Electronic Colloquium on Computational Complexity, Report No. 52 (1998)

ON THE DECOMPOSITION OF LATTICES

BORIS HEMKEMEIER AND FRANK VALLENTIN

ABSTRACT. A lattice in euclidean space which is an orthogonal sum of nontrivial sublattices is called decomposable. We present an algorithm to construct a lattice's decomposition into indecomposable sublattices. Similar methods are used to prove a covering theorem for generating systems of lattices and to speed up variations of the LLL algorithm for the computation of lattice bases from large generating systems.

1. INTRODUCTION

Let L be a lattice on euclidean n-dimensional space $(V, (-,-))$, i.e. $L = \mathbf{Z}b_1 + \ldots + \mathbf{Z}b_n$ for a basis $\{b_1, \cdots, b_n\}$ of V. L is called integral if $(b_i, b_j) \in \mathbf{Z}$ holds for all $1 \leqslant i, j \leqslant n$.

Definition 1.1. A nontrivial lattice L on V is called *decomposable* if there exist (proper) sublattices $L_1, L_2 \subset L$ such that $L = L_1 \oplus L_2$ and $(L_1, L_2) = 0$, otherwise *indecomposable*.

Throughout this paper we denote an inner direct, orthogonal sum with $\oplus$. The norm of a shortest nonzero lattice vector in L is called $\min L$.

For each nontrivial lattice L there exists a decomposition $L = L_1 \oplus \ldots \oplus L_r$ into indecomposable sublattices L_i. This decomposition is unique up to the order of summands. The first proof of this fact is due to Eichler [Eic52], later M. Kneser gave a constructive, much simpler proof [Kne54]. He used an Erathostenes' sieve construction to find generating systems for these sublattices L_i.

In section 2 we summarize the basic steps to illustrate the geometric concepts of his idea. In section 3 we present the improved Algorithm 1 to solve the decomposition problem. The original method has a quadratic running time with respect to the size of a complete generating system, ours is linear.

A very common problem is the construction of a lattice basis from a generating system. There are some well known methods to solve this task like the computation of the Hermite Normal Form, the Buchmann-Pohst-LLL [BP89] or the MLLL [Poh87], Pohst's modification of the basis reduction algorithm of Lenstra, Lenstra, and Lovász [LLL82]. But also the fast MLLL will slow down if the generating system is large (e.g. see Figure 1, p. 8). We observed that even in large generating system there is often only a small number of vectors needed to span the lattice. We use the methods from Section 3 to prove Theorem 3.5: In every (finite) generating system of a lattice with vectors not longer than B we will find at most $n + \frac{1}{2}\log_2(n!(\frac{B}{M})^n) \in O(n \log \frac{nB}{M})$ vectors which generate the lattice. It is possible to identify such a vector by inspection of the result of a single matrix vector multiplication. Thus we get a significantly speed up of algorithms which construct a lattice basis if we restrict their input to these vectors. More exactly: let an integral n-dimensional lattice L be generated by s vectors of norm not larger than B. Then Algorithm 2 constructs a lattice

Date: August 5, 1998.

ISSN 1433-8092

basis of L in at most $O(n\log(nB)\cdot(n^5+n^4\log B)+sn^2)\subset O(n^6\log^2(nB)+sn^2)$ arithmetic operations (Theorem 4.1). This is better than the worst-case bound of at most $O((n+s)^4\log B$ arithmetic operations for the Buchmann-Pohst-LLL cited above. Also simple iteration methods which need at most $O(s\cdot(n^5+n^4\log B))$ arithmetic operations are not superior.

2. Kneser's Method

Definition 2.1. A vector $v\in L\setminus 0$ is called *decomposable* if there are $x,y\in L$ with $||v||>||x||\geqslant||y||$ and $v=x+y$, otherwise *indecomposable*.

