Computing an LLL-reduced Basis of the Orthogonal Lattice

Jingwei Chen
Chongqing Key Lab of Automated Reasoning & Cognition, Chongqing Institute of Green and Intelligent Technology, CAS, Chongqing, China
chenjingwei@cigit.ac.cn

Damien Stehlé
Univ Lyon, ENS de Lyon, CNRS, Inria, Université Claude Bernard Lyon 1, LIP UMR 5668, F-69007
Lyon, France
damien.stehle@ens-lyon.fr

Gilles Villard
Univ Lyon, CNRS, ENS de Lyon, Inria, Université Claude Bernard Lyon 1, LIP UMR 5668, F-69007
Lyon, France
gilles.villard@ens-lyon.fr

ABSTRACT

As a typical application, the Lenstra-Lenstra-Lovász lattice basis reduction algorithm (LLL) is used to compute a reduced basis of the orthogonal lattice for a given integer matrix, via reducing a special kind of lattice bases. With such bases in input, we propose a new technique for bounding from above the number of iterations required by the LLL algorithm. The main technical ingredient is a variant of the classical LLL potential, which could prove useful to understand the behavior of LLL for other families of input bases.

CCS CONCEPTS

• **Computing methodologies → Symbolic and algebraic algorithms**;

KEYWORDS

Lattice basis reduction, LLL, orthogonal lattice, kernel lattice

ACM Reference Format:
Jingwei Chen, Damien Stehlé, and Gilles Villard. 2018. Computing an LLL-reduced Basis of the Orthogonal Lattice. In *ISSAC'18: 2018 ACM International Symposium on Symbolic and Algebraic Computation, July 16–19, 2018, New York, NY, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3208976.3209013

1 INTRODUCTION

Let $k < n$ be two positive integers. Given a full column rank $n \times k$ integer matrix $A = (a_{i,j})$, we study the behaviour of the Lenstra-Lenstra-Lovász algorithm [7] for computing a reduced basis for the *orthogonal lattice* of $\mathbf{A}$

$$\mathcal{L}^{\perp}(\mathrm{A}) = \left\{ \boldsymbol{m} \in \mathbb{Z}^n : \mathrm{A}^T \boldsymbol{m} = \mathbf{0} \right\} = \mathrm{Ker}(\mathrm{A}^T) \cap \mathbb{Z}^n. \qquad (1)$$

The algorithm proceeds by unimodular column transformations from the input matrix $\mathrm{Ext}_K(\mathbf{A}) \in \mathbb{Z}^{(n+k)\times n}$:

$$\mathrm{Ext}_K(\mathbf{A}) := \begin{pmatrix} K \cdot \mathbf{A}^T \\ \mathbf{I}_n \end{pmatrix} = \begin{pmatrix} K\cdot a_{1,1} & K\cdot a_{2,1} & \cdots & K\cdot a_{n,1} \\ \vdots & \vdots & \ddots & \vdots \\ K\cdot a_{1,k} & K\cdot a_{2,k} & \cdots & K\cdot a_{n,k} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix}. \qquad (2)$$

where K is a sufficiently large positive integer. The related definitions and the LLL algorithm are given in Section 2. The reader may refer to [11] for a comprehensive review of LLL, and to [14] and [9] concerning the orthogonal lattice.

Usual techniques gives that LLL reduction requires $O(n^2 \log(K \cdot \|\mathbf{A}\|))$ *swaps* (see Step 7 of Algorithm 1) for a basis as in (2), where $\|\mathbf{A}\|$ bounds from above the Euclidean norms of the rows and columns of $\mathbf{A}$. We recall that most known LLL reduction algorithms iteratively perform two types of vector operations: translations and swaps. The motivation for studying bounds on the number of swaps comes from the fact that this number governs known cost analyses of the reduction.

Folklore applications of the reduction of bases as in (2) include, for example, the computation of integer relations between real numbers [1, 3], the computation of minimal polynomials [6] (see also [11]). A main difficulty however, both theoretically and practically, remains to master the *scaling parameter* K that can be very large. Heuristic and practical solutions may for instance rely on a doubling strategy (successive trials with $K = 2, 2^2, 2^4, \ldots$) for finding a suitable scaling. Or an appropriate value for K may be derived from *a priori* bounds such as heights of algebraic numbers [6] and may overestimate the smallest suitable value for actual inputs. Since the usual bound on the number of swaps is linear in $\log K$, the overestimation could be a serious drawback. We show that this may not be always the case.

We consider the reduction of a basis as in (2) for obtaining a basis of the orthogonal lattice (1). We establish a bound on the number of swaps that does not depend on K as soon as K is above a threshold value (as specified in (7)). This threshold depends only on the dimension and invariants of the orthogonal lattice.

Our contribution. The analyses of LLL and many LLL variants bound the number of iterations using the geometric decrease of a potential that is defined using the Gram-Schmidt norms of the basis vectors; see (6). We are going to see that this classical potential does not capture a typical unbalancedness of the Gram-Schmidt

ISSAC'18, July 16–19, 2018, New York, NY, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5550-6/18/07...$15.00
https://doi.org/10.1145/3208976.3209013

norms that characterizes bases in (2). Taking into account the latter structure will lead us to a better bound for the number of iterations (see Table 1). Intuitively, as the basis being manipulated becomes reduced, two groups of vectors are formed: some with small Gram-Schmidt norms, and some others with large Gram-Schmidt norms. As soon they are formed, the two groups do not interfere much.

In Section 3 we introduce a new LLL potential function that generalizes the classical one for capturing the previously mentioned unbalancedness. Its geometric decrease during the execution also leads to a bound on the number of iterations (see Theorem 3.3). In Section 4, we specialize the potential to the case of bases as in (2) for computing the orthogonal lattice $\mathcal{L}^\perp(\mathbf{A})$. As discussed above, we will see that at some point the number of iterations can be shown to be independent of the scaling parameter K, or, in other words, independent of a further increase of the input size. We note that this new potential is defined for all lattice bases, but it may not always lead to better bounds on the number of LLL iterations.

The extended gcd algorithm in [4] uses a basis as in (2) with $k = 1$. It is shown in [4, Sec. 3, p. 127] that if K is sufficiently large, then the sequence of operations performed by LLL is independent of K. A somewhat similar remark had been made in [13]. We also note that in the analysis of the gradual sub-lattice reduction algorithm of [5], a similar separation of large and small basis vectors was used, also for a better bound on the number of iterations. Our new potential function allows a better understanding of the phenomenon.

