Fast Factoring Integers by SVP Algorithms

Claus Peter Schnorr

Fachbereich Informatik und Mathematik,
Goethe-Universit"at Frankfurt, PSF 111932,
D-60054 Frankfurt am Main, Germany.
schnorr@cs.uni-frankfurt.de

Abstract. To factor an integer $N$ we construct $n$ triples of $p_n$-smooth integers $u, v, |u - vN|$ for the $n$-th prime $p_n$. Denote such triple a fac-relation. We get fac-relations from a nearly shortest vector of the lattice $L(R_{n,f})$ with basis matrix $R_{n,f} \in \mathbb{R}^{(n+1)\times(n+1)}$ where $f : [1,n] \rightarrow [1,n]$ is a permutation of $[1,2,\ldots,n]$ and $(Nf(1),\ldots,Nf(n))$ is the diagonal of $R_{n,f}$. We get an independent fac-relation from an independent permutation $f'$. We find sufficiently short lattice vectors by strong primal-dual reduction of $R_{n,f}$. We factor $N \approx 2^{400}$ by $n = 47$ and $N \approx 2^{800}$ by $n = 95$. Our accelerated strong primal-dual reduction of $[GN08]$ factors integers $N \approx 2^{400}$ and $N \approx 2^{800}$ by $4.2 \cdot 10^6$ and $8.4 \cdot 10^{10}$ arithmetic operations, much faster then the quadratic sieve $QS$ and the number field sieve $NFS$ and using much smaller primes $p_n$. This destroys the RSA cryptosystem.

Keywords. Primal-dual reduction, SVP, fac-relation.

1 Introduction and survy

Section 3 presents factoring algorithms for $N$ that construct independent fac-relations from nearly shortest vectors of the lattice $L(R_{n,f})$ and quite distinct permutations $f : [1,n] \rightarrow [1,n]$. We construct a nearly shortest vector of $L(R_{n,f})$ by primal-dual reduction of the basis $R_{n,f}$ of dim $n$. Differerent random permutations of $[1,n]$ yield independent fac-relations. Our accelerated primal-dual reduction further halves the number of arithmetic operations. Then integers $N \approx 2^{400}$ and $N \approx 2^{800}$ are factored using a much smaller prime basis than the quadratic sieve $QS$ and the number field sieve $NFS$. The main result in section 3 uses from section 5 only the upper bound (5.2) for $M_r^*$ and (5.3). Sections 4 and 5 can be replaced by slide reduction of Gama and Nguyen $[GN08]$ which uses no heuristics.

The enumeration algorithm $\text{Enum}$ of $[SE94]$ for short lattice vectors cuts stages by linear pruning. New $\text{Enum}$ of $[SE94]$ uses the success rate $\delta_t$ of stages based on the GAUSSIAN volume heuristic. It first performs stages with high success rate and stores stages of smaller but still reasonable success rate for later performance. New $\text{Enum}$ finds short vectors much faster than previous algorithms of KANNNAN $[Kn87]$ and FINCKE, PWOHT $[FP85]$ that disregard the success rate of stages. This greatly reduces the number of stages for finding a shortest lattice vector. Section 4 presents time bounds of New $\text{Enum}$ under linear pruning for SVP for arbitrary lattice bases $B = [b_1,\ldots,b_n] \in \mathbb{Z}^{n\times n}$.

2 Lattices

Let $B = [b_1,\ldots,b_n] \in \mathbb{R}^{m\times n}$ be a basis matrix consisting of $n$ linearly independent column vectors $b_1,\ldots,b_n \in \mathbb{R}^m$. They generate the lattice $L(B) = \{Bx : x \in \mathbb{Z}^n\}$ consisting of all integer linear combinations of $b_1,\ldots,b_n$. The dimension of $L$ is $n$, the determinant of $L$ is $\det L = (\det B'B)^{1/2}$ for any basis matrix $B$ and its transpose $B'$. The length of $b \in \mathbb{R}^m$ is $|b| = (b'b)^{1/2}$.

Let $\lambda_1 = \lambda_1(L)$ be the length of the shortest nonzero vector of $L$. The HERMITE constant $\gamma_n$ is the minimal $\gamma$ such that $\lambda_2^2 \leq \gamma(\det L)^{2/n}$ holds for all lattices of dimension $n$.

The basis matrix $B$ has the unique decomposition $B = QR \in \mathbb{R}^{m\times n}$, $R = [r_{i,j}]_{1 \leq i,j \leq n} \in \mathbb{R}^{n\times n}$ where $Q \in \mathbb{R}^{m\times m}$ is isometric (with pairwise orthogonal column vectors of length 1) and $R$ is
upper-triangular with positive diagonal entries \( r_{i,i} \). \( R = \text{GNF}(B) \) is the generic normal form of \( B \). Its Gram-Schmidt coefficients \( \mu_{j,i} = r_{j,i}/r_{i,i} \) are rational for integer matrices \( B \). The orthogonal projection \( b_i^* \) of \( b \in \text{span}(b_1, \ldots, b_{n-1}) \) has length \( r_{i,i} = \|b_i^*\|, r_{i,1} = \|b_i\| \).

LLL-bases. A basis \( B = QR \) is LLL-reduced or an LLL-basis for \( \delta \in (\frac{1}{4}, 1] \) if
1. \( |r_{i,j}|/r_{i,i} \leq \frac{1}{2} \) for all \( j > i \) (size-reduced),
2. \( \delta r_{i,i}^2 \leq r_{i,i+1} + r_{i+1,i+1}^2 \) for \( i = 1, \ldots, n-1 \).

Obviously, LLL-bases satisfy \( r_{i,i}^2 \leq \alpha r_{i,i+1}^2 + r_{i+1,i+1}^2 \) for \( \alpha := 1/(\delta - \frac{1}{2}) \). [LLL82] introduced LLL-bases focusing on \( \delta = 3/4 \) and \( \alpha = 2 \). A famous result of [LLL82] shows that LLL-bases for \( \delta < 1 \) can be computed in polynomial time and that they nicely approximate the successive minima:
3. \( \alpha^{-i+1} \leq \|b_i\|^2 \leq \alpha^{-n-1} \) for \( i = 1, \ldots, n \).
4. \( \|b_i\|^2 \leq \alpha^{-1/n} (\text{det} \, L)^{2/n} \).

A basis \( B = QR \in \mathbb{R}^{m \times n} \) is an HKZ-basis (Hermite, Korkine, Zolotareff) if \( |r_{i,j}|/r_{i,i} \leq \frac{1}{2} \) for all \( j > i \), and if each diagonal entry \( r_{i,i} \) of \( R = [r_{i,j}] \in \mathbb{R}^{n \times n} \) is minimal under all transforms of \( B \) to \( BT, T \in \text{GL}_n(\mathbb{Z}) \) that preserve \( b_1, \ldots, b_{n-1} \).

A basis \( B = QR \in \mathbb{R}^{m \times n} \), \( R = [r_{i,j}]_{1 \leq i,j \leq n} \) is a BKZ-basis for block size \( k \), (or is a BKZ-reduced) if the matrices \( [r_{i,j}]_{h \leq i,j \leq k} \in \mathbb{R}^{k \times k} \) form HKZ-bases for \( h = 1, \ldots, n - k + 1 \), see [SE94].

The efficiency of some algorithms depends on the lattice invariant \( \text{rd}(L) := \lambda_2 \gamma_n^{-1/2} (\text{det} \, L)^{-1/n} \), thus \( \lambda_2^2 = \text{rd}(L)^2 \gamma_n (\text{det} \, L)^{-\frac{2}{n}} \). We call \( \text{rd}(L) \) the relative density of \( L \). Clearly \( 0 < \text{rd}(L) \leq 1 \) holds for all \( L \), and \( \text{rd}(L) = 1 \) if and only if \( L \) has maximal density. Lattices of dim \( n \) of maximal density and \( \gamma_n \) are known for \( n = 1, 2, 3 \), and \( n = 24 \).

