Fast Factoring Integers by SVP Algorithms

Claus Peter Schnorr

Fachbereich Informatik und Mathematik,
Goethe-Universität 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 $\mathcal{L}(\mathbf{R}_{n,f})$ with basis matrix $\mathbf{R}_{n,f} \in \mathbb{R}^{(n+1)\times(n+1)}$ where $f : [1, n] \to [1, n]$ is a permutation of $[1, 2, ..., n]$ and $(Nf(1), ..., Nf(n))$ is the diagonal of $\mathbf{R}_{n,f}$. We get an independent fac-relation from an independent permutation f'. We find sufficiently short lattice vectors by strong primal-dual reduction of $\mathbf{R}_{n,f}$. 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 $4.2 \cdot 10^9$ 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 surview

Section 3 presents factoring algorithms for N that construct independent fac-relations from nearly shortest vectors of the lattice $\mathcal{L}(\mathbf{R}_{n,f})$ and quite distinct permutations $f : [1, n] \to [1, n]$. We construct a nearly shortest vector of $\mathcal{L}(\mathbf{R}_{n,f})$ by primal-dual reduction of the basis $\mathbf{R}_{n,f}$ of dim $n+1 \in 24 \cdot \mathbb{N}$ and n = 47 for $N \approx 2^{400}$, and n = 95 for $N \approx 2^{800}$ using blocks of size 24. Alg. 6.7 accelerates strong primal-dual reduction of [GN08]. This yields a nearly shortest vector of $\mathcal{L}(\mathbf{R}_{n,f})$. Lemma 5.1 shows that this reduction yields a **fac-relation**. The determinant of $\mathbf{R}_{n,f}$ is the same for all f. Differant 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 $\mathcal{M}_t^\varrho$ 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 Enum of [SE94] for short lattice vectors cuts stages by linear pruning. New Enum of [SE94] uses the success rate β_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 Enum finds short vectors much faster than previous algorithms of Kannan [Ka87] and Fincke, Pohst [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 Enum under *linear pruning* for **SVP** for arbitrary lattice bases $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \mathbb{Z}^{n\times n}$.

2 Lattices

Let $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n] \in \mathbb{R}^{m\times n}$ be a basis matrix consisting of n linearly independent column vectors $\mathbf{b}_1, ..., \mathbf{b}_n \in \mathbb{R}^m$. They generate the lattice $\mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} \,|\, \mathbf{x} \in \mathbb{Z}^n\}$ consisting of all integer linear combinations of $\mathbf{b}_1, ..., \mathbf{b}_n$. The *dimension* of $\mathcal{L}$ is n, the *determinant* of $\mathcal{L}$ is $\det \mathcal{L} = (\det \mathbf{B}^t\mathbf{B})^{1/2}$ for any basis matrix $\mathbf{B}$ and its transpose $\mathbf{B}^t$. The *length* of $\mathbf{b} \in \mathbb{R}^m$ is $\|\mathbf{b}\| = (\mathbf{b}^t\mathbf{b})^{1/2}$.

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

The basis matrix $\mathbf{B}$ has the unique decomposition $\mathbf{B} = \mathbf{Q}\mathbf{R} \in \mathbb{R}^{m\times n}$, $\mathbf{R} = [r_{i,j}]_{1\leq i,j\leq n} \in \mathbb{R}^{n\times n}$ where $\mathbf{Q} \in \mathbb{R}^{m\times n}$ is isometric (with pairwise orthogonal column vectors of length 1) and $\mathbf{R}$ is upper-triangular with positive diagonal entries $r_{i,i}$. $\mathbf{R} = \mathrm{GNF}(\mathbf{B})$ is the *generic normal form* of $\mathbf{B}$. Its Gram-Schmidt coefficients $\mu_{j,i} = r_{i,j}/r_{i,i}$ are rational for integer matrices $\mathbf{B}$. The orthogonal projection $\mathbf{b}_i^*$ of $\mathbf{b}_i \in \mathrm{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1})^\perp$ has length $r_{i,i} = \|\mathbf{b}_i^*\|$, $r_{1,1} = \|\mathbf{b}_1\|$.

LLL-*bases.* A basis $\mathbf{B} = \mathbf{QR}$ is LLL-*reduced* or an LLL-*basis* for $\delta \in (\frac{1}{4}, 1]$ if

1. $|r_{i,j}|/r_{i,i} \le \frac{1}{2}$ for all $j > i$ (size-reduced), **2.** $\delta r_{i,i}^2 \le r_{i,i+1}^2 + r_{i+1,i+1}^2$ for $i = 1, ..., n-1$.

Obviously, LLL-bases satisfy $r_{i,i}^2 \le \alpha\, r_{i+1,i+1}^2$ for $\alpha := 1/(\delta - \frac{1}{4})$. [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. $\alpha^{-i+1} \le \|\mathbf{b}_i\|^2 \lambda_i^{-2} \le \alpha^{n-1}$ for $i = 1, ..., n$, **4.** $\|\mathbf{b}_1\|^2 \le \alpha^{\frac{n-1}{2}} (\det \mathcal{L})^{2/n}$.