We see our potential function for LLL as a new complexity analysis tool that may help further theoretical and practical studies of LLL and its applications. Various approaches exist for computing the orthogonal lattice $\mathbf{A}$, or equivalently an integral kernel basis of $\mathbf{A}^T$. A detailed comparison of the methods remains to be done and would be however outside the scope of this paper that focuses on the properties of the potential. An integral kernel basis may be obtained from a unimodular multiplier for the Hermite normal form of $\mathbf{A}$ [19] (see also [18] for the related linear system solution problem), which may be combined as in [15, Ch. 8] and [2] with LLL for minimizing the bit size of the output. A direct application of LLL to $\mathrm{Ext}_K(\mathbf{A})$ is an important alternative solution. We refer to [16] and references therein concerning existing LLL variants.

Future work. Future research directions are to apply this potential to bit complexity studies of the LLL basis reduction [8, 12, 17], especially for specific input bases. Indeed, an interesting problem is to design an algorithm for computing a reduced basis for $\mathcal{L}^\perp(\mathbf{A})$ that features a bit complexity bound independent of the scaling parameter, and to compare it to approaches based on the Hermite normal form.

Notations. Throughout the paper, vectors are in column and denoted in bold. For $\boldsymbol{x} \in \mathbb{R}^m$, $\|\boldsymbol{x}\|$ is the Euclidean norm of $\boldsymbol{x}$. Matrices are denoted by upper case letters in bold, such as $\mathbf{A}$, $\mathbf{B}$, etc. For a matrix $\mathbf{A}$, $\mathbf{A}^T$ is the transpose of $\mathbf{A}$, and $\|\mathbf{A}\|$ bounds the Euclidean norms of the columns and rows of $\mathbf{A}$. The base of logarithm is 2.

2 PRELIMINARIES

We give some basic definitions and results that are needed for the rest of the paper. A comprehensive presentation of the LLL algorithm and its applications may be found in [11].

Gram-Schmidt orthogonalization. Let $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n \in \mathbb{R}^m$ be linearly independent vectors. Their *Gram-Schmidt orthogonalization* $\boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_n^*$ is defined as follows:

$$\boldsymbol{b}_1^* = \boldsymbol{b}_1 \text{ and } \forall i > 1 : \boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^*,$$

where the $\mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\langle \boldsymbol{b}_j^*, \boldsymbol{b}_j^* \rangle}$ for all $i > j$ are called the *Gram-Schmidt coefficients*. We call the $\|\boldsymbol{b}_i^*\|$'s the *Gram-Schmidt norms* of the $\boldsymbol{b}_i$'s.

Lattices. A *lattice* $\Lambda \subseteq \mathbb{R}^m$ is a discrete additive subgroup of $\mathbb{R}^m$. If $(\boldsymbol{b}_i)_{i \le n}$ is a set of generators for Λ, then

$$\Lambda = \mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \left\{ \sum_{i=1}^{n} z_i \boldsymbol{b}_i \,:\, z_i \in \mathbb{Z} \right\}.$$

If the $\boldsymbol{b}_i$'s are linearly independent, then they are said to form a *basis* of Λ. When $n \ge 2$, there exist infinitely many bases for a lattice. Every basis is related by an integral unimodular transformation (a linear transformation with determinant ±1) to any other. Further, the number of vectors of different bases of a lattice Λ is always the same, and we call this number the *dimension* of the lattice, denoted by $\dim(\Lambda)$. If $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{m \times n}$ is a basis for a lattice $\Lambda = \mathcal{L}(\mathbf{B})$, the *determinant* of the lattice is defined as $\det(\Lambda) = \sqrt{\det(\mathbf{B}^T\mathbf{B})}$. It is invariant across all bases of Λ.

Successive minima. For a given lattice Λ, we let $\lambda_1(\Lambda)$ denote the minimum Euclidean norm of vectors in $\Lambda \setminus \{\mathbf{0}\}$. From Minkowski's first theorem, we have $\lambda_1(\Lambda) \le \sqrt{n} \cdot \det(\Lambda)^{1/n}$, where $n = \dim(\Lambda)$. More generally, for all $1 \le i \le n$, we define the i-th *minimum* as

$$\lambda_i(\Lambda) = \min_{\substack{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_i \in \Lambda \\ \text{linearly independent}}} \max_{j \le i} \|\boldsymbol{v}_j\|.$$

Minkowski's second theorem states that $\prod_{i \le n} \lambda_i(\Lambda) \le \sqrt{n}^n \cdot \det(\Lambda)$.

Sublattices. Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice. We say that Λ' is a *sublattice* of Λ if $\Lambda' \subseteq \Lambda$ is a lattice as well. If Λ' is a sublattice of Λ then $\lambda_i(\Lambda) \le \lambda_i(\Lambda')$ for $i \le \dim(\Lambda')$. A sublattice Λ' of $\Lambda \subset \mathbb{R}^n$ is said to be *primitive* if there exists a subspace E of $\mathbb{R}^n$ such that $\Lambda' = \Lambda \cap E$.

Orthogonal lattices. Given a full column rank matrix $\mathbf{A} \in \mathbb{Z}^{n \times k}$, the set $\mathcal{L}^\perp(\mathbf{A})$ defined in (1) forms a lattice, called the *orthogonal lattice* of $\mathbf{A}$. We have $\dim(\mathcal{L}^\perp(\mathbf{A})) = n - k$. Using $\ker(\mathbf{A}^T)^\perp = \mathrm{Im}(A)$ and [14, Cor. p. 328] for primitive lattices we have

$$\det(\mathcal{L}^\perp(\mathbf{A})) = \det(\mathbb{Z}^n \cap \ker(\mathbf{A}^T)) = \det(\mathbb{Z}^n \cap \mathrm{Im}(\mathbf{A})),$$

then $\mathcal{L}(\mathbf{A}) \subseteq \mathbb{Z}^n \cap \mathrm{Im}(\mathbf{A})$ and Hadamard's inequality lead to:

$$\det(\mathcal{L}^\perp(\mathbf{A})) \le \det(\mathcal{L}(\mathbf{A})) \le \|\mathbf{A}\|^k. \tag{3}$$