3 Fast factoring integers by short vectors of the lattices \( \mathcal{L}(R_{n,f}) \)

Let \( N > 2 \) be an odd integer that is not a prime power and with all prime factors larger than \( p_n \) the \( n \)-th smallest prime. An integer is \( p_n \)-smooth if it has no prime factor larger than \( p_n \). The classical method factors \( N \) by \( n + 1 \) independent pairs of \( p_n \)-smooth integers \( a, \text{gcd}(a,v) \neq 1 \). We call such \( u, v \) \( [u-v \in \mathbb{N}] \) a fac-relation. Our factoring method generates fac-relations with \( p_n \)-smooth \( v \).

The classical method of factoring \( N \). Given \( n + 1 \) fac-relations \( (u_j, v_j) \) we have for \( j = 0 \):

\[
\begin{align*}
\prod_{i=0}^{n} p_i^{e_{i,j}} = u_j \quad & \iff \text{mod } N \quad \text{mod } N \\
\prod_{i=0}^{n} p_i^{e_{i,j}} = u_j \quad & \iff \text{mod } N \quad \text{mod } N \\
\prod_{i=0}^{n} p_i^{e_{i,j}} = u_j \quad & \iff \text{mod } N \quad \text{mod } N \tag{3.1}
\end{align*}
\]

We have \( u_j = v_j \text{ mod } N \) and \( u_j = v_j \text{ mod } N \) if \( u_j = v_j \text{ mod } N \). Hence

\[
\prod_{i=0}^{n} p_i^{e_{i,j}} = u_j \quad \text{mod } N \]

solves \( X^2 - 1 = (X - 1)(X + 1) = \text{mod } N \). If \( X \neq 1 \text{ mod } N \) this yields two non-trivial factors \( \gcd(X \pm 1, N) \notin \{1, N\} \) of \( N \).

The linear equations (3.2) can be solved within \( O(n^3) \) bit operations. We neglect this minor part of the work load of factoring \( N \). Hence \( N \) can be factored by finding about \( n + 1 \) fac-relations. This factoring method goes back to Morrison & Brillhart [MB76] and let to the first factoring algorithm in subexponential time by J. Dixon [DS81].

We generate fac-relations from short vectors of the lattices \( \mathcal{L}(R_{n,f}) \) where \( f : [1, n] \to [1, n] \) is a permutation of \( [1, n] \). We construct short vectors of \( R_{n,f} \in \mathbb{Z}^{n \times (n+1)} \) by strong primal-dual reduction with algorithm 3.2. In order to get distinct fac-relations from distinct permutations \( f : [1, n] \to [1, n] \) it is important that these permutations are quite different, for instance nearly random. The first \( n \) lines of \( R_{n,f} \) have all nonzero entries on the diagonal. \( \text{det } R_{n,f} = N^{n+1} \frac{\phi(n)}{n} \text{ mod } N \).

\[
R_{n,f} = \begin{bmatrix}
Nf(1) & 0 & 0 \\
0 & \ddots & 0 \\
0 & \cdots & Nf(n) \\
N \ln p_1 & \cdots & N \ln p_n \ln N
\end{bmatrix} = \begin{bmatrix}
b_1, \ldots, b_{n+1}
\end{bmatrix},
\]

Here \( \ln = \log_e \) for the Euler number \( e = 2.718281828 \cdots \). Let \( R'_{n,f} = [b_1, \ldots, b_n] \). We identify each vector \( b = \sum_{i=1} b_i b_i \in \mathcal{L}(R_{n,f}) \) with the pair \( (u, v) \) of relative prime and \( p_n \)-smooth integers.
\[ u = \prod_{i > 0} P_i^{e_i}, \quad v = \prod_{i < 0} P_i^{-e_i} \in \mathbb{N} \text{ denoting } \mathbf{b} \sim (u, v). \]

For \( \mathbf{b} \sim (u, v) \) we denote \( \hat{z}_N := N \ln \frac{1}{\varepsilon} \), \( \hat{b}_N - b_{n+1} := N \ln \frac{1}{\varepsilon} \) the last coordinates of \( \mathbf{b} \) and \( \mathbf{b} - b_{n+1} \).

As a factor \( P_i^{e_i} \) of \( \mathbf{uv} \) adds \( e_i \ln p_i \) to \( \ln \mathbf{uv} \) and \( e_i^2 \ln p_i \) to \( |\mathbf{b}|^2 \) we have \( |\mathbf{b}|^2 \geq \ln \mathbf{uv} + \hat{z}_N \) with equality if and only if \( \mathbf{uv} \) is squarefree so that \( e_i \in \{-1, 0, 1\} \) for all \( i \). Similarly

\[ |\mathbf{b} - b_{n+1}|^2 \geq \ln \mathbf{uv} + \hat{z}_N - b_{n+1} \text{ holds for } (u, v) \in \mathcal{L}(\mathbf{R}_n^{(a)}) \quad (3.3) \]

with equality iff \( \mathbf{uv} \) is square-free.

**Lemma 3.1** We have \( \hat{z}_N - b_{n+1} = N \ln \left( \frac{1}{\varepsilon} \right) - N \sum_{i=1}^{\infty} (-x)^i / i \) for \( x = \frac{u-vN}{v} \) and \( |\mathbf{b} - b_{n+1}| = \lambda_1(\mathcal{L}(\mathbf{R}_n^{(a)})) \) then \( |u - vN| < v|\hat{z}_N - b_{n+1}|/(1 - \varepsilon/2) \) holds if either \( vN < u - vN(1 + \varepsilon) \) or \( u < vN \geq u(1 + \varepsilon) \).

**Proof.** We apply the Taylor form \( \ln(1+x) = -\sum_{i=1}^{\infty} (-x)^i / i \) holding for \( x \in [-\frac{1}{2}, \frac{1}{2}] \). Clearly \( \hat{z}_N - b_{n+1} \) lies between the sums \( -N \sum_{i=1}^{\infty} (-x)^i / i \) for \( j = 1, 2 \). If \( vN < u(1 + \varepsilon) \) then

\[ N \frac{u-vN}{v} - N \sum_{i=1}^{\infty} (-x)^i / i < \hat{z}_N - b_{n+1} \]

and this implies

\[ u - vN < vN(1 - \varepsilon) \hat{z}_N - b_{n+1} / (1 - \varepsilon/2) \]

If \( u < vN \) then

\[ N vN - u/v - N \sum_{i=1}^{\infty} (-x)^i / i < |\hat{z}_N - b_{n+1}| \]

and this implies

\[ v - u < v|\hat{z}_N - b_{n+1}| / (1 - \varepsilon/2) \].

Lemma 3.1 shows \( |u - vN| = p_i^a \) for \( z = \ln |u - vN| / \ln p_i \). Hence random \( |u - vN| \) is \( p_i \)-smooth and yields a fac-relation with probability \( \rho(\varepsilon) \). If \( z = \lfloor z \rfloor + \tilde{z} \), with \( 0 < \tilde{z} < 1 \) then \( \rho(z) \approx \rho(\lfloor z \rfloor)(1 + (\frac{\lfloor z \rfloor - 1}{\rho(\lfloor z \rfloor)})^2) \). Note that large \( f(i) \) implies that \( p_i \) is unlikely a factor of \( (u, v) \) of the constructed fac-relation \( u, v, u - vN \). For quite different permutations \( f, f' \) this implies that they generate different fac-relations.

**Algorithm 3.2 for lattice reduction of \( \mathcal{L}(\mathbf{R}_n^{(a)}) \)**

1. LLL-reduce \( \mathbf{R}_n^{(a)} \) for \( \alpha = 1/(\delta - \frac{1}{2}) \), compute \( \mathbf{R} = \text{GNF}(\mathbf{R}_n^{(a)}) \in \mathbb{R}^{(a+1) \times (n+1)} \) in pol. time.
2. Primal-dual reduce \( \mathbf{R} \) to \( \mathbf{RT} \), following [GHKN06] by algorithm 6.3 and iteratively increasing the block size following [AWHTT16]. This yields a vector \( \mathbf{b}_1 \in \mathcal{L}(\mathbf{R}) \) satisfying

\[ |\mathbf{b}_1|^2 \leq \gamma_k(\alpha_{\gamma_k}^2)^{\frac{2}{\gamma_k}}(\text{det}(\mathbf{R}))^{\frac{1}{\gamma_k}}. \]

**Number of arithmetic operations of algorithm 3.2 for \( N \approx 2^{400} \) and factoring \( N \approx 2^{400} \):**

Primal-dual reduce the basis \( \mathbf{R}_{47,f} \in \mathbb{R}^{48 \times 48} \) by alg. 3.2 where \( 48 = h k = 24, h = 48/k = 2 \).