A basis $\mathbf{B} = \mathbf{QR} \in \mathbb{R}^{m\times n}$ is an HKZ-*basis* (Hermite, Korkine, Zolotareff) if $|r_{i,j}|/r_{i,i} \le \frac{1}{2}$ for all $j > i$, and if each diagonal entry $r_{i,i}$ of $\mathbf{R} = [r_{i,j}] \in \mathbb{R}^{n\times n}$ is minimal under all transforms of $\mathbf{B}$ to $\mathbf{BT}$, $\mathbf{T} \in \mathrm{GL}_n(\mathbb{Z})$ that preserve $\mathbf{b}_1, ..., \mathbf{b}_{i-1}$.

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

The efficiency of some algorithms depends on the lattice invariant $rd(\mathcal{L}) := \lambda_1 \gamma_n^{-1/2} (\det \mathcal{L})^{-1/n}$, thus $\lambda_1^2 = rd(\mathcal{L})^2 \gamma_n (\det(\mathcal{L}))^{\frac{2}{n}}$. We call $rd(\mathcal{L})$ the *relative density* of $\mathcal{L}$. Clearly $0 < rd(\mathcal{L}) \le 1$ holds for all $\mathcal{L}$, and $rd(\mathcal{L}) = 1$ if and only if $\mathcal{L}$ has maximal density. Lattices of dim n of maximal density and γ_n are known for $n = 1, ..., 8$ and $n = 24$. .

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

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 $u, |u - vN|$. We call such $u, |u - vN|$ a fac-relation. Our factoring method generates fac-relations wih p_n-smooth v.

The classical method of factoring N. Given $n+1$ fac-relations (u_j, v_j) we have for $p_0 := -1$

$$u_j = \textstyle\prod_{i=1}^n p_i^{e_{i,j}}, \qquad u_j - v_j N = \prod_{i=0}^n p_i^{e'_{i,j}} \qquad \text{with } e_{i,j}, e'_{i,j} \in \mathbb{N}. \tag{3.1}$$

We have $(u_j - v_j N)/u_j \equiv 1 \mod N$ since $(u_j - v_j N) = u_j \mod N$. Hence $\prod_{i=0}^n {p_i}^{e_{i,j} - e'_{i,j}} \equiv 1 \mod N$. Any solution $t_1, ..., t_{n+1} \in \{0, 1\}$ of the equations

$$\textstyle\sum_{j=1}^{n+1} t_j (e_{i,j} - e'_{i,j}) \equiv 0 \mod 2 \quad \text{for } i = 0, ..., n \tag{3.2}$$

solves $X^2 - 1 = (X-1)(X+1) = 0 \mod N$ by $X = \prod_{i=0}^n p_i^{\frac{1}{2}\sum_{j=1}^{n+1} t_j (e_{i,j} - e'_{i,j})} \mod N$. If $X \ne \pm 1 \mod N$ 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 [MB75] and let to the first factoring algorithm in subexponetial time by J. Dixon [D81].

We generate fac-relations from short vectors of the lattices $\mathcal{L}(\mathbf{R}_{n,f})$ where $f : [1, n] \to [1, n]$ is a permutation of $[1, n]$. We construct short vectors of $\mathbf{R}_{n,f} \in \mathbb{R}^{(n+1)\times(n+1)}$ 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 differant, for instance nearly random. The first n lines of $\mathbf{R}_{n,f}$ have all nonzero entries on the diagonal. $\det \mathbf{R}_{n,f} = N^{n+1} \frac{n(n+1)}{2} \ln N$.

$$\mathbf{R}_{n,f} = \begin{bmatrix} Nf(1) & & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & & Nf(n) & 0 \\ N\ln p_1 & \cdots & N\ln p_n & N\ln N \end{bmatrix} = [\mathbf{b}_1, ..., \mathbf{b}_{n+1}],$$

Here $\ln = \log_e$ for the Euler number $e = 2.7182818284\cdots$. Let $\mathbf{R}'_{n,f} = [\mathbf{b}_1, ..., \mathbf{b}_n]$. We identify each vector $\mathbf{b} = \sum_{i=1}^n e_i \mathbf{b}_i \in \mathcal{L}(\mathbf{R}'_{n,f})$ with the pair (u, v) of relative prime and p_n-smooth integers

$$u = \prod_{e_i>0} p_i^{e_i}, \;\; v = \prod_{e_i<0} p_i^{-e_i} \in \mathbb{N} \quad \text{denoting } \mathbf{b} \sim (u,v).$$