Let $B\in\mathbf{R}$ be such that $S:=\{v_1,\cdots,v_s\}=\{v\in L\setminus 0\mid ||v||\leqslant B\}$ is a generating system of L. We call such a set a *complete generating system*. Let I be the set of indecomposable vectors in S. We define the orthogonality graph $\Gamma=(I,E)$, $E=\{\,\{v,w\}\mid v,w\in I$ and $(v,w)\neq 0\}$. Let $\Gamma_i=(I_i,E_i)$, $1\leqslant i\leqslant r$ be all connected components of Γ and $L_i=\langle I_i\rangle_{\mathbf{Z}}$.

Theorem 2.2 ([Kne54]). *All L_i are indecomposable, $L=L_1\oplus\ldots\oplus L_r$, and this decomposition is unique up to the order of summands.*

Proof. Assume $v\in S\setminus\langle I\rangle_{\mathbf{Z}}$ and $||v||$ be minimal with this property. There are $x,y\in S$ with $||v||>||x||\geqslant||y||$ and $v=x+y$. At least one of them is not in $\langle I\rangle_{\mathbf{Z}}$ and we have a contradiction to the minimality of $||v||$. This proves $\langle I\rangle_{\mathbf{Z}}=\langle S\rangle_{\mathbf{Z}}$ and thus $L=\sum_{i=1}^r L_i$.

$(I_i,I_j)=0$ for all $i\neq j$ implies $(L_i,L_j)=0$. Then it follows from the definiteness of $(-,-)$ that $L=\bigoplus_{i=1}^r L_i$.

For a fixed i let $L',L''\subseteq L_i, L'\neq 0$ be sublattices such that $L_i=L'\oplus L''$. For each $v\in I_i$ it holds either $v\in L'$ or $v\in L''$. With $(L'\cap I_i,L''\cap I_i)=0$ we have $L'\cap I_i=I_i$ because Γ_i is connected. This shows $L'=L_i$ and that L_i is indecomposable. From $(v,I_j)=0$ for all $j\neq i$ follows that L_i is the unique indecomposable sublattice of L containing v, thus the given decomposition is unique up to the order of summands. □

We give a sketch of Kneser's simple algorithm to construct $\Gamma=(I,E)$:

Construct I: For each pair $v,w\in S$ do: If $v+w\in S$ and $||v+w||>\max\{||v||,||w||\}$ then mark $v+w$ as decomposable. Let I be the set of all unmarked vectors.

Construct E: For each pair $v,w\in I$ do: If $(v,w)\neq 0$ then add $\{v,w\}$ to E.

To mark all the decomposable vectors in S is an expensive task because we have to inspect $\binom{s}{2}$ pairs of vectors. This method is infeasable for large s.

Remark 2.3. In the original definition [Kne54] a vector is called indecomposable if it is not a sum of nonzero orthogonal lattice vectors. This notion is completely analogous to Definition 1.1 but slightly weaker than ours. In generic lattices all nonzero lattice vectors are indecomposable in the latter meaning, for example $L=\langle(1,0),(\theta,1)\rangle_{\mathbf{Z}}$ with $\theta^2\notin\mathbf{Q}$. In particular every nonzero vector of a one-dimensional lattice is indecomposable. Our next lemma shows that with respect to Definition 2.1 there is only a finite number of indecomposable vectors.

Lemma 2.4. *Let L be a lattice on the n-dimensional euclidean space V with covering radius R. Then a vector $v\in L$ is decomposable if $||v||>2R$. In particular the number of indecomposable vectors of L is finite.*

Proof. Let R be the covering radius of L, i.e. $R=\max_{x\in V}\min_{v\in L}||x-v||$. Let $v\in L$ be a lattice vector with $||v||>2R$. There exists a $w\in L$ with $||w-\frac{1}{2}v||\leqslant R$. Then holds $||w||\leqslant||w-\frac{1}{2}v||+||\frac{1}{2}v||\leqslant R+\frac{1}{2}||v||<||v||$ and $||w-v||\leqslant||w-\frac{1}{2}v||+||-\frac{1}{2}v||\leqslant$

$R+\frac{1}{2}||v|| < ||v||$. Thus v is decomposable because it is the sum of the strict shorter lattice vectors w and $v-w$. The proof will be completed by the fact that there are only finitely many vectors of norm smaller than $2R$. □

3. An Improved Algorithm

The main observation is that it is not necessary to determine all decomposable vectors. It suffices to delete all decomposable vectors from S which are a sum of indecomposable vectors from more than one L_i. We show that we can determine such a vector with a single matrix vector multiplication. Additionally we need to update our stepwise generated L_i's but we will show that these updates are cheap and rarely needed with respect to s.

