2 norm of the initial basis vectors. Subsequently various variants have been developed, see for example [10, 11]. In 1992 Roch and Villard [9, 13] developed a parallel algorithm for n^2 processors, but the efficiency of that algorithm could not be proved. Moreover, larger numbers than in the sequential LLL-algorithm can occur in this parallel algorithm. They show the bound $O(n^2 \log B)$ for the binary length of the integers. In this paper a new analysis of the parallel algorithm is presented. It is shown that on an $n \times n$ mesh it needs $O(n^2 \log B)$ arithmetic operations and communication steps on integers having binary length $O(n \log B)$. Thus, using n^2 processors an optimal speed up can be reached.

This paper is organized as follows: the next section gives a brief survey of the notations used in the paper. In section 3 the parallel algorithm is reviewed. The complexity in terms of the number of arithmetic operations is studied in section 4. Finally in section 5, the size of the numbers of the serial and parallel algorithms and the binary complexity are considered.

A more practical algorithm using fast floating point arithmetic can be found in [3]. Implementation details and practical results of different parallel algorithms are given in [2].

* Universität-GH Paderborn, Fachbereich 17, Automath, Warburgerstr. 100, D-33098 Paderborn, Germany. The research of this author was done at the Universität des Saarlandes, Lehrstuhl für Mikroelektronik, Postfach 151150, D-66041 Saarbrücken and was supported by the Deutsche Forschungsgemeinschaft, SFB 124.

† ETH Zürich, Technische Informatik und Kommunikation, ETH-Zentrum, CH-8092 Zürich, Switzerland. The research was done while this author was at the Universität des Saarlandes, Saarbrücken, Germany

2. Basic notations. We assume that the reader is familiar with the foundations of lattice basis reduction as described in section 1 of [7] (see for example also [11, 12]). Here we only describe our basic notations.

An n-dimensional lattice is a discrete subgroup of $\mathbf{R}^n$.

DEFINITION 2.1. *Let $b_1, b_2, \ldots, b_n \in \mathbf{R}^n$ be n linearly independent vectors. Then $L = L(b_1, \ldots, b_n) = \{\sum_{i=1}^{n} \lambda_i b_i | \lambda_i \in \mathbf{Z}, i = 1, \ldots, n\}$ is called a lattice with basis $(b_1, \ldots, b_n)$.*

Let us recall the Gram-Schmidt orthogonalization process. Let $\|\cdot\|$ denote the L_2 norm of a vector and let $(b_1, b_2, \ldots, b_n)$ be a basis, then the Gram-Schmidt orthogonalization $(b_1^*, b_2^*, \ldots, b_n^*)$ of mutually orthogonal vectors can be computed by $b_1^* = b_1$ and $b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$ for $i = 2, \ldots, n$, where $\mu_{i,j} = \frac{b_i^T b_j^*}{||b_j^*||^2}$. Let be $\mu_{i,i} = 1$ and $\mu_{i,j} = 0$ for $i < j$ and $M = (\mu_{i,j})_{1 \le i,j \le n}$ the matrix of the Gram-Schmidt coefficients. Then M is a lower triangular matrix where all elements of the main diagonal are equal to 1 and the above equations can be expressed as $(b_1, \ldots, b_n) = (b_1^*, \ldots, b_n^*) M^T$.

The following procedure (designated "size reduction") is used to obtain a basis of the lattice with small Gram-Schmidt coefficients $|\mu_{i,j}| \le 0.5$.

ALGORITHM 2.2.

Size reduction of the basis $(b_1, \ldots, b_n)$ **[7]**

```
for i = 2 to n do
    reduce size of column i of the basis and of M^T:
      for j = i − 1 downto 1 do
          reduce size of coefficient μ_{i,j} of column i:
            let ⌈μ_{i,j}⌋ be the nearest integer to μ_{i,j}
            and let μ_j denote the column j of M^T:
            b_i ← b_i − ⌈μ_{i,j}⌋ b_j;  μ_i ← μ_i − ⌈μ_{i,j}⌋ μ_j
      od
od
```

For technical reasons we use the following definition of a reduced basis (due to [12]) which is almost the same as the original one given in [7].