LLL-reduced bases. The goal of lattice basis reduction is to find a basis with vectors as short and orthogonal to each other as possible. Among numerous lattice reduction notions, the LLL-reduction [7] is one of the most commonly used. Let $\frac{1}{4} < \delta < 1$. Let $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{m \times n}$ be a basis of a lattice Λ. We say that $\mathbf{B}$ is *size-reduced* if all Gram-Schmidt coefficients satisfy $|\mu_{ij}| \le \frac{1}{2}$. We say that $\mathbf{B}$ satisfies the *Lovász conditions* if for all i we have $\delta \|\boldsymbol{b}_i^*\|^2 \le \|\boldsymbol{b}_{i+1}^*\|^2 + \mu_{i+1,i}^2 \|\boldsymbol{b}_i^*\|^2$. If a basis $\mathbf{B}$ is size-reduced and satisfies the Lovász conditions, then we say that $\mathbf{B}$ is *LLL-reduced* (with respect to the parameter δ). If a basis $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ of Λ is LLL-reduced, then we have:

$$\forall i < n, \|\boldsymbol{b}_i^*\|^2 \le \alpha \|\boldsymbol{b}_{i+1}^*\|^2,$$

$$\forall i \le n, \|\boldsymbol{b}_i\|^2 \le \alpha^{i-1} \|\boldsymbol{b}_i^*\|^2, \tag{4}$$

$$\forall i \le j \le n, \|\boldsymbol{b}_i\| \le \alpha^{\frac{n-1}{2}} \lambda_j(\Lambda), \tag{5}$$

where $\alpha = \frac{4}{4\delta - 1}$. In particular, we have $\|\boldsymbol{b}_1\| \le \alpha^{\frac{n-1}{2}} \lambda_1(\Lambda)$. In this paper, we use the original LLL parameter $\delta = \frac{3}{4}$ and hence $\alpha = 2$.

The LLL algorithm. We now sketch the LLL algorithm. Although there exist many LLL variants in the literature, most of them follow the following structure. Step 7 is called an *LLL swap*.

Algorithm 1 (LLL)

```
Input: A basis $(\boldsymbol{b}_i)_{i \le n}$ of a lattice $\Lambda \subseteq \mathbb{Z}^n$.
Output: An LLL-reduced basis of $\Lambda$.
 1: $i := 2$;
 2: while $i \le n$ do
 3:     Size-reduce $\boldsymbol{b}_i$ by $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{i-1}$;
 4:     if Lovász condition holds for $i$ then
 5:         Set $i := i + 1$;
 6:     else
 7:         (LLL swap) Swap $\boldsymbol{b}_i$ and $\boldsymbol{b}_{i-1}$; set $i := \max\{i - 1, 2\}$;
 8:     end if
 9: end while
10: Return $(\boldsymbol{b}_i)_{i \le n}$.
```

To clarify the structure of the algorithm, we omit some details in the above description, e.g., the update of Gram-Schmidt coefficients. From the sketch, we see that we can bound the running-time of LLL by the number of while loop iterations times the cost of each iteration. In fact, most cost bounds for LLL variants proceed via this simple argument. It was showed in [7] that the number of LLL swaps is $O(n^2 \log \|\mathbf{B}\|)$. The following lemma plays a very important role in the analysis of LLL; see [7] for a proof.

Lemma 2.1. *Let* $\mathbf{B}$ *and* $\mathbf{B}'$ *be bases after and before an LLL swap between* $\boldsymbol{b}_i$ *and* $\boldsymbol{b}_{i+1}$*. Then*

$$\begin{array}{rcl}
\max\{\|\boldsymbol{b}_i'^*\|, \|\boldsymbol{b}_{i+1}'^*\|\} & \le & \max\{\|\boldsymbol{b}_i^*\|, \|\boldsymbol{b}_{i+1}^*\|\}, \\
\min\{\|\boldsymbol{b}_i'^*\|, \|\boldsymbol{b}_{i+1}'^*\|\} & \ge & \min\{\|\boldsymbol{b}_i^*\|, \|\boldsymbol{b}_{i+1}^*\|\}, \\
\|\boldsymbol{b}_i^*\| \cdot \|\boldsymbol{b}_{i+1}^*\| & = & \|\boldsymbol{b}_i'^*\| \cdot \|\boldsymbol{b}_{i+1}'^*\|, \\
\dfrac{\|\boldsymbol{b}_{i+1}'^*\|}{\|\boldsymbol{b}_{i+1}^*\|} = \dfrac{\|\boldsymbol{b}_i^*\|}{\|\boldsymbol{b}_i'^*\|} & \ge & \dfrac{2}{\sqrt{3}}, \\
\forall j \notin \{i, i+1\} & : & \boldsymbol{b}_j'^* = \boldsymbol{b}_j^*.
\end{array}$$

3 A NEW POTENTIAL

In this section, we introduce a variant of the classical LLL potential

$$\Pi(\mathbf{B}) = \sum_{i=1}^{n-1} (n - i) \log \|\boldsymbol{b}_i^*\| \tag{6}$$

of a lattice basis $\mathbf{B}$. The variant we introduce is well-suited for analyzing the number of LLL swaps for the case that both the input and output bases have k large Gram-Schmidt norms and $n - k$ small Gram-Schmidt norms, for some $k < n$. This is for example the case for the input basis as (2); see Section 4.2. The new potential is aimed at accurately measuring the progress made during the LLL execution, for such unbalanced bases.

Definition 3.1. Let $k \le n \le m$ be positive integers and $\mathbf{B} \in \mathbb{R}^{m \times n}$ be full column rank. We let $s_1 < \ldots < s_{n-k}$ be the indices of the $n - k$ smallest Gram-Schmidt norms of $\mathbf{B}$ (using the lexicographical in case there are several $(n - k)$-th smallest Gram-Schmidt norms), and set $S = \{s_i\}_{i \le n-k}$. We let $\ell_1 < \ldots < \ell_k$ be the indices of the other k Gram-Schmidt norms, and set $L = \{\ell_j\}_{j \le k}$. The *k-th LLL potential* of $\mathbf{B}$ is defined as:

$$\Pi_k(\mathbf{B}) = \sum_{j=1}^{k-1} (k - j) \log \|\boldsymbol{b}_{\ell_j}^*\| - \sum_{i=1}^{n-k} i \log \|\boldsymbol{b}_{s_i}^*\| + \sum_{i=1}^{n-k} s_i.$$

Note that for $k = n$, we recover the classical potential Π. The rationale behind Π_k is that in some cases we know that the output basis is made of vectors of very unbalanced Gram-Schmidt norms. As this basis is reduced, this means the first vectors have a small Gram-Schmidt norm, while the last vectors have large Gram-Schmidt norms. During the execution of LLL, such short and large vectors do not interfere much. This is an unusual phenomenon: most often, long vectors are made shorter and short vectors are made longer, so that they are all balanced at the end. But this can happen if the long vectors are rather orthogonal to the short ones. When this is the case, LLL actually runs faster than usual, because it merely "sorts" the short vectors and the long vectors, without making them interact to create shorter vectors. Of course, it can do more intense computations among the short vectors and among the long vectors. Unbalancedness of Gram-Schmidt norms is not captured by the classical potential, but it is with Π_k. In particular, the new potential Π_k allows to not "pay" for the output unbalancedness in the analysis of the number of LLL swaps.