Theorem 6.4 shows that step 2 of algorithm 3.2 performs at most \( 4^{k/2} \log_{\mathfrak{G}}(\alpha) \) \( \log_{\mathfrak{G}^{1/2}}(\alpha) \) iterations. Each iteration HKZ-reduces two blocks \( \mathbf{R}_{47} \), \( \mathbf{R}_i \in \mathbb{R}^{k \times k} \), performing per block \( k^{k+1} \) arithmetic operations according to (5.3), and activates this reduction if a subsequent size-reduction of the columns \( [\mathbf{b}_{48-k+1}, \ldots, \mathbf{b}_{48+k}] \) of the GNF decreases \( \text{det}(\mathbf{R}_i) \) by the factor \( \delta_i^2 \). Step 2 of alg. 3.2 performs at most \( 4 k^{2k} \log_{\mathfrak{G}}(\alpha) k^{k+1} \) \( \leq 1.75 \cdot 10^8 \) arithmetic operations, where \( \log_{\mathfrak{G}}(\alpha) = 1, \delta_B = 3/4, \alpha = 4/3 \). The minor work for LLL-reduction can be neglected. Alg. 3.2 is performed 48 times to find 48 fac-relations. This requires at most \( 48 \cdot 1.75 \cdot 10^8 = 8.4 \cdot 10^9 \) arithmetic operations.

**Number of arithmetic operations of algorithm 3.2 for \( N \approx 2^{1000} \) and factoring \( N \approx 2^{1000} \):**

Primal-dual reduce the basis \( \mathbf{R}_{96,f} \in \mathbb{R}^{96 \times 96} \) by algorithm 6.3 where \( 96 = h k = 24, h = 96/k = 4 \).

By theorem 6.2 this yields a vector \( \mathbf{b}_1 \in \mathcal{L}(\mathbf{R}_{96,f}) \) with \( |\mathbf{b}_1|^2 \leq \gamma_k(\alpha_{\gamma_k}^2)^{\frac{2}{\gamma_k}}(\text{det}(\mathbf{R}_{96,f}))^{\frac{1}{\gamma_k}} \) \( < 0.8408696 \), where \( \text{det}(\mathbf{R}_{96,f}) \approx 2.133441 \cdot 10^{-3} \). Hence \( |\mathbf{b}_1|^2 < 0.917 \). Lemma (3.1) shows

\[ |u - vN| < p_{n_A} \hat{z}_N - b_{n+1}| / (1 - \varepsilon/2) \approx 522.946 = p_{n_A}^{0.007544} \rho(0.007544) = 0.991126 \]

and \( |u - vN| \) is \( p_{n_A} \)-smooth and yields fac-relation with prob. \( \approx 1 \). Then alg. 6.3 performs \( 2k^{2k} \log_{\mathfrak{G}}(\alpha) k^{k+1} \) \( < 1.38 \cdot 10^9 \) arithmetic operations for \( \delta_B = 3/4, \alpha = 4/3 \). Alg. 3.2 is performed 96 times to find 96 fac-relations. This requires at most \( 96 \cdot 1.75 \cdot 10^9 = 1.68 \cdot 10^{11} \) arithmetic operations.
Using strong primal-dual reduction of Gama, Nguyen [GN08] based on the accelerated alg.
6.7 performs about half as many arithmetic operations as alg. 6.3 for primal-dual reduction.
It factors integers \( N \approx 2^{400} \) and \( N \approx 2^{800} \) by \( 4.2 \cdot 10^9 \) and \( 8.4 \cdot 10^{10} \) arithmetic operations.

Using Improved Progressive BKZ Algorithm of [AWHTT16] can still accelerate the time
for finding shortest lattice vectors, in particular for factoring \( N \approx 2^{800} \) and finding a nearly shortest
vector in a lattice with basis \( R_{05, f} \in \mathbb{R}^{56 \times 96} \). It may also be helpful to use the results of [MW16]
for speed up lattice reduction.

Factoring time bounds for quadratic sieve QS and number field sieve NFS: The QS
uses for the factoring of \( N \approx 2^{400} \) that \( p_n \approx 1.2 \sqrt{\ln N \ln \ln N} \approx 3.76 \cdot 10^8 \), see [CP01, section 6.1].
The prime base for NFS is bigger than for QS. The number of arithmetic steps of our factorisation
is quite small compared with QS and NFS factorisation but the bit length of integers is large.
The numbers of arithmetic operations for QS, NFS factorisation of \( N \approx 2^{800} \) in [CP01, section 6.2]:

\[
\begin{align*}
\frac{e^{\sqrt{\ln N \ln \ln N}}}{2} & \approx 1.415 \cdot 10^{17} \text{ for QS} \\
\frac{e^{(64/9)^{1/3}(\ln N)^{1/3}(\ln \ln N)^{2/3}}}{2} & \approx 1.675 \cdot 10^{17} \text{ for NFS.}
\end{align*}
\]

NFS factoring of \( N \approx 2^{800} \) performs \( 2.8126 \times 10^{17} \) arithmetic operations.

4 Efficient enumeration of short lattice vectors

We outline the SVP-algorithm based on the success rate of stages. New Enum improves the algorithm Enum [SE94, SH95]. We recall Enum and present New Enum as a modification that essentially performs all stages of Enum in decreasing order of success rates. This SVP-algorithm New Enum finds a shortest lattice vector fast without enumerating all short lattice vectors.

Let \( \mathbf{b} = [\mathbf{b}_1, ..., \mathbf{b}_n] = \mathbf{Q} \mathbf{R} \in \mathbb{R}^{n \times n}, \mathbf{R} = [r_{ij}]_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n} \), be the given basis of \( \mathcal{L} = \mathcal{L}(\mathbf{Q}). \) Let \( \pi_t : \text{span}(\mathbf{b}_1, ..., \mathbf{b}_n) \to \text{span}(\mathbf{b}_1, ..., \mathbf{b}_{t-1})^\perp = \text{span}(\mathbf{b}_t, ..., \mathbf{b}_n) \) for \( t = 1, ..., n \) denote the orthogonal projections and let \( \mathcal{L}_t = \mathcal{L}(\mathbf{b}_1, ..., \mathbf{b}_{t-1}). \)

The success rate of stages. At stage \( \mathbf{u} = (u_t, ..., u_n) \) of Enum for SVP of \( \mathcal{L} \) a vector \( \mathbf{b} = \sum_{i=1}^{n} u_i \mathbf{b}_i \in \mathcal{L} \) is given such that \( \|\pi_t(\mathbf{b})\|_2^2 \leq \lambda_t^2. \) (When \( \lambda_t^2 \) is unknown we use instead some \( A > \lambda_t^2 \) ) Stage \( t \) calls the substages \( (u_{t-1}, ..., u_n) \) such that \( \|\pi_{t-1}(\sum_{i=t}^{n} u_i \mathbf{b}_i)\|_2^2 \leq \lambda_t^2. \) We have

\[
\|\sum_{i=1}^{n} u_i \mathbf{b}_i\|_2^2 = \|\zeta_t + \sum_{i=1}^{t} u_i \mathbf{b}_i\|_2^2 + \|\pi_t(\mathbf{b})\|_2^2,
\]