DEFINITION 2.3 (REDUCED BASIS [7, 12]). *A lattice basis $(b_1, \ldots, b_n)$ is called reduced, if both of the following conditions are fulfilled:*

- $|\mu_{i,j}| \le \frac{1}{2}$ *for all* $1 \le j < i \le n$
- $||b_{i-1}^*||^2 \le 2||b_i^*||^2$ *for all* $1 < i \le n$

The idea of the LLL lattice basis reduction algorithm of [7] consists in iteratively computing the size reduction of one basis vector b_i and swapping b_i and b_{i-1} if the second condition of the above definition is not valid. The complexity of the LLL-algorithm is given in the next theorem.

THEOREM 2.4 (SEQUENTIAL COMPLEXITY [7]). *Let $B = \max_{i=1,\ldots,n} ||b_i||^2$ be the square of the length of the longest vector of the original basis. Given a basis $(b_1, \ldots, b_n)$ of a lattice with $b_i \in \mathbf{Z}^n$ for all $1 \le i \le n$, the sequential LLL-algorithm of [7] computes a reduced basis of the same lattice in $O(n^4 \log B)$ arithmetic operations on integers having binary length $O(n \log B)$.*

3. The parallel algorithm. The main idea of the parallel algorithm of [9, 13] is to perform all possible swaps in one phase (see also [3, 4]) and to compute the size reduction of the whole matrix. Strictly speaking, b_i and b_{i+1} are swapped for *all odd* indices i with $||b_i^*||^2 > 2||b_{i+1}^*||^2$ in one phase of the algorithm and for *all even* indices i in the next phase of the algorithm (s. algorithm 3.1). The concatenation of an odd and an even phase is called an *"all swap phase"*.

ALGORITHM 3.1.

All-Swap Lattice Basis Reduction Algorithm [3, 4, 9, 13]
input: $b_1, b_2, \ldots, b_n$ a lattice basis
output: $b_1, b_2, \ldots, b_n$ a reduced basis
method:

```
Compute the Gram-Schmidt orthogonalization of
the basis, i.e. M and ||b_i*||^2, i = 1,...,n;
ordering <- odd;
while (any swap is possible) do
    Compute the size reduction of the basis (b_1,...,b_n) and
    the Gram-Schmidt coefficients M;
    if (ordering = odd) then
        swap b_i and b_{i+1} for every odd index i with
        ||b_i*||^2 > 2||b_{i+1}*||^2;
        ordering <- even;
    else
        swap b_i and b_{i+1} for every even index i with
        ||b_i*||^2 > 2||b_{i+1}*||^2;
        ordering <- odd;
    fi;
    Repair the Gram-Schmidt orthogonalization
od.
```

In the serial case, $O(n^3)$ arithmetic operations are required for one phase. However, the two major steps of the algorithm, that is the orthogonalization of the basis (respectively the repair of the orthogonalization) and the size reduction of the whole matrix, are well suited for parallelization. Thus, the following theorem holds.

THEOREM 3.2 ([3, 4, 9, 13]). *An all swap phase can be done in $O(n)$ parallel steps (according to the number of arithmetic operations and the number of communications of numbers) on a mesh connected network of n^2 processors.*

4. Parallel complexity. In order to achieve a speedup of n^2 using n^2 processors in comparison with the LLL algorithm it must be shown that a short vector can be found after at most $n \log B$ all-swap phases. The following analysis is taken from [3].

THEOREM 4.1 (PARALLEL TIME COMPLEXITY). *After at most $n \log B$ all-swap phases*[1] *the following situation holds for all i with $1 \le i \le n$*

$$||b_1^*||^2 \cdot ||b_2^*||^2 \cdot \ldots \cdot ||b_i^*||^2 \le \tfrac{4}{3} \cdot c^{\frac{i(n-i)}{2}} \cdot (\det L)^{\frac{2i}{n}}$$

with $c = \frac{32}{9}$.