Similarly to the classical potential, the k-th LLL potential monotonically decreases with the number of LLL swaps. More precisely, we have the following

Proposition 3.2. *Let* $\mathbf{B}$ *and* $\mathbf{B}'$ *be the current* n*-dimensional lattice bases before and after an LLL swap. Then for any* $k \le n$*, we have* $\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}') \ge \log(2/\sqrt{3})$.

Proof. Recall that S and L are the index sets for the $n - k$ Gram-Schmidt norms and the other k Gram-Schmidt norms for the lattice basis $\mathbf{B}$. We define S' and L' for $\mathbf{B}'$ similarly.

Suppose that this LLL swap occurs between $\boldsymbol{b}_\kappa$ and $\boldsymbol{b}_{\kappa+1}$. Then we must be in one of the following four cases.

Case 1: $\kappa \in S$ and $\kappa + 1 \in S$.

Let $i_0 \le n - k$ such that $\kappa = s_{i_0}$ and $\kappa + 1 = s_{i_0+1}$. From Lemma 2.1, we have $S' = S$ and $L' = L$, and hence $\kappa = s'_{i_0}$ and $\kappa + 1 = s'_{i_0+1}$. For the other indices, we have $s'_i = s_i$ (for $i \le n - k$) and $\ell'_j = \ell_j$ (for

$j \le k$). Then

$$\begin{aligned}\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}') &= \sum_{j=1}^{k}(k-j)\log\frac{\|\boldsymbol{b}^*_{\ell_j}\|}{\|\boldsymbol{b}'_{\ell'_j}\|} + \sum_{i=1}^{n-k} i\log\frac{\|\boldsymbol{b}'^*_{s'_i}\|}{\|\boldsymbol{b}^*_{s_i}\|} \\ &\quad + \sum_{i=1}^{n-k}\left(s_i - s'_i\right) \\ &= i_0\log\frac{\|\boldsymbol{b}'^*_{s'_{i_0}}\|}{\|\boldsymbol{b}^*_{s_{i_0}}\|} + (i_0+1)\log\frac{\|\boldsymbol{b}'^*_{s'_{i_0+1}}\|}{\|\boldsymbol{b}^*_{s_{i_0+1}}\|} \\ &= \log\frac{\|\boldsymbol{b}'^*_{\kappa+1}\|}{\|\boldsymbol{b}^*_{\kappa+1}\|} \ge \log\left(\frac{2}{\sqrt{3}}\right),\end{aligned}$$

where the last inequality follows from Lemma 2.1.

Case 2: $\kappa \in L$ and $\kappa + 1 \in L$.

The treatment of Case 1 can be adapted readily.

Case 3: $\kappa \in L$, $\kappa + 1 \in S$, $S' = S$ and $L' = L$.

Let $j_0 \le k$ such that $\kappa = \ell_{j_0}$, and $i_0 \le n - k$ such that $\kappa + 1 = s_{i_0}$. Then we have $\kappa = \ell'_{j_0}$ and $\kappa + 1 = s'_{i_0}$. For the other indices, we have $s'_i = s_i^{(t)}$ (for $i \le n - k$) and $\ell'_j = \ell_j^{(t)}$ (for $j \le k$). Thus

$$\begin{aligned}\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}') &= \sum_{j=1}^{k}(k-j)\log\frac{\|\boldsymbol{b}^*_{\ell_j}\|}{\|\boldsymbol{b}'^*_{\ell'_j}\|} + \sum_{i=1}^{n-k} i\log\frac{\|\boldsymbol{b}'^*_{s'_i}\|}{\|\boldsymbol{b}^*_{s_i}\|} \\ &\quad + \sum_{i=1}^{n-k}\left(s_i - s'_i\right) \\ &= (k-j_0)\log\frac{\|\boldsymbol{b}^*_{\ell_{j_0}}\|}{\|\boldsymbol{b}'^*_{\ell'_{j_0}}\|} + i_0\log\frac{\|\boldsymbol{b}'^*_{s'_{i_0}}\|}{\|\boldsymbol{b}^*_{s_{i_0}}\|} \\ &= (k-j_0+i_0)\log\frac{\|\boldsymbol{b}'^*_{\kappa+1}\|}{\|\boldsymbol{b}^*_{\kappa+1}\|} \ge \log\left(\frac{2}{\sqrt{3}}\right),\end{aligned}$$

where the last inequality follows from Lemma 2.1 and the fact that $k - j_0 + i_0 \ge 1$.

Case 4: $\kappa \in L, \kappa{+}1 \in S, S' = S \cup \{\kappa\} \backslash \{\kappa{+}1\}$ and $L' = L \cup \{\kappa{+}1\} \backslash \{\kappa\}$.