where \( \zeta_t := \mathbf{b} - \pi_t(\mathbf{b}) \in \mathcal{L}_t \) is \( \mathbf{b} \)'s orthogonal projection in span \( \mathcal{L}_t. \) Stage \( t \) and its substages enumerate the intersection \( \mathcal{B}_{t-1}(\zeta_t, \mathcal{G}_t) \cap \mathcal{L}_t \) of the sphere \( \mathcal{B}_{t-1}(\zeta_t, \mathcal{G}_t) \subset \text{span} \mathcal{L}_t \) with radius \( \mathcal{G}_t := (\lambda_t^2 - \|\pi_t(\mathbf{b})\|_2^2)^{1/2} \) and center \( \zeta_t. \) The Gaussian volume heuristics estimates for \( t = 1, ..., n \) the expected size \( |\mathcal{B}_{t-1}(0, \mathcal{G}_t) \cap (\zeta_t + \mathcal{L}_t)| \) to be the success rate

\[
\beta_t(\mathbf{u}) = \text{det} \mathcal{L}_t \text{ vol } \mathcal{B}_{t-1}(0, \mathcal{G}_t) / \text{ det } \mathcal{L}_t \tag{4.1}
\]

standing for the probability that there is an extension \( (u_1, ..., u_n) \) of \( \mathbf{u} = (u_t, ..., u_n) \) such that
\( \|\sum_{i=1}^{n} u_i \mathbf{b}_i\|_2^2 \leq \lambda_t. \) Here \( \text{vol } \mathcal{B}_{t-1}(0, \mathcal{G}_t) = V_{t-1}^1 \mathcal{G}_t^{t-1}, \) \( V_{t-1} = \frac{\pi^{t-1}}{\sqrt{\pi}} \approx \frac{1}{\lambda_t^{t-1}} \) is the volume of the unit sphere of dimension \( t-1 \) and det \( \mathcal{L}_t = \lambda_{t-1} \cdots \lambda_{t-1,n-1}. \) If \( \zeta_t \in \text{span } \mathcal{L}_t \) is uniformly distributed the expected size of this intersection satisfies \( \mathbb{E}[\#(\mathcal{B}_{t-1}(0, \mathcal{G}_t) \cap (\zeta_t + \mathcal{L}_t))] = \beta_t(\mathbf{u}). \) This holds because \( 1/ \text{ det } \mathcal{L}_t \) is the number of lattice points of \( \mathcal{L}_t \) per volume in span \( \mathcal{L}_t. \) We do not simply cut \( \mathbf{u} \) due to a small \( \beta_t \) because there might be a vector in \( \mathcal{L}_t \) very close to \( \zeta_t. \)

The success rate \( \beta_t \) has been used in [SH95] to speed up Enum by cutting stages of very small success rate. New Enum first performs all stages with sufficiently large \( \beta_t \) giving priority to small \( t \) and collects during this process the unperformed stages in the list \( L. \) For instance it first performs all stages with \( \beta_t \geq 2^{-s} \log_2(t). \) Thereafter New Enum increases \( s \) to \( s + 1. \) So far our experiments simply perform all stages with \( \beta_t \geq 2^{-s}. \) If \( \lambda_t^2 \) is unknown we can compute \( \gamma_t \beta_t \) replacing \( \lambda_t^2 \) by the upper bound \( A = \frac{1.244}{s^2} \text{ det } (\mathbf{Q} \mathbf{R})^2 \geq \lambda_t^2 \) which holds since \( \gamma_t \leq \frac{1.244}{s^2} n \approx 0.10211 n \) holds for \( n \geq n_0 \) by a computer proof of Kabatiants, Levenstein [KaLe78]. Dabei ist \( e = 2.7182818284 \cdots \)
Euler’s number und \( \pi = 3.141592654 \cdots \).  

4
Outline of New Enum

INPUT BKZ-basis $B = QR$, $R = [r_{i,j}] \in \mathbb{R}^{n \times n}$ of block size 32, $A, s = \lceil \log n \rceil \log_2 n$

OUTPUT a sequence of $b \in \mathcal{L}(B)$ of decreasing length terminating with $\|b\| = \lambda_1$.

1. $L := \emptyset$.

2. Let New Enum perform all stages $u_i = (u_{1},\ldots,u_n)$ with $\beta_i(u) \geq 2^{-s} \log t$:

Upon entry of stage $(u_1,\ldots,u_n)$ compute $\beta_i(u)$. If $\beta_i(u) < 2^{-s} \log t$ then store $(u_1,\ldots,u_n)$ in the list $L$ of delayed stages. Otherwise perform stage $(u_1,\ldots,u_n)$. If $t = 1$ some $b \in \mathcal{L} \setminus \emptyset$ of length $\|b\|^2 \leq A$ has been found, give out $b$, we can then decrease $A := \|b\|^2 - 1$ if $R^R \in \mathbb{Z}^{n \times n}$.

3. $s := s + 1$, IF $L \neq \emptyset$ THEN perform all stages $u_i \in L$ with $\beta_i(u) \geq 2^{-s} \log t$.

Running in linear space. If instead of storing the list $L$ we restart New Enum in step 3 on level $s + 1$ then New Enum runs in linear space and its running time increases at most by a factor $n$.

Practical optimization. New Enum computes $R, \beta_i, V_i, \varrho_i, c_i$ in floating point and $b$, $\|b\|^2$ in exact arithmetic. The final output $b$ has length $\|b\| = \lambda_1$, but this is only known when the more expensive final search does not find a vector shorter than the final $b$.

Reason of efficiency. For short vectors $b = \sum_{i=1}^{n} u_i b_i \in \mathcal{L} \setminus \emptyset$ the stages $u = (u_1,\ldots,u_n)$ have large success rate $\beta_i(u)$. On average $\|\pi_1(b)\|^2 \approx \frac{1}{2} A^2$ holds for a random $b \in R$ of length $\lambda_1$. Therefore $\beta_i^2 = A - \|\pi_1(b)\|^2$ and $\beta_i$ are large. New Enum tends to output very short lattice vectors first.

New Enum is particularly fast for small $\lambda_1$. The size of its search space approximates $\lambda_1^3 V_n$, and is by Prop. 4.1 heuristically polynomial if $rd(L) = o(n^{-1/2})$. Having found $b'$ New Enum proves $\|b'\| = \lambda_1$ in exponential time by a complete exhaustive enumeration.

New Enum for SVP

INPUT BKZ-basis $B = QR$, $R = [r_{i,j}] \in \mathbb{R}^{n \times n}$, $A \geq \lambda_1^2$, $s_{\max}$

OUTPUT a sequence of $b \in \mathcal{L}(B)$ such that $\|b\|$ decreases to $\lambda_1$.

1. $L := \emptyset$, $t := t_{\max} := 1$, FOR $i = 1,\ldots,n$ DO $c_i := u_i := y_i := 0$, $y_i := u_i := 1$, $s := 5$

2. WHILE $t \leq n$ #perform stage $u_i := (u_{1},\ldots,u_n, y_i, c_i, \varrho_i, \varsigma_i, \beta_i, A)$:

$[c_i := c_i + 1 + (u_i + y_i)^2 r_{i,i},$

IF $c_i \geq A$ THEN GO TO 2.1,

$y_i := (A - c_i)^{1/2}$, $\beta_i := V_i y_i^{-1} (r_{1,1} \cdots r_{t-1,t-1})$,

IF $t = 1$ THEN $b := \sum_{i=1}^{n} u_i b_i$,

IF $\|b\|^2 < A$ THEN $A := \|b\|^2$, output $(b, s, A)$, GO TO 2.1 $]$ IF $\beta_i \geq 2^{-s}$ THEN $[t := t - 1$, $y_i := \sum_{i=1}^{t} u_i r_{i,i}$, $u_i := -y_i$, $\varrho_i := \text{sign}(u_i + y_i)$, $\nu_i := 1$, GO TO 2 $]$ ELSE IF $\beta_i \geq 2^{-s}$ THEN store $u_i := (u_{1},\ldots,u_n, y_i, c_i, \varrho_i, \varsigma_i, \beta_i, A)$ in $L$. 2.1. $t := t + 1$, $t_{\max} := \max(t, t_{\max})$.