Proof. Consider the ratios

$$(1)\quad \frac{||b_1^*||^2}{c^{\frac{n-1}{2}}(\det L)^{\frac{2}{n}}} \cdot \frac{||b_2^*||^2}{c^{\frac{n-3}{2}}(\det L)^{\frac{2}{n}}} \cdot \ldots \cdot \frac{||b_i^*||^2}{c^{\frac{n-2i+1}{2}}(\det L)^{\frac{2}{n}}} = \frac{\prod_{j=1}^{i} ||b_j^*||^2}{c^{\frac{i(n-i)}{2}}(\det L)^{\frac{2i}{n}}} =: v(i)$$

We are finished when all these ratios $v(i), 1 \le i \le n$, are less than $\frac{4}{3}$. Let max be the maximum of all these ratios, that is $max = \max\{v(i) | 1 \le i \le n\}$ and max' the maximum after one further all-swap phase.

[1] In the following log denotes the logarithm with basis $\frac{4}{3}$.

It will be shown that every all-swap phase decreases the value of max by a factor less than $\frac{3}{4}$, that is $max' \leq \frac{3}{4}max$, if $max > \frac{4}{3}$. In the beginning, $max \leq B^n$ since $||b_i^*||^2 \leq B$ holds for all $1 \leq i \leq n$. Therefore, all ratios are less than $\frac{4}{3}$ after at most $n \log B$ all-swap phases.

PROPOSITION 4.2. *If $max > \frac{4}{3}$ and $v(i) > \frac{3}{4}max$, the product $||b_1^*||^2 \cdot ||b_2^*||^2 \cdot \ldots \cdot ||b_i^*||^2$ and thereby $v(i)$ is decreased by a factor less than $\frac{3}{4}$ in one all-swap phase. All other ratios remain unchanged or decrease.*

Proof of the proposition:

If two vectors b_j and b_{j+1} are swapped, the product $||b_1^*||^2 \cdot ||b_2^*||^2 \cdot \ldots \cdot ||b_j^*||^2$ is decreased by a factor less than $\frac{3}{4}$; all other products remain unchanged, see [7] for this facts. Thus, we have to show that $max \leq \frac{4}{3}$ or b_i and b_{i+1} are swapped for all i with $v(i) > \frac{3}{4}max$ in one all-swap phase, that is if $v(i) > \frac{3}{4}max$, then $||b_i^*||^2 > 2||b_{i+1}^*||^2$. If this swap condition holds for an even index i, it is also valid after one odd swapping phase.

If $v(n) > \frac{3}{4}max$, then $max < \frac{4}{3}$ since $v(n) = 1$.

Therefore, consider an index $i, 1 \leq i \leq n-1$, with $v(i) > \frac{3}{4}max$. We show in the following that the vectors b_i and b_{i+1} are swapped in this case.

If $v(i) > \frac{3}{4}max$, then $v(i) > \frac{3}{4}v(i+1)$. Using (1) we obtain:

$$\frac{\frac{4}{3}(\det L)^{\frac{2}{n}} c^{\frac{n-2i-1}{2}}}{||b_i^*||^2} > \frac{||b_{i+1}^*||^2}{||b_i^*||^2} \tag{2}$$

In the same way, if $v(i) > \frac{3}{4}max$ and $i > 1$, then $v(i) > \frac{3}{4}v(i-1)$ holds and therefore

$$||b_i^*||^2 > \frac{3}{4}c^{\frac{n-2i+1}{2}}(\det L)^{\frac{2}{n}} \tag{3}$$

This relation is also valid for $i = 1$, if $max \geq 1$ and $v(1) > \frac{3}{4}max$.

However, it follows from (2), (3) and the choice of c that

$$\frac{\left(\frac{4}{3}\right)^2}{c} = \frac{1}{2} > \frac{||b_{i+1}^*||^2}{||b_i^*||^2} \tag{4}$$

and therefore the swap condition is satisfied.

Thus, the proposition and the theorem have been proven. □

In particular, we have found a short vector as shown in the following corollary:

COROLLARY 4.3. *After at most $n \log B$ all-swap phases the vector b_1 of the resulting lattice basis satisfies*

$$||b_1|| \leq \tfrac{4}{3}c^{\frac{n-1}{4}}(\det L)^{\frac{1}{n}}$$

Proof. Use theorem 4.1 with $i = 1$ and $b_1 = b_1^*$. □

COROLLARY 4.4. *A short vector of a lattice can be found in $O(n^2 \log B)$ parallel steps using a network of n^2 processors. Thus, an asymptotical*[2] *optimal speed-up of n^2 can be achieved in comparison with the serial LLL algorithm.*