Theorem 3.1. *Let L be a n-dimensional lattice and let $S := \{v_1, \cdots, v_s\} = \{v \in L \setminus 0 \mid ||v|| \leqslant B\}$ for a $B \in \mathbf{R}$ such that S is a generating system of L. Then Algorithm 1 computes the decomposition of L into indecomposable sublattices $L_i \subseteq L, 1 \leqslant i \leqslant r$.*

Algorithm 1 Decomposition of a lattice L

```
*Input:* $B \in \mathbf{R}$ and a generating system $S = \{v \in L \setminus 0 \mid ||v|| \leqslant B\}$ of $L$.
*Output:* Indecomposable sublattices $L_i$ s.t. $L = L_1 \oplus \ldots \oplus L_r$.
// For a $L_i$ let $\pi_i : V \to \langle L_i \rangle_{\mathbf{Q}}$ be the orthogonal projection on the vectorspace spanned by it.
// 1. Initialize.
Choose $v \in S$ with minimal $||v||$.
$S \leftarrow S \backslash \{v\}, k \leftarrow 1, L_k \leftarrow \langle v \rangle_{\mathbf{Z}}$
// 2. Join vectors of $S$ into indecomposable sublattices.
while $S \neq \emptyset$ do
  Choose $v \in S$ with minimal $||v||$.
  $S \leftarrow S \backslash \{v\}$.
  if $v \notin L_1 \oplus \ldots \oplus L_k$ then
    // $v$ is indecomposable.
    $J \leftarrow \{i \in \{1, \cdots, k\} \mid \pi_i(v) \neq 0\}$.
    $\tilde{L} \leftarrow \mathbf{Z}v + \bigoplus_{i \in J} L_i$
    // Reorder list of lattices $L_i$
    $\{L_1, \cdots, L_{k-|J|}\} \leftarrow \{L_i \mid i \notin J\}, \quad L_{k-|J|+1} \leftarrow \tilde{L}, \quad k \leftarrow k - |J| + 1$
  end if
end while
```

Proof. We will show that the sublattices L_i, $i = 1, \cdots, k$, are indecomposable and pairwise orthogonal. Further we have $L = \langle S \rangle_{\mathbf{Z}} + \sum_{i=1}^{k} L_i$ in all loops, in particular is $L = \sum_{i=1}^{k} L_i$ after all v are processed. We prove now the loop invariance of these properties.

Assume that indecomposable, orthogonal sublattices $L_1, \cdots, L_k$ are already constructed and let $v \in S$ be a vector with smallest norm. If $v \in \bigoplus_{i=1}^{k} L_i$ we skip v, otherwise we add it. And so is $L = \langle S \rangle_{\mathbf{Z}} + \sum_{i=1}^{k} L_i$ at the end of each loop.

Let now $v \notin \bigoplus_{i=1}^{k} L_i$. Then v is indecomposable. In particular the orthogonal sum $v = (v - \pi_i(v)) + \pi_i(v), 1 \leqslant i \leqslant k$ forces that either $\pi_i(v) = 0$ or $\pi_i(v) \notin L_i$. So there is no nontrivial projection of v on $\langle L_i \rangle_{\mathbf{Q}}$ which is in L_i.