IF $t = t_{\max}$ THEN $u_i := u_i + 1$, $y_i := 1$, $s := 0$

ELSE $u_i := -[y_i + \nu_i/2](-1)^{\nu_i} \varsigma_i$, $\nu_i := \nu_i + 1$ ]

3. perform all stages $u_i = (u_{1},\ldots,u_n, y_i, c_i, \varrho_i, \varsigma_i, \beta_i, A) \in L$ with $\beta_i \geq 2^{-s}$, IF steps 2, 3 did not decrease $A$ for the current $s$ THEN terminate.

4. $s := s + 1$, IF $s > s_{\max}$ THEN restart with a larger $s_{\max}$.

Notation. We use the following function $c_i : \mathbb{Z}^{n-t+1} \to \mathbb{R}:

[c_i(u_{1},\ldots,u_n) = \|\pi_1(\sum_{i=1}^{n} u_i b_i))\|^2 = \sum_{i=1}^{n} (\sum_{j=1}^{n} u_j r_{i,j})^2.

Hence

c_i(u_{1},\ldots,u_n) = (\sum_{i=1}^{n} u_i r_{i,i})^2 + c_{i+1}(u_{1},\ldots,u_n).

Given $u_{i+1},\ldots,u_n$ Enum takes for $u_i$ the integers that minimize $|u_i + y_i|$ for $y_i := \sum_{i=1}^{n} u_i r_{i,i}$ in order of increasing distance to $-y_i$ adding to the initial $u_i := -[y_i]$ iteratively $[\nu_i/2](-1)^{\nu_i} \varsigma_i$ where $\varsigma_i := \text{sign}(u_i + y_i) \in \{\pm 1\}$ and $\nu_i$ numbers the iterations starting with $\nu_i = 0, 1, 2, \ldots$:

$-[y_i], -[y_i] - \varsigma_i, -[y_i] + \varsigma_i, -[y_i] - 2\varsigma_i, -[y_i] + 2\varsigma_i, \ldots, -[y_i] + [\nu_i/2](-1)^{\nu_i} \varsigma_i, \ldots,$
where \( \text{sign}(0) := 1 \) and \([r]\) denotes a nearest integer to \( r \in \mathbb{R} \). The iteration does not decrease \( |u_t + y_t| \) and \( c_t(u_t, \ldots, u_n) \), it does not increase \( g_t \) and \( \beta_t \). Enum performs the stages \((u_t, \ldots, u_n)\) for fixed \( u_{t+1}, \ldots, u_n \) in order of increasing \( c_t(u_t, \ldots, u_n) \) and decreasing success rate \( \beta_t \). \( \beta_t \) extends this priority to stages of distinct \( t, t' \) taking into account the size of two spheres of distinct dimensions \( n - t, n - t' \). The center \( g_t = b - \pi_t(b) = \sum_{i=t}^n u_i(b_i - \pi_i(b_i)) \in \text{span}(L) \) changes continously within New Enum which improves Enum When step 3 performs stages \( u_{t'} \in L \) the current \( A \) can be smaller than the \( A \) of \( u_t \) and this can make the stored \( \beta_{t'} \) of \( u_{t'} \) smaller than \( 2^{-x} \) so that \( u_{t'} \) will not be performed but must be stored in \( L \) with the adjusted smaller values \( A, \beta_{t'} \). The stored stages \( u_{t'} \) with \( \beta_{t'} \geq 2^{-x} \) should be performed in a succession giving priority to large success rates and small \( t' \).

**Time for solving SVP for \( L(B) \).** New Enum performs for each \( s = 5, 6, \ldots, s_{\max} \) only stages \( u_t \) with success rate \( \beta_t \geq 2^{-x} \). Let \#\( t, s, A \) denote the number of performed stages with \( t, s, A \). If \( \beta_t \) is a reliable probability then New Enum performs on average at most \( 2^{s} \) stages with success rate \( \beta_t \geq 2^{-x} \). New Enum performs for each stage of step 2 on average at most \( 2(n - t)(1 + o(1)) \) arithmetical steps for computing \( y_t \), which are added up to \( \sum_{t=1}^n 2(n - t)(1 + o(1)) \approx n(n + 1)(1 + o(1)) \) arithmetical steps and it performs \( O(n) \) arithmetical steps for testing that \( \ln \beta_t \geq -s \ln 2 \) for \( t = 1, \ldots, n \), using \( \beta_t \approx V_{t-1} b_{t-1} / \det L \) assuming that \( \ln(2EN), \ln n, \ln(1 + x) \) for \( x = 1, \ldots, n \) are given for free.

If the initial basis \( B \in \mathbb{R}^{n \times n} \) is a BKZ-basis with block size \( k \) then \( \|b_t\| \leq L \lambda_t^{1/k} \). As New Enum performs stages with high success rates first then each decrease of \( A \) will on average halve \( A/\lambda_t^2 \) so that there are at most \( \log_2(A/\lambda_t^2) \) iterations of step 2 that decrease the initial \( A \) of step 1. So after the initial reduction of \( B \) New Enum solves SVP for \( s_{\max} \), with error probability \( o(1) \) and performs on average at most \( O(n^22^s_{\max}) \) arithmetical steps for each \( A \). Hence SVP is solved by

\[
2^s_{\max} (n^2 + O(n))2^{s_{\max} - 1} \log_2 \gamma \quad \text{arithmetical steps.} \quad (4.2)
\]

5 New Enum for SVP with linear pruning

The heuristics of linear pruning gives weaker results but is easier to justify than handling the success rate \( \beta_t \) as a probability function. Proposition 5.1 bounds under linear pruning the time to find \( b' \in L(B) \) with \( \|b'\| = \lambda_1 \). It shows that SVP is polynomial time if \( rd(L) \) is sufficiently small. Note that finding an unproved shortest vector \( b' \) is easier than proving \( \|b'\| = \lambda_1 \). New Enum finds an unproved shortest lattice vector \( b' \) in polynomial time under the following conditions and assumptions:

- the given lattice basis \( B = [b_1, \ldots, b_n] \) and the relative density \( rd(L) \) of \( L(B) \) satisfy 
  \[ rd(L) \leq (\sqrt{\frac{n}{\lambda_1^2}})^{\frac{1}{2}} \text{, i.e., both } b_1 \text{ and } rd(L) \text{ are sufficiently small.} \]
- GSA: The basis \( B = QR, R = [r_{1,i}]_{1\leq i\leq n} \) satisfies \( r_{2,i}^2 / r_{1,i-1}^2 = q \) for \( 2 \leq i \leq n \) for some \( q > 0 \).
- SA: There is a vector \( b' \in L(B) \) such that \( \|b'\| = \lambda_1 \) and \( \|\pi_t(b')\| \leq \frac{n+i+1}{n} \lambda_t^2 \) for \( t = 1, \ldots, n \).

( Later we will use a similar assumption CA for CVP. )

- the vol. heap. is close: \( M_t := \#B_{n-t+1}(0, \varrho_t) \cap \pi_t(L) \approx \frac{1}{\det \pi_t(L)} \frac{\varrho_t^{n-t+2}}{\varrho_t^{n-t+1}} \) for \( \varrho_t = \frac{n+i+1}{n} \lambda_t^2 \).

**Remarks.** 1. If GSA holds with \( q \geq 1 \) the basis \( B \) satisfies \( \|b_i\| \leq \frac{1}{\sqrt{1 + 3 L_1 / \lambda_1}} \) for all \( i \) and \( \|b_1\| = \lambda_1 \). Therefore, \( q < 1 \) unless \( \|b_1\| = \lambda_1 \). GSA means that the reduction of the basis is "locally uniform" i.e., the \( r_{1,i}^2 \) form a geometric series. It is easier to work with the idealized property that all \( r_{1,i}/r_{1,i-1} \) are equal. In practice \( r_{1,i}/r_{1,i-1} \) slightly increases on the average with \( i \). [BL05] studies "near equality v." GSA has been used in [S03, S06, G08, S10, N10] and in the security analysis of NTRU in [H07, HHHW09].

2. The assumption SA is a fact proven in the full paper of [GNR10]:

\[ \Pr[\|\pi_t(b')\|^2 \leq \frac{n+i+1}{n} \lambda_t^2] \]

for random \( b' \in R \text{ span}(L) \) with \( \|b'\| = \lambda_1 \).