Proof. See corollary 4.3 and theorem 3.2 and corollary 2.4. □

[2] i.e. for $n \to \infty$

5. Size of the numbers and binary complexity. The time for one operation depends on the size of the numbers. If we start with an integral basis, that is $b_i \in \mathbf{Z}^n$ for all $1 \leq i \leq n$, all numbers occuring throughout the algorithm are rational with denominators less than B^n [7]. See [7, 5] for an analysis of the sizes of the numbers of the LLL-algorithm. The bounds are summarized in table 1. The known analysis of the parallel algorithm is reviewed in the next section. Then a new result for the parallel case is given.

5.1. Known analysis of the parallel case. In the parallel case the Gram-Schmidt orthogonalization is computed for the whole matrix whereby bigger numbers than in the serial case can occur where the orthogonalization is only computed for the reduced part of the basis.

THEOREM 5.1 ([7]). *After the computation (or repair) of the Gram-Schmidt orthogonalization we have* $\mu_{i,j}^2 \leq iB^n$.

In the parallel algorithm the size reduction is computed simultaneously for all columns. Therefore, the size reduction of each column is performed using non-reduced columns of the basis. This leads to the following theorem.

THEOREM 5.2. *Let* $\mu_{i,j}^2 \leq M$ *for all* $1 \leq j < i \leq n$ *before the size reduction. Then the following inequalities are valid during the size reduction phase of the parallel algorithm (see also [4, 13]).*

$$\mu_{i,j}^2 \leq 4^n M^n \quad \textit{for} \quad 1 \leq j < i \leq n \tag{5}$$

$$\|b_i\|^2 \leq nB4^n M^n \quad \textit{for} \quad 1 \leq i \leq n \tag{6}$$

Proof. Consider the size reduction of column k. During the reduction of the element $\mu_{k,k-1}$ the computation $\mu_{k,i} \leftarrow \mu_{k,i} - \lceil \mu_{k,k-1} \rfloor \mu_{k-1,i}$ for $i = 1, \ldots, k-1$ is performed. In the case of a parallel execution, column $k-1$ is not size reduced. Therefore, only the estimation $|\mu_{k-1,i}| \leq \sqrt{M}$ is possible. Thus, the new value of $|\mu_{k,i}|$ satisfies $|\mu_{k,i}| \leq 2M$. The proposition follows by induction and with $\|b_i\|^2 = \sum_{j=1}^{i} \mu_{i,j}^2 \|b_j^*\|^2$. □

Using theorem 5.1 and 5.2 the following result about the size of the numbers of the parallel lattice basis reduction algorithm can be derived.

COROLLARY 5.3. *During the parallel lattice basis reduction the following inequalities hold for all values of* $\|b_i\|^2$ *and* $\mu_{i,j}^2$ *(see also [13]).*

$$\mu_{i,j}^2 \leq (4nB^n)^n \quad \textit{for} \quad 1 \leq j < i \leq n \tag{7}$$

$$\|b_i\|^2 \leq nB(4nB^n)^n \quad \textit{for} \quad 1 \leq i \leq n \tag{8}$$

5.2. New results. Two examples for theorems 5.1 and 5.2 are presented where large numbers do in fact occur. However, both situations never occur simultaneously. Therefore, a better analysis of the size of the numbers can be derived than the one given in corollary 5.3. This new result is given in theorem 5.7.

THEOREM 5.4. *The bound given in theorem 5.1 is asymptotically tight.*

Proof. Consider the following $n \times n$ basis for some $D > 1 \in \mathbf{Z}$.

$$\begin{pmatrix} D & D+1 & \cdots & \cdots & D+1 & 0 \\ 0 & D & D+1 & \cdots & D+1 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & D & D+1 & 0 \\ 1 & \cdots & 1 & 1 & 1 & 1 \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{pmatrix}$$

For the Gram-Schmidt orthogonalization of that basis we have

$$\mu_{n,n-1} = (-1)^{n-2} D^{n-2}$$

The proof can be carried out by induction over n using 'Givens' rotations for the computation of the Gram-Schmidt orthogonalization (see for example [1] for Givens rotations and [3] for the computation of the Gram-Schmidt orthogonalization via Givens rotations). □

THEOREM 5.5. *The bound for $\mu_{i,j}^2$ given in theorem 5.2 is asymptotically tight.*

Proof. Consider the following $n \times n$ basis for some $D > 1 \in \mathbf{Z}$.

$$(b_1, \ldots, b_n) = \begin{pmatrix} 1 & D & \cdots & D \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & D \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

For the matrix of the Gram-Schmidt coefficients of the basis above we have that $M^T = (b_1, \ldots, b_n)$. By induction it can easily be shown that coefficients of the size D^{n-1} occur during the size reduction of column n. □

In order to prove that both cases given above cannot occur simultaneously we use the following lemma.