Let $j_0 \le k$ such that $\kappa = \ell_{j_0}$, and $i_0 \le n - k$ such that $\kappa + 1 = s_{i_0}$. Then $\kappa = s'_{i_0}$ and $\kappa + 1 = \ell'_{j_0}$. For other indices, we have $s'_i = s_i$ (for $i \le n - k$) and $\ell'_j = \ell_j$ (for $j \le k$). Then

$$\begin{aligned}\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}') &= \sum_{j=1}^{k}(k-j)\log\frac{\|\boldsymbol{b}^*_{\ell_j}\|}{\|\boldsymbol{b}'^*_{\ell'_j}\|} + \sum_{i=1}^{n-k} i\log\frac{\|\boldsymbol{b}'^*_{s'_i}\|}{\|\boldsymbol{b}^*_{s_i}\|} \\ &\quad + \sum_{i=1}^{n-k}\left(s_i - s'_i\right) \\ &= (k-j_0)\log\frac{\|\boldsymbol{b}^*_{\ell_{j_0}}\|}{\|\boldsymbol{b}'^*_{\ell'_{j_0}}\|} + i_0\log\frac{\|\boldsymbol{b}'^*_{s'_{i_0}}\|}{\|\boldsymbol{b}^*_{s_{i_0}}\|} + 1 \\ &= (k-j_0)\log\frac{\|\boldsymbol{b}^*_{\kappa}\|}{\|\boldsymbol{b}'^*_{\kappa+1}\|} + i_0\log\frac{\|\boldsymbol{b}'^*_{\kappa}\|}{\|\boldsymbol{b}^*_{\kappa+1}\|} + 1 \\ &\ge 1,\end{aligned}$$

where the last inequality follows from Lemma 2.1. The observation that $1 \ge \log(2/\sqrt{3})$ allows to complete the proof. □

With the above property of the k-th LLL potential, we can bound the number of LLL swaps that LLL performs.

Theorem 3.3. *Let $\mathbf{B} \in \mathbb{R}^{m\times n}$ be a full column rank matrix. Let $\mathbf{B}'$ be the basis returned by the LLL algorithm when given $\mathbf{B}$ as input. Then the number of swaps that LLL performs is no greater than*

$$\min_{1\le k\le n}\frac{\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}')}{\log\left(\frac{2}{\sqrt{3}}\right)}.$$

4 ORTHOGONAL LATTICES

As an application of the k-th LLL potential Π_k, we consider the problem of computing an LLL-reduced basis of an orthogonal lattice. Let $\mathbf{A} \in \mathbb{Z}^{n\times k}$ with $n \ge k$. We aim at computing an LLL-reduced basis of the orthogonal lattice $\mathcal{L}^{\perp}(\mathbf{A})$, by LLL-reducing $\mathrm{Ext}_K(\mathbf{A})$ (as defined in (2)), for a sufficiently large integer K.

In Subsection 4.1, we provide a sufficient condition on the scaling parameter K so that a LLL-reduced basis of $\mathcal{L}^{\perp}(\mathbf{A})$ can be extracted from a LLL-reduced basis of $\mathcal{L}(\mathrm{Ext}_K(\mathbf{A}))$. For such a sufficiently large K, we study the Gram-Schmidt orthogonalizations of the input and output bases of the LLL call to $\mathrm{Ext}_K(\mathbf{A})$ in Subsection 4.2, and we provide a bound on the number of required LLL swaps which is independent of K in Subsection 4.3.

4.1 Correctness

For $n \ge k$, we define $\sigma_{n,k}$ as the map that embeds $\mathbb{R}^n$ into $\mathbb{R}^{n+k}$ by adding 0's in the first k coordinates.

$$\begin{array}{rcl}\sigma_{n,k}:\ \mathbb{R}^n & \to & \mathbb{R}^{n+k} \\ (x_1,\cdots,x_n)^T & \mapsto & (\underbrace{0,\cdots,0}_{k},\underbrace{x_1,\cdots,x_n}_{n})^T.\end{array}$$

We also define $\delta_{n,k}$ as the map that erases the first k coordinates of a vector in $\mathbb{R}^{n+k}$.

$$\begin{array}{rcl}\delta_{n,k}:\ \mathbb{R}^{n+k} & \to & \mathbb{R}^n \\ (x_1,\cdots,x_k,x_{k+1},\cdots,x_{k+n})^T & \mapsto & (x_{k+1},\cdots,x_{k+n})^T.\end{array}$$

We extend these functions to matrices in the canonical way. The following proposition is adapted from [9, Theorem 4] (see also [10, Proposition 2.24]). It shows that if K is sufficiently large, then calling the LLL algorithm on $\mathrm{Ext}_K(\mathbf{A})$ provides an LLL-reduced basis of $\mathcal{L}^{\perp}(\mathbf{A})$.

Proposition 4.1. *Let $\mathbf{A} \in \mathbb{Z}^{n\times k}$ be full column rank and $\mathbf{B} = \mathrm{Ext}_K(\mathbf{A})$. If $\mathbf{B}'$ is an LLL-reduced basis of $\mathcal{L}(\mathbf{B})$ and*

$$K > 2^{\frac{n-1}{2}}\cdot\lambda_{n-k}(\mathcal{L}^{\perp}(\mathbf{A})), \tag{7}$$

then $\delta_{n,k}(\boldsymbol{b}'_1),\cdots,\delta_{n,k}(\boldsymbol{b}'_{n-k})$ is an LLL-reduced basis of $\mathcal{L}^{\perp}(\mathbf{A})$.

Proof. As $\mathbf{A} \in \mathbb{Z}^{n\times k}$ is full column rank, we have $\dim(\mathcal{L}^{\perp}(\mathbf{A})) = n-k$. For any basis $\mathbf{C} \in \mathbb{Z}^{n\times(n-k)}$ of $\mathcal{L}^{\perp}(\mathbf{A})$, we have $\sigma_{n,k}(\mathbf{C}) = \mathbf{B}\cdot\mathbf{C}$, and hence the lattice $\sigma_{n,k}(\mathcal{L}^{\perp}(\mathbf{A}))$ is a sublattice of $\mathcal{L}(\mathbf{B})$. This implies that, for all $i \le n-k$,

$$\lambda_i(\mathcal{L}(\mathbf{B})) \le \lambda_i(\sigma_{n,k}(\mathcal{L}^{\perp}(\mathbf{A}))) = \lambda_i(\mathcal{L}^{\perp}(\mathbf{A})).$$

It follows from (5) that, for all $i \le n-k$,

$$\|\boldsymbol{b}'_i\|^2 \le 2^{n-1}\cdot\lambda^2_{n-k}(\mathcal{L}(\mathbf{B})) \le 2^{n-1}\cdot\lambda^2_{n-k}(\mathcal{L}^{\perp}(\mathbf{A})). \tag{8}$$

We now assume (by contradiction) that $\delta_{n,k}(\boldsymbol{b}'_i) \notin \mathcal{L}^{\perp}(\mathbf{A})$ for some $i \leq n-k$. Note that

$$\boldsymbol{b}'_i = \mathbf{B} \cdot \delta_{n,k}(\boldsymbol{b}'_i) = (K \cdot \delta_{n,k}(\boldsymbol{b}_{i'})^T \cdot \mathbf{A} \mid \delta_{n,k}(\boldsymbol{b}'_i)^T)^T.$$

As the subvector $K \cdot \delta_{n,k}(\boldsymbol{b}'_i)^T \cdot \mathbf{A}$ is non-zero, and using the assumption on K, we obtain that

$$\begin{aligned}\|\boldsymbol{b}'_i\|^2 &= \|K \cdot \delta_{n,k}(\boldsymbol{b}'_i)^T \cdot \mathbf{A}\|^2 + \|\delta_{n,k}(\boldsymbol{b}'_i)\|^2 \\ &\geq K^2 > 2^{n-1} \cdot \lambda^2_{n-k}(\mathcal{L}^{\perp}(\mathbf{A})),\end{aligned}$$

which contradicts (8).