For $\mathbf{b} \sim (u,v)$ we denote $\hat{z}_{\mathbf{b}} := N \ln \frac{u}{v}$, $\hat{z}_{\mathbf{b-b_{n+1}}} := N \ln \frac{u}{vN}$ the last coordinates of $\mathbf{b}$ and $\mathbf{b} - \mathbf{b}_{n+1}$. As a factor $p_i^{\pm e_i}$ of uv adds $\pm e_i \ln p_i$ to $\ln uv$ and $e_i^2 \ln p_i$ to $\|\mathbf{b}\|^2$ we have $\|\mathbf{b}\|^2 \geq \ln uv + \hat{z}_{\mathbf{b}}^2$ with equality if and only if uv is squarefree so that $e_i \in \{-1,0,1\}$ for all i. Similarly

$$\|\mathbf{b} - \mathbf{b_{n+1}}\|^2 \geq \ln uv + \hat{z}^2_{\mathbf{b}-\mathbf{b}_{n+1}} \text{ holds for } (u,v) \sim \mathbf{b} \in \mathcal{L}(\mathbf{R}'_{n,q}) \qquad (3.3)$$
$$\text{with equality iff } uv \text{ is square-free.}$$

Lemma 3.1 We have $\hat{z}_{\mathbf{b}-\mathbf{b}_{n+1}} = N \ln(\frac{u}{vN}) = -N \sum_{i=1}^{\infty} (-x)^i/i$ for $x = \frac{u-vN}{vN}, (u,v) \sim \mathbf{b} \in \mathcal{L}(\mathbf{R}'_{n,f})$. Let $x \in [-\frac{1}{2}, \frac{1}{2}]$ and $||\mathbf{b} - \mathbf{b}_{n+1}|| = \lambda_1(\mathcal{L}(\mathbf{R}_{n,f}))$ then $|u - vN| < v|\hat{z}_{\mathbf{b}-\mathbf{b}_{n+1}}|/(1 - \varepsilon/2)$ holds if either $vN < u < vN(1+\varepsilon)$ or $u < vN < u(1+\varepsilon)$.

Proof. We apply the Taylor form $\ln(1+x) = -\sum_{i=1}^{\infty}(-x)^i/i$ holding for $x \in [-\frac{1}{2}, 1]$. Clearly $\hat{z}_{\mathbf{b}-\mathbf{b}_{n+1}}$ lies between the sums $-N \sum_{i=1}^{j} (-x)^i/i$ for $j = 1, 2$.
If $vN < u < (1+\varepsilon)vN$ then $\quad N\frac{u-vN}{vN}(1 - \frac{u-vN}{2vN}) < \hat{z}_{\mathbf{b}-\mathbf{b}_{n+1}}$
and this implies $\quad u - vN < vN^{1-c}\hat{z}_{\mathbf{b}-\mathbf{N}_c}/(1-\varepsilon/2)$.
If $u < vN < (1+\varepsilon)u$ then $\quad N\frac{vN-u}{vN}(1 - \frac{vN-u}{2vN}) < |\hat{z}_{\mathbf{b}-\mathbf{b}_{n+1}}|$
and this implies $\quad v - u < v|\hat{z}_{\mathbf{b}-\mathbf{b}_{n+1}}/(1-\varepsilon/2)$. □

Lemma 3.1 shows $|u - vN| = p_n^z$ for $z = \ln|u - vN|/\ln p_n$. Hence random $|u - vN|$ is p_n-smooth and yields a fac-relation with probability $\rho(z)$. If $z = \lfloor z \rfloor + \tilde{z}$, with $0 < \tilde{z} < 1$ then $\rho(z) \approx \rho(\lfloor z \rfloor)\left(\frac{\rho(\lfloor z \rfloor + 1)}{\rho(\lfloor z \rfloor)}\right)^{\tilde{z}}$. 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 $\mathcal{L}(\mathbf{R}_{n,f})$

1. LLL-reduce $\mathbb{R}_{n,f}$ for $\alpha = 1/(\delta - \frac{1}{4})$, compute $\mathbf{R} = \text{GNF}(\mathbf{R}_{n,f}) \in \mathbb{R}^{(n+1)\times(n+1)}$ in pol. time.

2. Primal-dual reduce $\mathbf{R}$ to $\mathbf{RT}_1$ following [GHKN06] by algorithm 6.3 and iteratively incriesing the block size following [AWHTT16]. This yields a vector $\mathbf{b}_1 \in \mathcal{L}(\mathbf{R})$ satisfying
$$||\mathbf{b}_1||^2 \leq \gamma_k(\alpha\gamma_k^2)^{\frac{h-1}{2}}(\det \mathbf{R})^{\frac{2}{n+1}}.$$