**Linear pruning** means to cut off all stages \((u_t, \ldots, u_n)\) that satisfy \( \|\pi_t(\sum_{i=t}^n u_i b_i)\|^2 > \frac{n+i+1}{n} \lambda_t^2 \).
Linear pruning is impractical because it does not provide any information on SVP in case of failure. We use linear pruning only as a theoretical model for easy analysis. We have implemented SVP via New Enum and we will show in section 5 that stages \((u_1, \ldots, u_n)\) that are cut by linear pruning have extremely low success probability so they will not be performed by New Enum.

3. Errors of the volume heuristics. The minimal and maximal values of \(\#_n := \#(B_n(\zeta, \varrho_n) \cap \mathcal{L})\), and similar for \#_1 := \#(B_1(\zeta, \varrho_1) \cap \pi_{n-1}(\mathcal{L}))\), are for fixed \(n, \varrho_n\) very close for large radius \(\varrho_n\), but can differ considerably for small \(\varrho_n\) since \#_n can change a lot with the actual center \(\zeta_n\) of the sphere. For small \(\varrho_n\) the minimum of \#_n can be very small and then the average value for random center \(\zeta_n\) is closer to the maximum of \#_n. For more details see the theorems and Table 1 of [MO90].

As New Enum works with average values for \#_n, \#_1 its success rate \(\beta_n\) frequently overestimates the success rate for the actual \(\zeta_n\). A cut of the smallest (resp. closest) lattice vector by New Enum in case that it underestimates \#_1 can nearly be excluded if stages are only cut for very small \(\beta_n\).

Our time bounds must be multiplied by the work load per stage, a modest polynomial factor covering the steps performed at stage \((u_1, \ldots, u_n)\) of Enum before going to a subsequent stage.

Proposition 5.1 Let the basis \(B = QR, R \in \mathbb{R}^{n \times n}\) of \(\mathcal{L}\) satisfy \(rd(\mathcal{L}) \leq \left(\frac{n}{\log n}\right)^{\frac{1}{2}}\) and GSA and let \(\mathcal{L}\) have a shortest lattice vector \(b'\) that satisfies SA. Then Enum with linear pruning finds such \(b'\) under the volume heuristic in polynomial time.

Proof. For simplicity we assume that \(\lambda_1\) is known. Pruning all stages \((u_1, \ldots, u_n)\) that satisfy \(\|\pi_t(\sum_{i=1}^n u_i b_i)\| > \frac{n+1}{\lambda_1}\) does not cut off any shortest lattice vector \(b'\) that satisfies SA.

The volume heuristics approximates the number \(M_t^\pi\) of performed stages \((u_1, \ldots, u_n)\) to

\[
M_t^\pi := \#B_{n-t+1}(0, \varrho_t) \cap \pi_t(\mathcal{L}) \approx \left(\frac{n-t+1}{\lambda_1}\right)^{n-t+1} \left(\frac{\pi_{n-t}}{\lambda_1}\right)
\]

Here \(\approx\) uses Stirling’s approximation \(\pi_n = \pi n/\sqrt{2\pi n}\). Obviously \(\|b'_i\| = r_{1,1}q_{1,1}\) holds by GSA and thus

\[
\left(\frac{\pi n}{\lambda_1}\right)^{n-t+1} = \frac{\pi_{n-t}}{\lambda_1}.
\]

For \(t = 1\) this yields \(\frac{n-1}{\lambda_1} = (\det \mathcal{L})/\lambda_{1,1} = 1\). Combining (5.1) with this equation and \(\gamma_n < \frac{n-1}{\lambda_1}\) which holds for \(n > n_0\), we get

\[
M_t^\pi \leq \left(\frac{\pi n}{\lambda_1}\right)^{n-t+1} \left(\frac{\pi_{n-t}}{\lambda_1}\right)^{\frac{n-t+1}{\lambda_1}}
\]

Evaluating this upper bound for \(rd(\mathcal{L}) \leq \left(\frac{\pi n}{\lambda_1}\right)^{\frac{1}{2}}\) yields

\[
M_t^\pi \leq \left(\frac{\pi_{n-t}}{\lambda_1}\right)^{n-t+1} \left(\frac{\pi n}{\lambda_1}\right)^{\frac{n-t+1}{\lambda_1}}.
\]

This approximate upper bound has for \(t \leq n\) its maximum 1 at \(t = n\). This proves Prop. 5.1. \(\Box\)

Note that (5.2) only assumes the volume heuristic and GSA, but no upper bound on \(rd(\mathcal{L})\).

SVP-time bound for \(rd(\mathcal{L}) \leq 1\) under linear pruning. (5.2) proves for \(rd(\mathcal{L}) \leq 1\) that

\[
M_t^\pi \leq \left(\frac{\pi n}{\lambda_1}\right)^{n-t+1} \left(\frac{\pi n}{\lambda_1}\right)^{\frac{n-t+1}{\lambda_1}}
\]

The exponent \(n - \frac{(t-1)(t-2)}{2} - n + t - 1\) is maximal for \(t = n/2 + 1\) with maximal value \(\frac{1}{4}\). This proves for \(r_{1,1}/\lambda_{1,1} = \pi^{\Omega(1)} / \sqrt{\pi n}\) the heuristic SVP time bound

\[
O(n)\left(\frac{\pi n}{\lambda_1}\right)^{\frac{1}{4}} \approx 2^{\frac{n^2}{2^{\frac{n}{4}}}} = n^{4\pi + 1}\cdot\frac{1}{\lambda_1}
\]

This beats under heuristics the proven SVP time bound \(n^{\frac{1}{2} + o(n)}\) of HANROT, STEHLE [HS07] which holds for a quasi-HKZ-basis \(B\) satisfying \(\|b_1\| \leq 2\|b_2\|\) and having a HKZ-basis \(\pi_2(B)\). In fact \(\frac{1}{4} \approx 0.159 > 0.125 = \frac{1}{2}\). The SVP-algorithm of Prop.1 can use fast BKZ for preprocessing and
works even for $\|b_1\| \gg 2 \lambda$ – see the attack on $\gamma$-unique SVP – whereas [HS07] requires quasi-HKZ-reduction for preprocessing. This eduction already guarantees $\|b_1\| \leq 2 \lambda$ and performs the main SVP work during preprocessing. Our SVP time bound $n^{n/8+o(n)}$ only assumes $\|b_1\| \leq n^{o(1)} \sqrt{\pi \lambda}$.

**Theorem 5.4** Given a lattice basis $B \in \mathbb{Z}^{n \times n}$ satisfying GSA and $\|b_1\| \leq \sqrt{\pi} n^h \lambda_1$ for some $b \geq 0$, New Enum solves SVP and proves to have found a solution in time $2^{O(n)}(n^{1+o(1)})$.

Theorem 5.4 is proven in [S10], it does not assume SA and the vol. heuristic. Recall from remark 4 that $n^{1+o(1)}$ is a lower bound for SVP under GSA. For $b = o(1)$ Thm. 5.4 shows the SVP-time bound $n^{1+o(1)}$ which beats $n^{1/2+o(n)}$ from Hanrot, Stehlé [HS07]. Cor. 1 translates Thm. 1 from SVP to CVP, it shows that the corresponding CVP-algorithm solves many important CVP-problems in simple exponential time $2^{O(n)}$ and linear space.

[HS07] proves the time bound $n^{n/2+o(n)}$ for solving CVP by Kannan’s CVP-algorithm [Ka87]. Minimizing $\|b\|$ for $b \in L \setminus \{0\}$ and minimizing $\|t - b\|$ for $b \in L$ require nearly the same work if $\|t - L\| \approx \lambda$. In fact the proof of Theorem 1 yields

ENM with linear pruning solves SVP of $L$ of dim $L = n$ by (5.4) in worst case heuristic time $n^{n/8+o(1)}$. New Enum solves SVP much faster. Short vectors are found much faster if available stages with large success rate are always performed first and if stages with very small success rate are cut.