Set $J = \{j \in \{1, \cdots, k\} \mid \pi_j(v) \neq 0\}$. Choose vectors $v_j \in I_j, j \in J$ such that $(v_j, v) \neq 0)$. The star $\{\,\{v_j, v\}\,\}_{j \in J} \subseteq \Gamma$ is connected and thus $\mathbf{Z}v + \sum_{j \in J} L_j$ is indecomposable. Further $\sum_{i \in I \setminus J} L_i \oplus (\sum_{j \in J} L_j + \mathbf{Z}v)$ is an orthogonal decomposition because $\pi_i(v) = 0$ for $i \notin J$.

This proves the correctness of algorithm 1. □

3.1. **Data structures and analysis.** We represent vectors as tuples of real numbers regardless of the machine precision or (possible) rational approximation. We count addition, multiplication and comparison of real numbers as a single arithmetic operation.

We use an ordered list as data structure for S.

A crucial rôle in this algorithm is played by the orthogonal sum $L_1 \oplus \ldots \oplus L_k$. Now we describe these data structures precomputed by induction and the update mechanism when processing v. Each L_i is described by a lattice basis $v_{i1}, \cdots, v_{ir_i}$. Further we have a vector space basis $w_1, \cdots, w_{n-l}$ of the orthogonal complement $\langle L_1, \cdots, L_k\rangle_{\mathbf{Q}}^{\perp}$ of $\bigoplus_i L_i$ with $l = \sum_i r_i$ (possibly empty). We write all these vectors as column vectors and set $A = \begin{pmatrix} v_{11} & \cdots & v_{kr_k} & w_1 & \cdots & w_{n-l}\end{pmatrix}^{-1}$.

For a $x \in V$ it holds $x \in \bigoplus_{i=1}^k L_i$ iff all the first l coefficients of Ax are in $\mathbf{Z}$ and the last $n - l$ coefficients are zero. So we can lookup a vector using $2n^2 + n$ arithmetic operations. If $v \in \bigoplus_{i=1}^k L_i$ we can skip v. Otherwise we have to compute $\tilde{L}$ with its lattice basis. A is a diagonal matrix with the blocks $\begin{pmatrix} v_{i1} & \cdots & v_{ir_i}\end{pmatrix}^{-1}$ on its diagonal. For each of the first l coefficients of Av which is not in $\mathbf{Z}$ we lookup its corresponding block (say j) and join j into J. At last we have to compute a basis of $\tilde{L}$, reorder the decomposition and update A.

We suggest to compute a basis of $\tilde{L}$ with a variation (e.g. the algorithms described in [Poh87] or [BP89], see section 4 for a discussion of these possibilities) of the well known LLL algorithm [LLL82]. $\tilde{L}$ has a generating system $\{v_{ji} \mid j \in J, 1 \leqslant i \leqslant r_j\} \cup \{v\}$ of at most $n + 1$ vectors. Now we reorder the list of lattices L_i such that they are indexed by 1 up to $k - |J| + 1$. At the end we update A.

The idea of this algorithm is based on the fact that almost all vectors $v \in S$ are already in $L_1 \oplus \ldots \oplus L_k$. The expensive construction of $\tilde{L}$ is rarely needed. We use Minkowski's second theorem [Min96] to prove a worst-case boundary. At first we recall some notions from the geometry of numbers.

We define the *determinant of* L as $\det L := \det((b_1 \cdots b_n)^t \cdot (b_1 \cdots b_n))$ for an arbitrary basis $\{b_1, \cdots, b_n\}$ of L. This is an invariant of the lattice and in particular independent from the choice of the basis. For $1 \leqslant k \leqslant n$ we define the *k-th successive minimum* $\lambda_k(L)$ as follows. There are linear independent $x_1, \cdots, x_k \in L$ with $||x_i|| \leqslant \lambda_k(L)$ and $\lambda_k(L)$ is minimal with this property.