Number of arithmetic operations of algorithm 3.2 for $N \approx 2^{400}$ and factoring $N \approx 2^{400}$: Primal-dual reduce the basis $\mathbf{R}_{47,f} \in \mathbb{R}^{48\times48}$ by alg. 3.2 where $48 = hk, k = 24, h = 48/k = 2$. Theorem 6.4 shows that step 2 of algorithm 3.2 performs at most $\frac{48^2h}{12} \cdot log_{1/\delta_B}(\alpha)$ iterations. Each iteration HKZ-reduces two blocks $\mathbf{R}_{\ell+1}, \mathbf{R}_\ell^* \in \mathbb{R}^{k\times k}$, performing per block $k^{k/8+1.1}$ arithmetic operations according to (5.3), and activates this reduction if a subsequent size-reduction of the columns $[\mathbf{b}_{\ell k-k+1}, ..., \mathbf{b}_{\ell k+k}]$ of the **GNF** decreases $\det(\mathbf{R}_\ell)$ by the factor δ_B^2. Step 2 of alg. 3.2 performs at most $\frac{48^2h}{12}\ log_{1/\delta_B}(\alpha)\ k^{k/8+1.1} \lesssim 1.75 \cdot 10^8$ arithmetic operations, where $log_{1/\delta_B}(\alpha) = 1$, $\delta_B = 3/4$, $\alpha = 4/3$. 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 $N \approx 2^{800}$ and factoring $N \approx 2^{800}$: Primal-dual reduce the basis $\mathbf{R}_{95,f} \in \mathbb{R}^{96\times96}$ by algorithm 6.3 where $96 = hk$, $k = 24$, $h = 96/k = 4$. By theorem 6.2 this yields a vector $\mathbf{b}_1 \in \mathcal{L}(\mathbf{R}_{95,f})$ with $||\mathbf{b}_1||^2 \leq \gamma_k(\alpha\gamma_k^2)^{\frac{h-1}{2}} \det(\mathcal{L}(\mathbf{R}_{95,f}))^{\frac{2}{96}}$ < 0.8408696, where $\det(\mathcal{L}(\mathbf{R}_{95,f}))^{\frac{2}{96}} = 2.133441 \cdot 10^{-3}$. Hence $||\mathbf{b}_1|| < 0.917$. Lemma (3.1) shows for $\mathbf{b} - \mathbf{b}_{n+1} \sim (u,v)$, $v \lesssim p_n = p_{95} = 499$, $\varepsilon = \frac{1}{4}$ that
$$|u - vN| < p_n|\hat{z}_{\mathbf{b}-\mathbf{b}_{n+1}}|/(1-\varepsilon/2) \lesssim 522.946 = p_n^{0.007544}$$
$\rho(0.007544) = \rho(2)^{0.007544} = 0.991126$ and $|u - vN|$ is p_n-smooth and yields fac-relation with prob. ≈ 1. Then alg. 6.3 performs $\frac{96^24}{12} log_{1/\delta_B}(\alpha) k^{k/8+1.1} < 1.38 \cdot 10^9$ 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 $\mathbf{R}_{95,f} \in \mathbb{R}^{96\times 96}$. It may also be helpful to use the results of [MW16] to 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 $p_n \approx e^{1/2\sqrt{\ln N \cdot \ln\ln N}} \approx 3.76 \cdot 10^8$, 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 $N \approx 2^{400}$ in [CP01, section 6.2] :

$$e^{\sqrt{\ln N \ln\ln N}} \approx 1.415 \cdot 10^{17} \text{ for } \mathbf{QS}$$
$$e^{(64/9)^{1/3}(\ln N)^{1/3}(\ln\ln N)^{2/3}} \approx 1.675 \cdot 10^{17} \text{ for } \mathbf{NFS}.$$