From the above, we obtain that $\delta_{n,k}(\boldsymbol{b}'_1), \cdots, \delta_{n,k}(\boldsymbol{b}'_{n-k})$ are linearly independent vectors in $\mathcal{L}^{\perp}(\mathbf{A})$. They actually form a basis of $\mathcal{L}^{\perp}(\mathbf{A})$. To see this, consider an arbitrary vector $\boldsymbol{c} \in \mathcal{L}^{\perp}(\mathbf{A})$. The vector $\mathbf{B} \cdot \boldsymbol{c}$ belongs to the real span of $\boldsymbol{b}'_1, \cdots, \boldsymbol{b}'_{n-k}$ and to $\mathcal{L}(\mathbf{B})$. As $\mathbf{B}'$ is a basis of $\mathcal{L}(\mathbf{B})$, vector $\mathbf{B} \cdot \boldsymbol{c}$ is an integer combination of $\boldsymbol{b}'_1, \cdots, \boldsymbol{b}'_{n-k}$ and vector $\boldsymbol{c}$ is an integer combination of $\delta_{n,k}(\boldsymbol{b}'_1), \cdots, \delta_{n,k}(\boldsymbol{b}'_{n-k})$.

Since $\mathbf{B}'$ is LLL-reduced and the first k coordinates of each of $\boldsymbol{b}'_1, \cdots, \boldsymbol{b}'_{n-k}$ are 0, we obtain that $\delta_{n,k}(\boldsymbol{b}'_1), \cdots, \delta_{n,k}(\boldsymbol{b}'_{n-k})$ form an LLL-reduced basis of $\mathcal{L}^{\perp}(\mathbf{A})$. □

To make this condition on K effective, we use some upper bounds on $\lambda_{n-k}(\mathcal{L}^{\perp}(\mathbf{A}))$. For instance, from Minkowski's second theorem, we have

$$\lambda_{n-k}(\mathcal{L}^{\perp}(\mathbf{A})) \leq (n-k)^{\frac{n-k}{2}} \cdot \det(\mathcal{L}^{\perp}(\mathbf{A})) \leq (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k.$$

Hence

$$K > 2^{\frac{n-1}{2}} \cdot (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k \tag{9}$$

suffices to guarantee that (7) holds.

The bound in (9) can be very loose. Indeed, in many cases, we expect the minima of $\mathcal{L}^{\perp}(\mathbf{A})$ to be balanced, and if they are so, then the following bound would suffice

$$K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}. \tag{10}$$

For such a scaling paramter K, according to Proposition 4.1, after termination of the LLL call with $\mathrm{Ext}_K(A)$ as its input, the output matrix must be of the following form:

$$\begin{pmatrix} \mathbf{0} & \mathbf{M} \\ \mathbf{C} & \mathbf{N} \end{pmatrix}, \tag{11}$$

where the columns of $\mathbf{C} \in \mathbb{Z}^{n \times (n-k)}$ form an LLL-reduced basis of the lattice $\mathcal{L}^{\perp}(\mathbf{A})$. [1]

4.2 On the LLL input and output bases

To bound the number of LLL swaps, we first investigate the matrix $\mathbf{B} = \mathrm{Ext}_K(\mathbf{A})$ given as input to the LLL algorithm, and the output matrix $\mathbf{B}'$.

Intuitively, from the shape of $\mathbf{B}$ and the fact that $\mathbf{A}$ is full rank, there must be k Gram-Schmidt norms of $\mathbf{B}$ that are "impacted" by the scaling parameter K, and hence have large magnitude, while other $n-k$ Gram-Schmidt norms of $\mathbf{B}$ should be of small magnitude.

On the other hand, recall that $\mathbf{B}'$ is of the form (11). Since only the first k coordinates are related to the scaling parameter K, the submatrix $\mathbf{C}$ is "independent" of K. Thus, each of $\|\boldsymbol{b}'^*_1\|, \cdots, \|\boldsymbol{b}'^*_{n-k}\|$ should be relatively small (for a sufficiently large K), while each of $\|\boldsymbol{b}'^*_{n-k+1}\|, \cdots, \|\boldsymbol{b}'^*_n\|$ is "impacted" by K, and hence with large magnitude. The following result formalizes this discussion.

Proposition 4.2. *Let* $\mathbf{A} \in \mathbb{Z}^{n \times k}$ *be of full column rank and* $\mathbf{B}'$ *the output basis of LLL with* $\mathbf{B} = \mathrm{Ext}_K(\mathbf{A})$ *as input. If the scaling parameter* $K \in \mathbb{Z}$ *satisfies* (7)*, then for the output matrix* $\mathbf{B}'$ *we have*

$$\forall i \leq n-k, \quad \forall j > n-k, \quad \|\boldsymbol{b}'^*_i\| < \|\boldsymbol{b}'^*_j\|.$$

Proof. From Proposition 4.1, we know that $\mathbf{B}'$ is of the form

$$\begin{pmatrix} \mathbf{0} & * \\ \mathbf{C} & * \end{pmatrix},$$

and that the columns of $\mathbf{C} \in \mathbb{Z}^{n \times k}$ form an LLL-reduced basis of $\mathcal{L}^{\perp}(\mathbf{A})$. We thus have, for $i \leq n-k$

$$\|\boldsymbol{b}'^*_i\|^2 \leq \|\boldsymbol{b}'_i\|^2 = \|\boldsymbol{c}_i\|^2 \leq 2^{n-k-1}\lambda^2_{n-k}(\mathcal{L}^{\perp}(\mathbf{A})).$$

Further, for $n-k < j \leq n$, we have

$$\|\boldsymbol{b}'^*_j\|^2 \geq 2^{-k}\|\boldsymbol{b}'^*_{n-k+1}\|^2 \geq 2^{-k}K^2.$$

The choice of K allows to complete the proof. □

We observe again that combining the condition of Proposition 4.2 together with a general purpose bound on $\lambda_{n-k}(\mathcal{L}^{\perp}(\mathbf{A}))$ allows to obtain a sufficient bound on K that can be efficiently derived from $\mathbf{A}$.