Theorem 3.2. *Let L be a lattice on the n-dimensional euclidean space V with successive minima $\lambda_1(L), \cdots, \lambda_n(L)$, and $B_n = \{x \in V \mid ||x|| \leqslant 1\}$ the n-dimensional unit ball. Then,*

$$\frac{2^n}{n!} \det L \leqslant \lambda_1(L) \cdot \ldots \cdot \lambda_n(L) \cdot \operatorname{vol} B_n \leqslant 2^n \det L. \tag{1}$$

Proposition 3.3. *Let L be a lattice on the n-dimensional euclidean space V, $M =$ its minimum, and $L' \subseteq L$ a sublattice of rank n generated by vectors of norm at most B. Assume further that there is a finite chain of sublattices $L_i \subseteq L$ such that*

$$L' = L_1 \subset \cdots \subset L_t = L \tag{2}$$

Then

i) $t \leqslant \frac{1}{2}(\log_2 \frac{B^n}{\det L} + \log_2 \frac{\pi^{\frac{n}{2}} n!}{2^n \Gamma(\frac{n}{2}+1)})$,
ii) $t \leqslant \frac{1}{2}(\log_2 \frac{2^n \det L'}{M^n} - \log_2 \frac{\pi^{\frac{n}{2}} n!}{2^n \Gamma(\frac{n}{2}+1)})$,
iii) $t \leqslant \frac{1}{2}\log_2(n!(\frac{B}{M})^n)$.

Proof. i) We set $s_i = [L : L_i]$ for the index of L_i in L, $1 \leqslant i \leqslant t$. With the determinant-index formula $s_i^2 = [L : L_i]^2 = \frac{\det L_i}{\det L}$ we have

$$\sqrt{\det L' / \det L} = [L : L'] = s_1 > \cdots > s_t = [L : L] = 1.$$

$s_{i+1} \cdot [L_{i+1} : L_i] = s_i$ implies that s_{i+1} divides s_i. So the maximal number of lattices in (2) is bounded by the maximal possible length of a divisor chain of $\sqrt{\det L' / \det L}$. So we have an upper bound of $\log_2 \sqrt{\det L' / \det L}$ for t.

With Theorem 3.2 we have

$$\det L' \leqslant \frac{n!}{2^n} \operatorname{vol} B_n \cdot \lambda_1(L') \cdot \ldots \cdot \lambda_n(L') \leqslant \frac{n!}{2^n} \operatorname{vol} B_n B^n$$

and further

$$s_1^2 = [L : L']^2 = \frac{\det L'}{\det L} \leqslant \frac{n! \operatorname{vol} B_n}{2^n \det L} B^n.$$

Finally it follows with $\operatorname{vol} B_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}$ that $t \leqslant \log_2 \sqrt{\det L' / \det L} \leqslant \frac{1}{2}(\log_2 \frac{B^n}{\det L} + \log_2 \frac{\pi^{\frac{n}{2}} n!}{2^n \Gamma(\frac{n}{2}+1)})$.

ii) Similar to i) but using the left inequality in (1).
iii) Similar to i) and ii) but using both inequalities in (1).

□

Corollary 3.4. *Let L be a lattice with minimum M on the n-dimensional euclidean space V, and sublattices $L_{ij} \subseteq L, 0 \leqslant j \leqslant m_i$ of rank i generated by vectors of norm smaller than B such that*

$$L_{01} \subset L_{11} \subset L_{12} \subset \cdots \subset L_{n,m_n-1} \subset L_{nm_n} = L. \tag{3}$$

Then the length $\sum_{i=0}^{n} m_i$ of this chain is bounded by $n + \frac{1}{2}\log_2(n!(\frac{B}{M})^n)$.

Proof. Let w_i a vector such that $L_{i-1,m_{i-1}} + \mathbf{Z}w_i = L_{i1}$ and set $F = \langle w_1, \cdots, w_n \rangle_{\mathbf{Z}}$. We add F to each lattice in (3) and examine the new chain of lattices which are all of rank n

$$(F =)\, L_{01} + F \subseteq \ldots \subseteq L_{nm_n} + F\, (= L). \tag{4}$$

It is easy to verify that the following holds
i) $L_{ij} + F \subset L_{i,j+1} + F$ for all $0 \leqslant i \leqslant n$ and $j < m_i$,
ii) $L_{i-1,m_{i-1}} + F = L_{i1} + F$ for all $0 < i \leqslant n$.