NFS factoring of $N \approx 2^{800}$ performs 2.8126×10^{23} 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 of [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 $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n] = \mathbf{QR} \in \mathbb{R}^{m\times n}$, $\mathbf{R} = [r_{i,j}]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$, be the given basis of $\mathcal{L} = \mathcal{L}(\mathbf{B})$. Let $\pi_t : \text{span}(\mathbf{b}_1, ..., \mathbf{b}_n) \to \text{span}(\mathbf{b}_1, ..., \mathbf{b}_{t-1})^\perp = \text{span}(\mathbf{b}_\mathrm{t}^*, ..., \mathbf{b}_\mathrm{n}^*)$ 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 $\mathbf{b} = \sum_{i=t}^n u_i\mathbf{b}_i \in \mathcal{L}$ is given such that $\|\pi_t(\mathbf{b})\|^2 \le \lambda_1^2$. (When λ_1^2 is unknown we use instead some $A > \lambda_1^2$.) Stage $\mathbf{u}$ calls the substages $(u_{t-1}, ..., u_n)$ such that $\|\pi_{t-1}(\sum_{i=t-1}^n u_i\mathbf{b}_i)\|^2 \le \lambda_1^2$. We have $\|\sum_{i=1}^n u_i\mathbf{b}_i\|^2 = \|\zeta_t + \sum_{i=1}^{t-1} u_i\mathbf{b}_i\|^2 + \|\pi_t(\mathbf{b})\|^2$, where $\zeta_t := \mathbf{b} - \pi_t(\mathbf{b}) \in \text{span}\,\mathcal{L}_t$ is $\mathbf{b}$'s orthogonal projection in $\text{span}\,\mathcal{L}_t$. Stage $\mathbf{u}$ and its substages enumerate the intersection $\mathcal{B}_{t-1}(\zeta_t, \varrho_t) \cap \mathcal{L}_t$ of the sphere $\mathcal{B}_{t-1}(\zeta_t, \varrho_t) \subset \text{span}\,\mathcal{L}_t$ with radius $\varrho_t := (\lambda_1^2 - \|\pi_t(\mathbf{b})\|^2)^{1/2}$ and center ζ_t. The Gaussian volume heuristics estimates for $t = 1, ..., n$ the expected size $|\mathcal{B}_{t-1}(\mathbf{0}, \varrho_t) \cap (\zeta_t + \mathcal{L}_t)|$ to be the success rate
$$\beta_t(\mathbf{u}) =_{def} \text{vol}\,\mathcal{B}_{t-1}(\mathbf{0}, \varrho_t) \,/\, \det \mathcal{L}_t \tag{4.1}$$
standing for the probability that there is an extension $(u_1, ..., u_n)$ of $\mathbf{u} = (u_t, ..., u_n)$ such that $\|(\sum_{i=1}^n u_i\mathbf{b}_i)\| \le \lambda_1$. Here $\text{vol}\,\mathcal{B}_{t-1}(\mathbf{0}, \varrho_t) = V_{t-1}\varrho_t^{t-1}$, $V_{t-1} = \pi^{\frac{t-1}{2}}/(\frac{t-1}{2})! \approx (\frac{2e\pi}{t-1})^{\frac{t-1}{2}}/\sqrt{\pi(t-1)}$ is the volume of the unit sphere of dimension $t-1$ and $\det \mathcal{L}_t = r_{1,1}\cdots r_{t-1,t-1}$. If $\zeta_t \in \text{span}\,\mathcal{L}_t$ is uniformly distributed the expected size of this intersection satisfies $\mathrm{E}_{\zeta_t}[\,\#\big(|\mathcal{B}_{t-1}(\mathbf{0}, \varrho_t) \cap (\zeta_t + \mathcal{L}_t)\big)\,] = \beta_t(\mathbf{u})$. This holds because $1/\det \mathcal{L}_t$ is the number of lattice points of $\mathcal{L}_t$ per volume in $\text{span}\,\mathcal{L}_t$. We do not simply cut $\mathbf{u}_t$ due to a small β_t because there might be a vector in $\mathcal{L}_t$ very close to ζ_t.

The success rate β_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 β_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 $\beta_t \ge 2^{-s}\log_2(t)$. Thereafter New Enum increases s to $s+1$. So far our experiments simply perform all stages with $\beta_t \ge 2^{-s}$. If λ_1^2 is unknown we can compute ϱ_t, β_t replacing λ_1^2 by the upper bound $A = \frac{1.744}{2e\pi} n \det(\mathbf{B}^t\mathbf{B})^{\frac{2}{n}} \ge \lambda_1^2$ which holds since $\gamma_n \le \frac{1.744}{2e\pi} n \approx 0.10211\,n$ holds for $n \ge n_0$ by a computer proof of Kabatiansky, Levenstein [KaLe78]. Dabei ist $e = 2.7182818284\cdots$ Euler's number und $\pi = 3.141592654\cdots$.

```
Outline of  New Enum
INPUT BKZ-basis $\mathbf{B} = \mathbf{QR}$, $\mathbf{R} = [r_{i,j}] \in \mathbb{R}^{n\times n}$ of block size 32, $A$, $s = \lg n = \log_2 n$
OUTPUT  a sequence of $\mathbf{b} \in \mathcal{L}(\mathbf{B})$ of decreasing length terminating with $\|\mathbf{b}\| = \lambda_1$.
1.  $L := \emptyset$.
2.  Let New Enum perform all stages $\mathbf{u}_t = (u_t, ..., u_n)$ with $\beta_t(\mathbf{u}) \geq 2^{-s} \lg t$:
    Upon entry of stage $(u_t, ..., u_n)$ compute $\beta_t(\mathbf{u}_t)$. If $\beta_t(\mathbf{u}_t) < 2^{-s} \lg t$ then
    store $(u_t, ..., u_n)$ in the list $L$ of delayed stages. Otherwise perform stage
    $(u_t, ..., u_n)$, set $t := t - 1$, $u_t := -\lceil \sum_{i=t+1}^n u_i r_{t,i}/r_{t,t} \rfloor$ and go to stage
    $(u_t, ..., u_n)$. If for $t = 1$ some $\mathbf{b} \in \mathcal{L} \setminus \mathbf{0}$ of length $\|\mathbf{b}\|^2 \leq A$ has been found,
    give out $\mathbf{b}$, we can then decrease $A := \|\mathbf{b}\|^2 - 1$ if $\mathbf{R}^t\mathbf{R} \in \mathbb{Z}^{n\times n}$.
3.  $s := s + 1$,  IF  $L \neq \emptyset$  THEN  perform all stages $\mathbf{u}_t \in L$ with $\beta_t(\mathbf{u}_t) \geq 2^{-s} \log_2 t$.
```

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 $\mathbf{R}$, $\beta_t, V_t, \varrho_t, c_t$ in floating point and $\mathbf{b}$, $\|\mathbf{b}\|^2$ in exact arithmetic. The final output $\mathbf{b}$ has length $\|\mathbf{b}\| = \lambda_1$, but this is only known when the more expensive final search does not find a vector shorter than the final $\mathbf{b}$.