Although $\|\boldsymbol{b}^*_{s_i}\|$ is relatively small with respect to K, it can be bounded from below. In fact, we have a more general lower bound:

$$\forall i \leq n, \ \|\boldsymbol{b}^*_i\| \geq 1. \tag{12}$$

This is because that there is a coefficient in $\boldsymbol{b}_i$ which is equal to 1 and 0 for all other b_j's. This lower bound will be helpful in the proof of Theorem 4.3.

4.3 Bounding the number of LLL swaps

Suppose that K is a sufficient large positive integer satisfying (7). Proposition 4.1 guarantees that we can use LLL with $\mathbf{B} = \mathrm{Ext}_K(\mathbf{A})$ as input to compute an LLL-reduced basis for $\mathcal{L}^{\perp}(\mathbf{A})$. We now study the number of LLL swaps performed in this call to the LLL algorithm.

Theorem 4.3. *Let* $\mathbf{A} \in \mathbb{Z}^{n \times k}$ *with a non-zero k-th principal minor, and K an integer satisfying* (7)*. Then, given* $\mathbf{B} = \mathrm{Ext}_K(\mathbf{A})$ *as its input, LLL computes (as a submatrix of the returned basis) an LLL-reduced basis of* $\mathcal{L}^{\perp}(\mathbf{A})$ *after at most* $O(k^3 + k(n-k)(1 + \log\|\mathbf{A}\|))$ *LLL swaps, where* $\|\mathbf{A}\|$ *is the maximum of the Euclidean norm of all rows and columns of the matrix* $\mathbf{A}$.

Proof. From Proposition 4.1, the LLL algorithm allows to obtain a LLL-reduced basis for $\mathcal{L}^{\perp}(\mathbf{A})$. We know from Theorem 3.3 that in order to obtain an upper bound on the number of LLL swaps, it suffices to find an upper bound to $\Pi_k(\mathbf{B})$ and a lower bound on $\Pi_k(\mathbf{B}')$, where $\mathbf{B}'$ is the basis returned by LLL when given $\mathbf{B}$ as

[1] In fact, the resulting matrix gives more information than an LLL-reduced basis of $\mathcal{L}^{\perp}(\mathbf{A})$. For instance, the columns of $\frac{1}{K} \cdot \mathbf{M}$ form a basis of the lattice generated by the rows of $\mathbf{A}$.

Table 1: Upper bounds on the number of LLL swaps for different k (K sufficiently large), $\alpha = \log \|\mathbf{A}\|$.

	Classical analysis (9)	Heuristic (10)	New analysis
$k = 1$	$O(n^2 \log n + n\alpha)$	$O(n^2 + n\alpha)$	$O(n\alpha)$
$k = n/2$	$O(n^3 \log n + n^3\alpha)$	$O(n^3 + n^2\alpha)$	$O(n^3 + n^2\alpha)$
$k = n - 1$	$O(n^2\alpha)$	$O(n^2\alpha)$	$O(n^3 + n\alpha)$

input. From (12) we have

$$\begin{aligned}\Pi_k(\mathbf{B}) &= \sum_{j=1}^{k}(k-j)\log\|\boldsymbol{b}^*_{\ell_j}\| - \sum_{i=1}^{n-k} i \log\|\boldsymbol{b}^*_{s_i}\| + \sum_{i=1}^{n-k} s_i \\ &\leq \sum_{j=1}^{k}(k-j)\log\|\boldsymbol{b}^*_{\ell_j}\| + \sum_{i=1}^{n-k} s_i \\ &\leq \sum_{j=1}^{k}(k-j)\log\|\boldsymbol{b}_{\ell_j}\| + \sum_{i=1}^{n-k} (k+i) \\ &\leq (1 + \log K + \log\|\mathbf{A}\|)\frac{k(k-1)}{2} + \frac{(n-k)(n+k+1)}{2}.\end{aligned}$$

Thanks to Proposition 4.2, we have

$$\begin{aligned}\Pi_k(\mathbf{B}') &= \sum_{j=1}^{k}(k-j)\log\|\boldsymbol{b}'^*_{\ell'_j}\| - \sum_{i=1}^{n-k} i \log\|\boldsymbol{b}'^*_{s'_i}\| + \sum_{i=1}^{n-k} s'_i \\ &= \sum_{j=1}^{k}(k-j)\log\|\boldsymbol{b}'^*_{n-k+j}\| - \sum_{i=1}^{n-k} i \log\|\boldsymbol{b}'^*_i\| + \sum_{i=1}^{n-k} i.\end{aligned}$$

Since the first k coefficients of $\boldsymbol{b}'^*_i$ are 0 (for $i \leq n-k$) and $\mathbf{A}$ is full-rank, we must have $\|\boldsymbol{b}'^*_{n-k+1}\| \geq K$. Further, since $\mathbf{B}'$ is LLL-reduced, combining with (4) we have, for $j \leq k$

$$\|\boldsymbol{b}'^*_{n-k+j}\| \geq 2^{\frac{1-j}{2}}\|\boldsymbol{b}'^*_{n-k+1}\| \geq 2^{\frac{1-j}{2}}K \geq 2^{\frac{1-k}{2}}K.$$

We hence obtain

$$\begin{aligned}\Pi_k(\mathbf{B}') &\geq \left(\log K + \frac{1-k}{2}\right)\sum_{j=1}^{k}(k-j) - \sum_{i=1}^{n-k} i\log\|\boldsymbol{b}'^*_i\| \\ &\quad + \frac{(n-k)(n-k+1)}{2} \\ &\geq \frac{k(k-1)}{2}\left(\log K + \frac{1-k}{2}\right) - (n-k)\sum_{i=1}^{n-k}\log\|\boldsymbol{b}'^*_i\| \\ &\quad + \frac{(n-k)(n-k+1)}{2},\end{aligned}$$

where we used the fact that all $\|\mathbf{b}'^*_i\|$'s are ≥ 1. This is true for the $\|\mathbf{b}^*_i\|$'s and LLL cannot make the minimum Gram-Schmidt norm decrease. Using (3), we obtain:

$$\begin{aligned}\Pi_k(\mathbf{B}') \geq \frac{k(k-1)}{2}\left(\log K + \frac{1-k}{2}\right) - (n-k)k\log\|\mathbf{A}\| \\ + \frac{(n-k)(n-k+1)}{2}.\end{aligned}$$