LEMMA 5.6. *Consider the size reduction of column k, $2 \leq k \leq n$, that is of the elements $\mu_{k,k-1}, \ldots, \mu_{k,1}$. After the size reduction of the element $\mu_{k,k-i}$ with $1 \leq i \leq k-2$, $2^i - 1$ products of Gram-Schmidt coefficients were subtracted from each coefficient $\mu_{k,k-j}$, $j > i$. Each product only consists of (rounded) coefficients of* different *rows of the matrix M^{T} and only of coefficients of the rows $k-1, k-2, \ldots, k-i$ and $k-j$.*

Proof. The proof is done by induction over i.

Let $\mu_{x,y}^{(0)}$ denote $\mu_{x,y}$ before the size reduction of column k and let $\mu_{x,y}^{(i)}$ denote $\mu_{x,y}$ after the size reduction of $\mu_{k,k-i}$.

$i = 1$: After the size reduction of $\mu_{k,k-1}$ the following is valid for $j > 1$:

$$\mu_{k,k-j}^{(1)} = \mu_{k,k-j}^{(0)} - \lceil \mu_{k,k-1}^{(0)} \rfloor \mu_{k-1,k-j}^{(0)}$$

Thus, the proposition is true for $i = 1$.

After the size reduction of $\mu_{k,k-i-1}$ the following holds for $j > 1$:

$$\mu_{k,k-j}^{(i+1)} = \mu_{k,k-j}^{(i)} - \lceil \mu_{k,k-i-1}^{(i)} \rfloor \mu_{k-i-1,k-j}^{(0)}$$

Now, the proposition for $i+1$ follows with the induction hypothesis for $\mu_{k,k-j}^{(i)}$ and $\mu_{k,k-i-1}^{(i)}$. □

Now, the following theorem about the size of the numbers of the parallel lattice basis reduction can be shown.

THEOREM 5.7. *During the parallel size reduction phase the following inequalities hold.*

$$\mu_{i,j}^2 \leq n^n(2B)^{2n} \qquad \text{for} \quad 1 \leq j < i \leq n \tag{9}$$

$$\|b_i\|^2 \leq nBn^n(2B)^{2n} \qquad \text{for} \quad 1 \leq i \leq n \tag{10}$$

Proof. During the size reduction only sums with at most 2^n terms occur. Each term is a product of coefficients of *different* rows. With $\mu_{i,j}^2 \leq \frac{\|b_i\|^2}{\|b_j^*\|^2}$ [7] we have

$$\prod \mu_{i,j}^2 \leq \prod \frac{\|b_i\|^2}{\|b_j^*\|^2}$$

With $\|b_i\|^2 \leq nB$ [7] it follows

$$\prod \|b_i\|^2 \leq (nB)^n$$

As $\|b_1^*\|^2 \cdot \ldots \cdot \|b_n^*\|^2 \geq 1$ and $\|b_i^*\|^2 \leq B$ for all $1 \leq i \leq n$ (see [7]) and because in each product only different b_j^*'s occur we obtain

$$\prod \|b_j^*\|^2 \geq \frac{1}{B^n}$$

and therefore for each product the following inequalities are valid

$$\begin{aligned} \prod \mu_{i,j}^2 &\leq n^n B^{2n} \\ |\prod \mu_{i,j}| &\leq n^{n/2} B^n \end{aligned}$$

Each sum consists of at most 2^n products. Therefore we obtain for all values

$$\begin{aligned} |\mu| &\leq n^{n/2}(2B)^n \\ \mu^2 &\leq n^n(2B)^{2n} \end{aligned}$$

Again, the proposition for the values of b_i follows with $\|b_i\|^2 = \sum_{j=1}^{i} \mu_{i,j}^2 \|b_j^*\|^2$. □

Table 1 summarizes the results with respect to the sizes of the numbers.