6 Primal-dual reduction

**Definition 6.1** Let $B = QR \in \mathbb{R}^{m \times h}$ be a lattice basis with $R = \text{GNF}(B) = [r_{ij}]_{1 \leq i,j \leq h}$ with blocks $R_\ell = [r_\ell,j]_{1 \leq i,j \leq h}, \ell = 1,\ldots,h$ of size $k$. Then $B$ is a *primal-dual basis* if

1. it is LLL-basis with HKZ-bases $R_\ell, \ell = 1,\ldots,h$.
2. $\max_{\ell} r_{\ell,\ell}^2 \leq \alpha r_{\ell+1,\ell+1}^2$ for $\ell = 1,\ldots,h-1$, where $r_{\ell,\ell}^2$ of GNF$(R_\ell T)$

is maximized over all $T \in GL_k(\mathbb{Z})$ for the $\alpha = 1/(\delta - 1/2)$ of LLL-reduction.

**Theorem 6.2** [GHKN06]

Every primal-dual basis $B = QR \in \mathbb{R}^{m \times h}$ of lattice $L$ satisfies $\|b_1\|^2 \leq \gamma_h (\alpha r_{\ell}^2)^{1/2} (\det R)^{1/2}$.

**Proof.** Def. 6.1 shows for $R = [r_{ij}]_{1 \leq i,j \leq h}$ and $r_{\ell,\ell}^2$ of GNF$(R_\ell T)$ that $\max_{\ell} r_{\ell,\ell}^2 \leq \alpha r_{\ell+1,\ell+1}^2$. The inverse matrix $U_\ell = \left(\begin{smallmatrix} 1 & 0 \\ \vdots & \vdots \end{smallmatrix}\right) \in \mathbb{Z}^{k \times k}$ yields for the lower triangular matrix $R_\ell^{-1} \in \mathbb{R}^{k \times k}$ the upper triangular matrix $R_\ell^* = U_\ell R_\ell^{-1} U_\ell^\top$, where $R_\ell^\top$ is the transpose of the matrix $R_\ell$. The Hermite inequality $\lambda_1^2(L(R_\ell^*)) \leq \gamma_h D_\ell^{-1/k}$ for $L(R_\ell^*)$, $D_\ell = (\det R_\ell)^2$, and HKZ-reduction of $R_\ell^*$ imply

$D_\ell^{1/k} \leq \gamma_h \max_{\ell} r_{\ell,\ell}^2 = \gamma_h / \lambda_1^2(L(R_\ell^*))$.

The HKZ-basis $R_{\ell+1}$ satisfies

$\lambda_1^2(L(R_{\ell+1})) = r_{\ell+1,\ell+1}^2 \leq \gamma_h D_{\ell+1}^{1/k}$.

The combination of these two inequalities and Def. 6.1, part 2, yields

$D_\ell^{1/k} \leq \gamma_h \max_{\ell} r_{\ell,\ell}^2 \leq \alpha \gamma_h D_{\ell+1}^{1/k} \leq \alpha \gamma_h r_{\ell+1,\ell+1}^2 \leq r_{\ell+1,\ell+1}^2$.

(6.1)

For the HKZ-basis $R_1$ follows by induction over $\ell$ that

$\|b_1\|^2 \leq \gamma_h D_{\ell}^{1/k} \leq \gamma_h (\alpha r_{\ell}^2)^{1/2} D_{\ell+1}^{1/k}$ für $\ell = 0,\ldots,h-1$.

The $h$-th root of the product of these $h$ inequalities proves the claim, as $\sum_{\ell=1}^{h-1} \ell = \frac{h-1}{2}$, $\prod_{\ell=1}^{h} D_\ell = (\det R)^2$. \qed
Algorithm 6.3: for Primal-dual reduction

**INPUT** LLL-basis \( B = \{ b_1, ..., b_{n+1} \} = QR \in \mathbb{Z}^{m \times h k}, n + 1 = h k, \) for \( \alpha = 1/(\delta - \frac{1}{4}) \), with blocks \( R_1, ..., R_h \subseteq R \) of size \( k \), \([ b_{k-1} k+1, ..., b_k ] = B_1 = B \). \( R_\ell, \ell = 1 \)

1. HKZ-reduce the block \( R_{\ell+1} \) to \( R_{\ell+1}T_k \) with \( T_k \in \text{GL}_k(\mathbb{Z}) \). \( B_{\ell+1} := B_{\ell+1}T_k \)

2. HKZ-reduce \( R_\ell^* \) to \( R_\ell^*T_\ell \), with \( T_\ell \in \text{GL}_k(\mathbb{Z}) \). \( B_\ell := B_\ell(U_\ell T_\ell^{-1} U_\ell, \) size-reduce \( B_\ell, B_{\ell+1}, \) compute \( R_{\ell, \ell+1} = \text{GNF}([B_\ell, B_{\ell+1}]) \). LLL-reduce \( R_{\ell, \ell+1} \) with \( \delta, \alpha \) to \( R_{\ell, \ell+1}T_{2k} \)

3. IF step 2 exchanged the columns \( k \) and \( k+1 \) of \( R_{\ell, \ell+1} \)

   THEN \( [B_\ell, B_{\ell+1}] := [B_\ell, B_{\ell+1}]T_2k, \) size-reduce \( [B_\ell, B_{\ell+1}], \) \( \ell := \max(\ell - 1, 1) \) ELSE \( \ell := \ell + 1 \)

4. IF \( \ell < h \) THEN GO TO 1

**OUTPUT** primal-dual basis \( B \)

**Comments on Alg. 6.3.** Step 2 maximizes the last diagonal entry \( r_k,k \) of \( \text{GNF}(R_\ell^*T_\ell) \in \mathbb{R}^{k \times k} \) for \( T \in \text{GL}_k(\mathbb{Z}) \). The reversal matrix \( U_\ell = [1, -1] \in \mathbb{Z}^{k \times k} \) yields for \( R_\ell^*T_\ell \in \mathbb{R}^{k \times k} \), the transpose of \( R_\ell^* \), an upper triangular matrix \( R_\ell^* = U_\ell R_\ell^* U_\ell \). \( R_\ell^* \) is the transpose of \( R_\ell, \) HKZ-reduction of \( R_\ell^* \) to \( R_\ell^*T_\ell \) with \( T_\ell \in \text{GL}_k(\mathbb{Z}) \) minimizes the first diagonal entry \( r_{1,1} \) of \( \text{GNF}(R_\ell^*T_\ell) \) and maximizes \( 1/r_{1,1}, \) the last diagonal entry of \( \text{GNF}(R_\ell^*U_\ell^T U_\ell) \). The transformation \( T_\ell \) of \( R_\ell^* \) yields the transformation \( U_\ell T_\ell^{-1} U_\ell \) for \( R_\ell \) and \( B_\ell \).

The LLL-reduction of \( R_{\ell, \ell+1} \) in step 2 either starts by exchanging the columns \( k \) and \( k+1 \) of \( R_{\ell, \ell+1} \) or \( R_{\ell, \ell+1} \) is already LLL-reduced, because \( r_k,k \) of \( R_\ell \) is maximal and \( r_{k+1,k+1} \) of \( R_{\ell, \ell+1} \) is minimal. The primal-dual output \( B \) of Alg. 6.3 is of the form \( BT_1 \) for the input \( B \) of Alg.6.3 and \( T_1 \in \text{GL}_n(\mathbb{Z}) \).

**Theorem 6.4 :** Alg. 6.3 performs at most \( \frac{n^2k^2}{12} \log_{1/\delta} \alpha \) iterations before arriving at \( D_B \leq 1 \).

**Proof.** We replace the Lovász invarante \( D \) by the following invarante \( D_B \) where \( D_\ell = (\det R_\ell)^2 : D_B := \prod_{i=0}^{\ell-1} (D_\ell/D_{\ell+1})^{h^2/4 - (h/2 - \ell)^2} \).