This implies that in chain (4) equality holds in exactly n positions. We apply Proposition 3.3 iii) to (4) and see that the chain (3) has not more than $n + \frac{1}{2}\log_2(n!(\frac{B}{M})^n)$ lattices. □

We summarize this in the following “covering theorem”.

Theorem 3.5. *Let L be a lattice with minimum M on the n-dimensional euclidean space V. Let $S \subset L$ be a generating set for L of vectors all not longer than B. Then there is a subset $S' \subseteq S$ which has not more than $n + \frac{1}{2}\log_2(n!(\frac{B}{M})^n)$ elements with $\langle S' \rangle_{\mathbf{Z}} = L$.*

Proof. For $S = \{v_1, \cdots, v_s\}$ we set $L_i = \langle v_1, \cdots, v_i \rangle_{\mathbf{Z}}, 0 \leqslant i \leqslant n$. Now we define with $S' = \{v_i \in S \mid L_{i-1} \subset L_i\}$. Then $\langle S' \rangle_{\mathbf{Z}} = L$ and the stated bound follows from Corollary 3.4. □

Corollary 3.6. *The number of update operations of Algorithm 1 is in* $O(n \log \frac{nB}{M})$.

Proof. We apply Stirling's formula to Corollary 3.4. □

For our next theorem we heavily use the properties of LLL-reduced lattice bases.

Lemma 3.7. *Let* L *be an integral lattice on the* n*-dimensional euclidean space* V *generated by* $n+1$ *vectors, all not longer than* $\sqrt{B}$. *A LLL-reduced basis* $\{b_1, \cdots, b_n\}$ *of* L *can be computed in* $O(n^4 \log B)$ *arithmetic operations and in particular for each* $1 \leqslant j \leqslant n$ *holds*

$$||b_j|| \leqslant 2^{\frac{n-1}{2}} \cdot \lambda_j(L) \leqslant 2^{\frac{n-1}{2}} B.$$

Proof. With [BP89] Theorem 3.2. we can compute a lattice basis in at most $O((n + (n+1))^4 \log B) = O(n^4 \log B)$ arithmetic operations. This basis is LLL-reduced. Thus [LLL82] Proposition 1.12 gives us the stated bounds for the $||b_j||$. □

We are now ready to prove

Theorem 3.8. *Let* L *be an integral* n*-dimensional lattice, and* $B \in \mathbf{R}$ *such that* $S := \{v_1, \cdots, v_s\} = \{v \in L \setminus 0 \mid ||v|| \leqslant B\}$ *is a generating system of* L. *Then we can compute an orthogonal decomposition* $L = \bigoplus_{i=1}^{r} L_i$ *of* L *into indecomposable sublattices* $L_i \subseteq L$ *in at most* $O(n^6 \log^2 B + sn^2)$ *arithmetic operations.*

Proof. We analyze the running time of algorithm 1. We use the data structures described in section 3.1.

With radix-sort ([Knu74]) we sort all vectors of S with respect to their norm, shortest first, and put them into an ordered list in $O(sn)$ operations. Now let $v_{11}, \cdots, v_{kr_k}$, $w_1, \cdots, w_{n-l}$ and A be precomputed by induction. We pick a v of minimal norm from S and test the first $n - l$ coefficients of Av to be in $\mathbf{Z}$. The overall cost for all $v \in S$ is $O(sn^2)$. Corollary 3.6 and $\min L \geqslant 1$ imply that the number of update steps is at most $O(n \log(nB))$. In each update step we compute a lattice basis from at most $n+1$ vectors. These vectors are either vectors of S or a result from a previous LLL, in particular their norm is bounded by $2^{\frac{n-1}{2}} \lambda_n(L) \leqslant 2^{\frac{n-1}{2}} B$. Using Lemma 3.7 this costs $O(n^4 \log(2^{\frac{n-1}{2}} B)) = O(n^5 + n^4 \log B)$ arithmetic operations. The update of A can be done in $O(n^3)$ arithmetic operations using Gauss transformations (cf. e.g. [GL96]). This results in an overall running time of at most $O(n^6 \log(nB) + n^5 \log(nB) \log B)$ operations. For brevity we state here the slightly weaker bound of $O(n^6 \log^2(nB) + sn^2)$. □