TABLE 1
Comparison of the sizes of the numbers

	LLL [7, 5]	parallel (old analysis)	parallel (new analysis)
$\mu_{i,j}^2$ after Gram-Schmidt	$n2^nB$	nB^n	nB^n
$\|b_i\|^2$ after Gram-Schmidt	nB	nB	nB
$\mu_{i,j}^2$ during size reduction	$n2^{3n}B$	$(4nB^n)^n$	$n^n(2B)^{2n}$
$\|b_i\|^2$ during size reduction	$n^2 2^{3n} B^2$	$nB(4nB^n)^n$	$nBn^n(2B)^{2n}$

5.3. Further remarks. There is a special class of lattices (which are constructed for the solution of the subset sum problem) for which a better analysis of the size of the numbers in the parallel case is possible [13].

Joux [4] has presented a method for avoiding big numbers during the parallel size reduction. But also with that method a better analysis than the one given in theorem 5.7 is not possible.

In practical experiments for the parallel algorithm of [13] and [2, 3] larger numbers than in the serial algorithm do not occur.

5.4. Binary complexity. The estimation of the sizes of the numbers and a detailed analysis by Kaltofen [5] lead to the following binary complexity of the LLL algorithm using standard arithmetic.

THEOREM 5.8. *[5] The LLL algorithm has binary complexity* $O(n^5(\log B)^3 + n^6(\log B)^2)$.

The binary parallel complexity is given by the following theorem.

THEOREM 5.9. *The binary complexity of parallel lattice basis reduction using* n^2 *processors is* $O(n^4(\log B)^3)$.

Proof. The size of all numbers (numerators and denominators) is bounded by $O(n \log B)$. Thus the time for one arithmetic operation is at most $O(n^2(\log B)^2)$ using standard arithmetic. At most $O(n^2 \log B)$ arithmetic operations must be performed. $\square$

Acknowledgments. The authors would like to thank the two unnamed referees for their helpfull remarks and suggestions.

REFERENCES

[1] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, 1983.
[2] C. HECKLER, *Automatische Parallelisierung und parallele Gitterbasisreduktion.* Dissertation, Institut für Mikroelektronik, Universität des Saarlandes, Saarbrücken, Germany, March 1995.
[3] C. HECKLER AND L. THIELE, *A parallel lattice basis reduction for mesh-connected processor arrays and parallel complexity*, in Proc. SPDP'93 (Dallas, December 1 - 4), 1993, pp. 400–407.
[4] A. JOUX, *A fast parallel lattice reduction algorithm*, in Proc. Second Gauss Symposium (Munich), 1993.
[5] E. KALTOFEN, *On the complexity of finding short vectors in integer lattices*, in Proc. EUROCAL83 (London), LNCS 162, Springer-Verlag, 1983, pp. 236–244.
[6] R. KANNAN, *Algorithmic geometry of numbers*, Ann. Rev. Comput. Sci., 16 (1987), pp. 231–267.
[7] A. K. LENSTRA, H. W. LENSTRA JR., AND L. LOVASZ, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.
[8] H. W. LENSTRA JR., *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.
[9] J. ROCH AND G. VILLARD, *Parallel gcd and lattice basis reduction*, in Proc. CONPAR92 (Lyon), LNCS 632, Springer-Verlag, 1992, pp. 557–564.
[10] C. SCHNORR, *A more efficient algorithm for lattice basis reduction*, J. Algorithms, 9 (1988), pp. 47–62.
[11] C. SCHNORR AND M. EUCHNER, *Lattice basis reduction: Improved practical algorithms and solving subset sum problems*, in Proc. FCT'91 (Gosen, Germany), LNCS 529, Springer, 1991, pp. 68–85.
[12] A. SCHRIJVER, *Theory of Linear and Integer Programming*, John Wiley and Sons, 1986.
[13] G. VILLARD, *Parallel lattice basis reduction*, in Proc. International Symposium on Symbolic and Algebraic Computation, Berkeley California USA, ACM Press, 1992, pp. 269–277.