The exponent \( h^2/4 - (h/2 - \ell)^2 \) is maximal for \( \ell = h/2 \), is zero for \( \ell = 0 \) and \( \ell = h \) and is symmetric to \( \ell = h/2 \). For the LLL-input basis we have \( D_\ell \leq \alpha^2 D_{\ell+1} \) and therefore her \( D_B \)-value \( D_B^{\text{imp}} \) satisfies \( D_B^{\text{imp}} \leq \alpha^{h^2/3} \) for \( s = \frac{h}{2} \sum_{\ell=1}^{h-1} h^2/4 - (h/2 - \ell)^2 \).

The well known sum \( \bar{s} := \frac{h}{2} \sum_{\ell=1}^{h-1} \ell^2 = h(h-1)(h-1)/3 \) yields \( \sum_{\ell=1}^{h-1} (h/2 - \ell)^2 = -h^2(h-1)/4 + \bar{s} = h(h-1)(h-2)/12 \) and therefore \( s = (h+1)(h-1)(h-1)/6 = (h^3 - h)/6 \).

Hence we have \( D_B^{\text{imp}} \leq \alpha^{h^2(h^3 - h)/6} \). An active step 3 changes from \( D_B \) only the factor \( \prod_{i=0}^{\ell-1} (D_\ell/D_{\ell+1})^{(h/2 - \ell)^2} \).

Every iteration with \( D_B^{\text{imp}} \leq \delta^2 D_B^{\text{old}} \) decreases as shown \( D_B \) to \( D_B^{\text{imp}} \leq \delta^2 D_B^{\text{old}} \). For the number \( \# \text{It} \) of iterations until arriving at \( D_B \leq 1 \) we get from \( s = \sum_{\ell=1}^{h-1} (h^2/4 - (h/2 - \ell)^2) = (h^3 - h)/6 \) that \( \# \text{It} \leq \frac{1}{2} \log_{1/\delta} D_B^{\text{imp}} \leq \frac{1}{2} \log_{1/\delta} \alpha^{h^2/3} \leq \frac{h^2}{12} \log_{1/\delta} \alpha = \frac{n^2k^2}{12} \log_{1/\delta} \alpha. \)

Part 2 of def. 6.1. has been hightened by GAMA, NGUYEN [GN08] to \( 2^+ \max_{R_\ell^*T_\ell} r_{k+1,k+1} \leq 1 + e \) for \( \ell = 1, ..., h - 1, \) \( 0 < e \approx 0 \). [GN08 slide-reduction]

Let \( R_o^* := [r_{i,j} k-1 k+2 \leq i, j \leq k+1] \in \mathbb{R}^{k \times k} \) denote the segment one unit to the right of \( R_\ell \). \( \max_{R_o^*T_\ell^*} r_{k+1,k+1} \) marks the maximum over \( r_{k+1,k+1}^2 \) of \([r_{i,j}] \) = \( \text{GNF}(R_o^*T) \) over all \( T \in \text{GL}_k(\mathbb{Z}) \), \( B_o := [b_{k-1 k+2} \cdots b_{k+1}] \)
Definition 6.5 A size-reduced basis $B = QR \in \mathbb{R}^{m \times h}$ is strong primal-dual if $R_1, \ldots, R_h$ are HKZ-bases satisfying $2^+$. 

Theorem 6.6 [GN08] A strong primal-dual basis $B = QR \in \mathbb{R}^{m \times h}$ with $0 \approx \varepsilon > 0$ in $2^+$ of the lattice $L$ satisfies $||b_1|| < ((1 + \varepsilon) \gamma_k)^{1/k_h} (\det L)^{1/k_h}$. 

Proof. Hermite showed for an HKZ-basis $R_h$ that $r_{k+1, k'}) \leq \gamma_k D_k$. The dual of this inequality shows for $D'_k := (\det R'_e)^2$ that $\max r_{k+1, k'} \geq D'_k/\gamma_k$ for $T \in GL_1(Z)$. 

For $\ell = \ell_{\max}$ this shows that $2^+$ implies for every primal-dual basis that 

$$D'_k \leq (1 + \varepsilon)^{2k^2} r_{k+1, k'}^{2k^2}.$$  

Combination of (6.2) with $r_{k+1, k'} \leq \gamma_k D_k$ and $D'_k/r_{k+1, k'} = D_k/r_{k+1, k'}^{2k^2}$ implies 

$$r_{k+1, k'} \leq ((1 + \varepsilon) \gamma_k)^{2k^2} r_{k+1, k'}^{2k^2}$$ 

for $\ell = \ell_{\max}$ and $\ell = h - 1$. 

Therefore we get for $\ell_{\max}$ and also for all $\ell = 1, \ldots, h - 1$ that $D_k \leq (1 + \varepsilon)^{2k^2} D_{\ell+1}$. 

The HKZ-basis $R_1$ this implies for $\ell = 1, \ldots, h$ that 

$$||b_1||^2 \leq \gamma_k D_1^{1/k} \leq \gamma_k ((1 + \varepsilon) \gamma_k)^{2k^2} D_1^{1/k}.$$ 

The product of these $h$ inequalities and $\sum_{\ell=1}^h (\ell - 1) = \frac{h(h-1)}{2}$ yields 

$$||b_1||^{2h} \leq \gamma_h ((1 + \varepsilon) \gamma_h)^{kk/(h-1)} (\det L)^2/h.$$ 

This proves the claim $||b_1||^2 \leq \gamma_h ((1 + \varepsilon) \gamma_k)^{1/2} (\det L)^2/k_h < (1 + \varepsilon) \gamma_k^{1/2} (\det L)^{2/k_h}$. 

Algorithm 6.7 : Accelerated, strong primal-dual reduction 

Input LLL-basis $B = QR \in \mathbb{Z}^{m \times n}$, $n = h$, $0 \approx \varepsilon > 0$. 

1. Choose $\ell$, $1 \leq \ell < n$ where $D_\ell/D_{\ell+1}$ is maximal. 

2. HKZ-reduce $R_{\ell+1}$ to $R_{\ell+1}T$ with $T \in GL_1(Z)$. $B_{\ell+1} := B_{\ell+1}T$, size-reduce $B_{\ell+1}$, renew $R_{\ell+1}$. HKZ-reduce (R')* to (R')*T, with $T' \in GL_1(Z)$, $[r'_{k+1, k}]_{2 \leq i, j \leq 1+k} := GNF(R'_i T'_{-1} U_k)$ for the reversal matrix $U_k = \left[ \begin{array}{c} 1 \\ \varepsilon \end{array} \right] \in \mathbb{Z}^{k \times k}$. 

3. IF $r'_{k+1, k'} \geq (1 + \varepsilon) r_{k+1, k'}$ THEN $B'_\ell := B'_\ell T'_{-1} U_k$, size-reduce $B'_\ell$, renew $R_{\ell+1}$, GOTO 1 

Output strong primal-dual basis $B$. 

Theorem 6.8 Alg. 6.7 performs at most $\frac{\varepsilon^2}{2} \log_{1+\varepsilon} \alpha$ iteration until arriving at $D_B \leq 1$. 

Proof. An active step 3 implies $D'^{\text{new}}_\ell \leq D'^{\text{old}}_\ell/(1 + \varepsilon)^2$ and thus $D'^{\text{new}}_B \leq D'^{\text{old}}_B/(1 + \varepsilon)^4$. The input-LLL-basis $B$ satisfies $D'^{\text{old}}_B \leq \alpha^{2} \frac{1}{(1 + \varepsilon)^2}$. Hence $#I \leq \frac{\varepsilon^{2}}{2} \log_{1+\varepsilon} \alpha$ until $D_B \leq 1$. 

10
[GN08] replaces $(1 + \varepsilon)$ in $2^{\ell}$ by $\sqrt{1 + \varepsilon}$ for all $\ell \leq h - 1$; [GN08] nearly proves Theorem 6.6 for slide reduced bases. Since we require $2^{\ell}$ only for $\ell_{\text{max}}$ then Alg. 6.7 for strong primal-dual reduction performs at most 2 HKZ-reductions of dim. $k$ per iteration and therefore is clearly faster than strong primal-dual reduction of [GN08]. Alg. 6.7 for accelerated, strong primal-dual reduction performs about half as many arithmetic operations as alg. 6.3 for primal-dual reduction.

References