Remark 3.9. Conway and Sloane found a lattice with a generating system of minmal vectors but but with no basis of minimal vectors, see [CS95]. Thus in general we can not avoid long vectors in lattice bases. However the LLL algorithm finds usually lattice bases much faster and better than stated in Lemma 3.7.

4. A MODFICATION OF THE MLLL FOR LARGE GENERATING SYSTEMS

A much more common problem than performing an orthogonal decomposition of lattices is the construction of a lattice basis from a generating system. There are some well known algorithms to do this. A LLL-reduced basis of a lattice with minimum M can be computed from a generating system $S = \{v_1, \cdots, v_s\}, ||v_i||^2 \leqslant B$ with (naïve) usage of Buchmann-Pohst's algorithm in at most $O((n+s)^4 \log \frac{B}{M})$ arithmetic operations, see [BP89]. This is not feasible for large s. Other possibilities are Pohst's MLLL algorithm [Poh87] and the computation of the Hermite Normal Form, but to the best of the authors' knowledge their running time has not be determined carefully yet. However, for large s all these algorithms slow down and one shouldn't use them without modifications on complete S, see Section 5 for examples. We can simplify Algorithm 1 to compute a lattice basis instead of a decomposition. The resulting Algorithm 2 uses a subalgorithm `construct_basis` to compute a lattice basis from a generating system of at most $n+1$ vectors. To prove Theorem 4.1 we choose Buchmann and Pohst's algorithm ([BP89]) but in practice we prefer the MLLL algorithm ([Poh87]) because it is a widely implemented and well performing algorithm. However we suggest to use some of the well known improvements of this algorithm if quality of the reduced basis is important (cf. e.g. [SE91]).

Algorithm 2 Computing a lattice basis from a generating system

```
Input: A generating sytem $S = \{v_1, \cdots, v_s\}$ of a lattice $L \subset V$ with $||v_i||^2 \leqslant B$.
Output: A lattice basis $\mathcal{B}$ of $L$.

// 1. Initialize.
Choose $0 \neq v \in S, S \leftarrow S \backslash \{v\}$, $\mathcal{B} \leftarrow \{v\}$
// 2. Add vectors vectors of $S$ successively.
while $S \neq \emptyset$ do
   Choose $v \in S$, and set $S \leftarrow S \backslash \{v\}$.
   if $v \notin \langle \mathcal{B} \rangle_{\mathbf{Z}}$ then
      $\mathcal{B} \leftarrow$ construct_basis($\mathcal{B} \cup \{v\}$)
   end if
end while
```

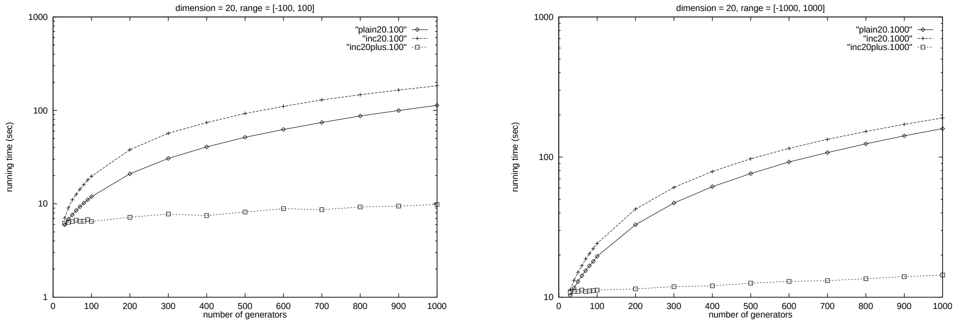
Theorem 4.1. *Let L be an integral lattice on the n-dimensional euclidean space, generated by s vectors not larger than $B \in \mathbf{R}$. Then Algorithm 2 computes a LLL-reduced basis of L in at most $O(n^6 \log^2(nB) + sn^2)$ arithmetic operations.*