Finally, using Theorem 3.3, we obtain that the number of LLL swaps is no greater than

$$\frac{\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}')}{\log\left(\frac{2}{\sqrt{3}}\right)} \leq \frac{k(n-\frac{k}{2})\log\|\mathbf{A}\| + k^3 + (n-k)k}{\log\left(\frac{2}{\sqrt{3}}\right)},$$

which is of $O(k^3 + k(n-k)(1 + \log\|\mathbf{A}\|))$. □

In Table 1 we compare favorably ($k = 1, n/2$) the result of Theorem 4.3 to the bounds on the number of swaps using the classical potential (6) and K fixed from the general threshold (9) or the heuristic one (10). We also consider $k = n-1$. However, in the latter case the problem reduces to linear system solving, and different techniques such as those in [18] should be considered.

With the potential function Π of (6), we have

$$\begin{aligned}\Pi(\mathbf{B}) &\leq \log\prod_{i \leq n}\left(K^2\|\mathbf{A}\|^2\right)^{\frac{\min(k,i)}{2}} \\ &\leq \frac{k(2n-k+1)}{2}\log(K\|\mathbf{A}\|).\end{aligned}$$

The bound on the number of LLL swaps obtained using the classical potential is therefore $O(k(n-k/2)(1+\log K+\log\|\mathbf{A}\|)$. While we see from Theorem 4.3 that the actual number of swaps for computing an LLL-reduced basis for $\mathcal{L}^{\perp}(\mathbf{A})$ does not grow with K when K is sufficiently large.

ACKNOWLEDGMENTS

Our thanks go to anonymous referees for helpful comments, which make the presentation of the paper better. Jingwei Chen was partially supported by NNSFC (11501540, 11671377, 11771421) and Youth Innovation Promotion Association, CAS. Damien Stehlé was supported by ERC Starting Grant ERC-2013-StG-335086-LATTAC.

REFERENCES

[1] J. Chen, D. Stehlé, and G. Villard. 2013. A new view on HJLS and PSLQ: Sums and projections of lattices. In *Proceedings of ISSAC'13 (June 26-29, 2013, Boston, MA, USA)*. ACM, 149–156.

[2] Z. Chen and A. Storjohann. 2005. A BLAS based C library for exact linear algebra on integer matrices. In *Proceedings of ISSAC'05 (Beijing, China, July 24–27, 2005)*. ACM, 92–99.

[3] J. Håstad, B. Just, J. C. Lagarias, and C. P. Schnorr. 1989. Polynomial time algorithms for finding integer relations among real numbers. *SIAM Journal of Computing* 18, 5 (1989), 859–881. Erratum: SIAM J. Comput., 43(1), 254–254, 2014.

[4] G. Havas, B. S. Majewski, and K. R. Matthews. 1998. Extended GCD and Hermite normal form algorithms via lattice basis reduction. *Experimental Mathematics* 7, 2 (1998), 125–136.

[5] M. van Hoeij and A. Novocin. 2012. Gradual Sub-lattice Reduction and a New Complexity for Factoring Polynomials. *Algorithmica* 63, 3 (2012), 616–633.

[6] R. Kannan, A. K. Lenstra, and L. Lovász. 1984. Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers. In *Proceedings of STOC'84 (April 30 - May 2, 1984, Washington, DC, USA)*. ACM, 191–200.

[7] A. K. Lenstra, H. W. Lenstra, and L. Lovász. 1982. Factoring polynomials with rational coefficients. *Mathematische Annalen* 261, 4 (1982), 515–534.

[8] A. Neumaier and D. Stehlé. 2016. Faster LLL-type reduction of lattice bases. In *Proceedings of ISSAC'16 (July 20–22, 2016, Waterloo, Ontario, Canada)*. ACM, 373–380.

[9] P. Nguyen and J. Stern. 1997. Merkle-Hellman revisited: A cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *Proceedings of CRYPTO'97 (August 17–21, 1997, Santa Barbara, CA, USA)*. LNCS, Vol. 1294. Springer, 198–212.

[10] P. Q. Nguyen. 1999. *La Géométrie des Nombres en Cryptologie*. Ph.D. Dissertation. Université Paris 7, Paris.

[11] P. Q. Nguyen and B. Vallée (Eds.). 2010. *The LLL Algorithm: Survey and Applications*. Springer, Berlin.

[12] A. Novocin, D. Stehlé, and G. Villard. 2011. An LLL-reduction algorithm with quasi-linear time complexity. In *Proceedings of STOC '11 (June 6–8, 2011, San Jose, USA)*. ACM, 403–412.

[13] M. E. Pohst. 1987. A modification of the LLL reduction algorithm. *Journal of Symbolic Computation* 4, 1 (1987), 123–127.

[14] W. M. Schmidt. 1968. Asymptotic formulae for point lattices of bounded determinant and subspaces of bounded height. *Duke Mathematical Journal* 35, 2 (1968), 327–339.

[15] C. C. Sims (Ed.). 1994. *Computation with Finitely Presented Groups*. Cambridge University Press.

[16] D. Stehlé. 2017. Lattice reduction algorithms. In *Proceedings of ISSAC '17 (July 25-28, 2017, Kaiserslautern, Germany)*. ACM, 11–12.

[17] A. Storjohann. 1996. *Faster algorithms for integer lattice basis reduction*. Technical Report 249. ETH, Department of Computer Science, Zürich, Switzerland.

[18] A. Storjohann. 2005. The shifted number system for fast linear algebra on integer matrices. *J. Complexity* 21, 4 (2005), 609–650.

[19] A. Storjohann and G. Labahn. 1996. Asymptotically fast computation of Hermite normal forms of integer matrices. In *Proceedings of ISSAC'96 (July 24-26, 1996, Zurich, Switzerland)*. ACM, 259–266.