Proof. We use the same data structures described in Section 3.1 and apply Buchmann and Pohst's method (see Lemma 3.7) for the subalgorithm `construct_basis`. It is clear that Algorithm 2 computes an LLL-reduced basis of L. The proof of it's running time is completely analogous to the proof of Theorem 3.8. □

5. Experimental Results

Lenstra, Lenstra, and Lovász proved in [LLL82] an upper bound for the running time of their algorithm of $O(n^4 \log B)$ arithmetic operations with the notations of section 4. Experiments shows that this bound is pessimistc. This holds for variations of the LLL like the MLLL, too. However, a large number of input vectors slow down the MLLL. The idea to generate successively bases from at most $n+1$ generators is not a competetive improvement. We call this algorithm *incremental MLLL* which is only a simplistic version of Algorithm 2, Figure 1 shows that the running times of the plain MLLL and the incremental MLLL are very similar. The situation changes when we use algorithm 2.

Our samples are lattices generated by vectors with pseudo-random integer coefficients of absolute value smaller than K. This implies in particular $B \leqslant nK^2$. It is obvious that the number of update steps which require an expensive MLLL is small with respect to s. A typical determinant of the first lattice of full rank is very large, e.g. for $\dim = 20, K = 1000$ we have determinants with two hundred binary digits. The final lattice has a very small determinant, often it is unimodular. It has not escaped our notice that the determinant



◊ MLLL, + incremental MLLL, □ Algorithm 2

FIGURE 1. Running time in dimension 20 for $K = 100$ and $K = 1000$

of a random integer lattices has usually a large prime divisor. Thus the number of update steps in Algorithm 2 is much smaller than stated in Corollary 3.6.

All computations are done in exact rational arithmetic using Victor Shoup's NTL library[1] and his implementation of the MLLL.

REFERENCES

[BP89] J. Buchmann and M. Pohst. Computing a lattice basis from a system of generating vectors. In *EUROCAL '87 (Leipzig, 1987)*, volume 378 of *Lecture Notes in Comput. Sci.*, pages 54–63. Springer, Berlin, 1989.

[CS95] J. H. Conway and N. J. A. Sloane. A lattice without a basis of minimal vectors. *Mathematika*, 42(1):175–177, 1995.

[Eic52] M. Eichler. Note zur Theorie der Kristallgitter. *Mathematische Annalen*, 125:51–55, 1952.

[GL96] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[Kne54] M. Kneser. Zur Theorie der Kristallgitter. *Mathematische Annalen*, 127:105–106, 1954.

[Knu74] D. E. Knuth. *The art of computer programming.*, volume 3: Sorting and searching. of *Addison-Wesley Series in Computer Science and Information Processing*. Addison-Wesley Publishing Company, 1974.

[LLL82] A. K. Lenstra, H. W. Jr. Lenstra, and L. Lov´asz. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[Min96] H. Minkowski. *Geometrie der Zahlen*. Teubner, Leipzig, 1896.

[Poh87] M. Pohst. A modification of the LLL reduction algorithm. *J. Symbolic Comput.*, 4(1):123–127, 1987.

[SE91] C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. In *Fundamentals of computation theory (Gosen, 1991)*, volume 529 of *Lecture Notes in Comput. Sci.*, pages 68–85. Springer, Berlin, 1991.

FACHBEREICH MATHEMATIK, UNIVERSITÄT DORTMUND, D-44221 DORTMUND
E-mail address: Boris.Hemkemeier@Math.Uni-Dortmund.DE

FACHBEREICH MATHEMATIK, UNIVERSITÄT DORTMUND, D-44221 DORTMUND
E-mail address: Frank.Vallentin@Math.Uni-Dortmund.DE

[1] URL: http://www.cs.wisc.edu/˜shoup/ntl/