Reason of efficiency. For short vectors $\mathbf{b} = \sum_{i=1}^n u_i \mathbf{b}_i \in \mathcal{L} \setminus \mathbf{0}$ the stages $\mathbf{u} = (u_t, ..., u_n)$ have large success rate $\beta_t(\mathbf{u})$. On average $\|\pi_t(\mathbf{b})\|^2 \approx \frac{n-t+1}{n}\lambda_1^2$ holds for a random $\mathbf{b} \in_R \mathcal{B}_n(\mathbf{0}, \lambda)$ of length λ_1. Therefore $\varrho_t^2 = A - \|\pi_t(\mathbf{b})\|^2$ and $\beta_t(\mathbf{u})$ are large. New Enum tends to output very short lattice vectors first.

New Enum is particularly fast for small λ_1. The size of its search space approximates $\lambda_1^n V_n$, and is by Prop. 4.1 heuristically polynomial if $rd(\mathcal{L}) = o(n^{-1/4})$. Having found $\mathbf{b}'$ New Enum proves $\|\mathbf{b}'\| = \lambda_1$ in exponential time by a complete exhaustive enumeration.

```
New Enum for SVP
INPUT  BKZ-basis $\mathbf{B} = \mathbf{QR}$, $\mathbf{R} = [r_{i,j}] \in \mathbb{R}^{n\times n}$, $A \geq \lambda_1^2$, $s_{max}$
OUTPUT  a sequence of $\mathbf{b} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{b}\|$ decreases to $\lambda_1$.
1. $L := \emptyset$, $t := t_{max} := 1$, FOR  $i = 1, ..., n$ DO $c_i := u_i := y_i := 0$, $\nu_1 := u_1 := 1$, $s := 5$
   $c_1 := r_{1,1}^2$,                    ($c_t = c_t(u_t, ..., u_n)$ always holds for the current $t$)
2.  WHILE  $t \leq n$  #perform stage $\mathbf{u}_t := (u_t, ..., u_n, y_t, c_t, \nu_t, \varsigma_t, \beta_t, A)$:
          [[ $c_t := c_{t+1} + (u_t + y_t)^2 r_{t,t}^2$,
          IF  $c_t \geq A$  THEN  GO TO  2.1,
          $\varrho_t := (A - c_t)^{1/2}$,  $\beta_t := V_{t-1}\varrho_t^{t-1}/(r_{1,1} \cdots r_{t-1,t-1})$,
          IF $t = 1$ THEN [ $\mathbf{b} := \sum_{i=1}^n u_i \mathbf{b}_i$,
                IF $\|\mathbf{b}\|^2 < A$ THEN [ $A := \|\mathbf{b}\|^2$, output $(\mathbf{b}, s, A)$, GO TO 2.1]  ]
          IF $\beta_t \geq 2^{-s}$ THEN [ $t := t - 1$,  $y_t := \sum_{i=t+1}^{t_{max}} u_i r_{t,i}/r_{t,t}$,
                      $u_t := -\lceil y_t \rfloor$,  $\varsigma_t := sign(u_t + y_t)$,  $\nu_t := 1$,  GO TO 2 ]
          ELSE IF $\beta_t \geq 2^{-s_{max}}$ THEN store $\mathbf{u}_t := (u_t, ...u_n, y_t, c_t, \nu_t, \varsigma_t, \beta_t, A)$ in $L$.
2.1.      $t := t + 1$,  $t_{max} := \max(t, t_{max})$,
          IF  $t = t_{max}$  THEN  $u_t := u_t + 1$,  $\nu_t := 1$,  $y_t := 0$
                      ELSE  $u_t := -\lceil y_t \rfloor + \lfloor \nu_t/2 \rfloor (-1)^{\nu_t} \varsigma_t$,  $\nu_t := \nu_t + 1$. ] ]
3.  perform all stages $\mathbf{u}_t = (u_t, ..., u_n, y_t, c_t, \nu_t, \varsigma_t, \beta_t, A) \in L$ with $\beta_t \geq 2^{-s}$,
    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 $c_t : \mathbb{Z}^{n-t+1} \to \mathbb{R}$:

$$c_t(u_t, ..., u_n) = \|\pi_t(\textstyle\sum_{i=t}^n u_i \mathbf{b}_i)\|^2 = \sum_{i=t}^n (\sum_{j=i}^n u_j r_{i,j})^2.$$

Hence $$c_t(u_t, ..., u_n) = (\textstyle\sum_{i=t}^n u_i r_{t,i})^2 + c_{t+1}(u_{t+1}, ..., u_n).$$

Given $u_{t+1}, ..., u_n$ Enum takes for u_t the integers that minimize $|u_t + y_t|$ for $y_t := \sum_{i=t+1}^n u_i r_{t,i}/r_{t,t}$ in order of increasing distance to $-y_t$ adding to the initial $u_t := -\lceil y_t \rfloor$ iteratively $\lfloor \nu_t/2 \rfloor (-1)^{\nu_t} \varsigma_t$ where $\varsigma_t := sign(u_t + y_t) \in \{\pm 1\}$ and ν_t numbers the iterations starting with $\nu_t = 0, 1, 2, ..$:

$$-\lceil y_t \rfloor,\ -\lceil y_t \rfloor - \varsigma_t,\ -\lceil y_t \rfloor + \varsigma_t,\ -\lceil y_t \rfloor - 2\varsigma_t,\ -\lceil y_t \rfloor + 2\varsigma_t, \cdots, -\lceil y_t \rfloor + \lfloor \nu_t/2 \rfloor (-1)^{\nu_t} \varsigma_t, \cdots,$$

where $sign(0) := 1$ and $\lceil r \rfloor$ denotes a nearest integer to $r \in \mathbb{R}$. The iteration does not decrease $|u_t + y_t|$ and $c_t(u_t, ..., u_n)$, it does not increase ϱ_t and β_t. Enum performs the stages $(u_t, ..., u_n)$ for fixed $u_{t+1}, ..., u_n$ in order of increasing $c_t(u_t, ..., u_n)$ and decreasing success rate β_t. β_t extends this priority to stages of distinct t, t' taking into accound the size of two spheres of distinct dimensions $n - t, n - t'$ The center $\zeta_t = \mathbf{b} - \pi_t(\mathbf{b}) = \sum_{i=t}^n u_i(\mathbf{b}_i - \pi_t(\mathbf{b}_i)) \in \operatorname{span}(\mathcal{L}_t)$ changes continously within New Enum which improves Enum
When step 3 performs stages $\mathbf{u}_{t^*} \in L$ the current A can be smaller than the A of $\mathbf{u}_{t^*}$ and this can make the stored β_{t^*} of $\mathbf{u}_{t^*}$ smaller than 2^{-s} so that $\mathbf{u}_{t^*}$ will not be performed but must be stored in L with the adjusted smaller values A, β_{t^*}. The stored stages $\mathbf{u}_{t^*}$ with $\beta_{t^*} \geq 2^{-s}$ should be performed in a succession giving priority to large success rates and small t^*.

Time for solving SVP for $\mathcal{L}(\mathbf{B})$. New Enum performs for each $s = 5, 6, ..., s_{max}$ only stages $\mathbf{u}_t$ with success rate $\beta_t \geq 2^{-s}$. Let $\#_{t,s,A}$ denote the number of performed stages with t, s, A. If β_t is a reliable probability then New Enum performs on average at most 2^s stages with success rate $\beta_t \geq 2^{-s}$ before decreasing A - this number of performed stages is even smaller than 2^s since New Enum also performs stages with success rate $\beta_t \geq 2^{-s+1}$. 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 add up to $\sum_{t=1}^n 2(n-t)(1+o(1)) \approx n(n+1)(1+o(1))$ arithmetic steps and it performs $O(n)$ arithmetical steps for testing that $\ln \beta_t \geq -s \ln 2$ for $t = 1, ..., n$ using $\beta_t \approx V_{t-t}\rho_t^{t-1} / \det \mathcal{L}$ assuming that $\ln(2e\pi), \ln \pi, \ln(1+x)$ for $x = 1, ..., n$ are given for free.

If the initial basis $\mathbf{B} \in \mathbb{R}^{n \times n}$ is a BKZ-basis with block size k then $||\mathbf{b}_1|| \leq \lambda_1 \gamma_k^{\frac{n-1}{k-1}}$. As New Enum performs stages with high success rates first then each decrease of A will on average halve A/λ_1^2 so that there are at most $\log_2(A/\lambda_1^2)$ iterations of step 2 that decrease the initial A of step 1. So after the initial reduction of $\mathbf{B}$ New Enum solves **SVP** for s_{max} with error probability $o(1)$ and performs on average at most $O(n^2 2^{s_{max}})$ arithmetic steps for each A. Hence **SVP** is solved by

$$2^{s_{max}}(n^2 + O(n)) 2 \tfrac{n-1}{k-1} \log_2 \gamma_k \quad \text{arithmetic steps.} \tag{4.2}$$

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 β_t as a probability function. Proposition 5.1 bounds under linear pruning the time to find $\mathbf{b}' \in \mathcal{L}(\mathbf{B})$ with $||\mathbf{b}'|| = \lambda_1$. It shows that **SVP** is polynomial time if $rd(\mathcal{L})$ is sufficiently small. Note that finding an unproved shortest vector $\mathbf{b}'$ is easier than proving $\|\mathbf{b}'\| = \boldsymbol{\lambda_1}$. New Enum finds an unproved shortest lattice vector $\mathbf{b}'$ in polynomial time under the following conditions and assumptions:

- the given lattice basis $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_n]$ and the relative density $rd(\mathcal{L})$ of $\mathcal{L}(\mathbf{B})$ satisfy
  $rd(\mathcal{L}) \leq \big(\sqrt{\frac{e\,\pi}{2\,n}} \frac{\lambda_1}{\|\mathbf{b}_1\|}\big)^{\frac{1}{2}}$, i.e., both $\mathbf{b}_1$ and $rd(\mathcal{L})$ are sufficiently small.

GSA: The basis $\mathbf{B} = \mathbf{QR}$, $\mathbf{R} = [r_{i,j}]_{1 \leq i,j \leq n}$ satisfies $r_{i,i}^2 / r_{i-1,i-1}^2 = q$ for $2 \leq i \leq n$ for some $q > 0$.

SA: There is a vector $\mathbf{b}' \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{b}'\| = \lambda_1$ and $\|\pi_t(\mathbf{b}')\|^2 \lesssim \frac{n-t+1}{n} \lambda_1^2$ for $t = 1, \ldots, n$.

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

- the vol. heur. is close: $\mathcal{M}_t^\varrho := \#\mathcal{B}_{n-t+1}(\mathbf{0}, \varrho_t) \cap \pi_t(\mathcal{L}) \approx \frac{V_{n-t+1}\varrho_t^{n-t+1}}{\det \pi_t(\mathcal{L})}$ for $\varrho_t^2 = \frac{n-t+1}{n} \lambda_1^2$.

Remarks. **1.** If **GSA** holds with $q \geq 1$ the basis $\mathbf{B}$ satisfies $\|\mathbf{b}_i\| \leq \frac{1}{2}\sqrt{i+3}\,\lambda_i$ for all i and $\|\mathbf{b}_1\| = \lambda_1$. Therefore, $q < 1$ unless $\|\mathbf{b}_1\| = \lambda_1$. **GSA** means that the reduction of the basis is "locally uniform", i.e., the $r_{i,i}^2$ form a geometric series. It is easier to work with the idealized property that all $r_{i,i}/r_{i-1,i-1}$ are equal. In practice $r_{i,i}/r_{i-1,i-1}$ slightly increases on the average with i. [BL05] studies "nearly equality". **GSA** has been used in [S03, NS06, GN08, S10, N10] and in the security analysis of NTRU in [H07, HHHW09].

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

$$\mathbf{Pr}[\, \|\pi_t(\mathbf{b}')\|^2 \leq \tfrac{n-t+1}{n} \lambda_1^2 \text{ for } t = 1, ..., n \,] = \tfrac{1}{n}$$

for random $\mathbf{b}' \in_R \operatorname{span}(\mathcal{L})$ with $||\mathbf{b}'|| = \lambda_1$.

Linear pruning means to cut off all stages $(u_t, ..., u_n)$ that satisfy $||\pi_t(\sum_{i=t}^n u_i \mathbf{b}_i)||^2 > \frac{n-t+1}{n} \lambda_1^2$.

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 $(u_t, ..., u_n)$ 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 $\#_n := \#(\mathcal{B}_n(\zeta_n, \varrho_n) \cap \mathcal{L})$, and similar for $\#_t := \#(\mathcal{B}_t(\zeta_t, \varrho_t) \cap \pi_{n-t+1}(\mathcal{L}))$, are for fixed n, ϱ_n very close for large radius ϱ_n, but can differ considerably for small ϱ_n since $\#_n$ can change a lot with the actual center ζ_n of the sphere. For small ϱ_n the minimum of $\#_n$ can be very small and then the average value for random center ζ_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 $\#_n, \#_t$ its success rate β_t frequently overestimates the success rate for the actual ζ_t. A cut of the smallest (resp. closest) lattice vector by New Enum in case that it underestimates $\#_t$ can nearly be excluded if stages are only cut for very small β_t.

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

Proposition 5.1 Let the basis $\mathbf{B} = \mathbf{QR}$, $\mathbf{R} \in \mathbb{R}^{n\times n}$ of $\mathcal{L}$ satisfy $rd(\mathcal{L}) \le \big(\frac{\lambda_1}{\|\mathbf{b}_1\|}\sqrt{\frac{e\,\pi}{2\,n}}\big)^{\frac{1}{2}}$ and **GSA** and let $\mathcal{L}$ have a shortest lattice vector $\mathbf{b}'$ that satisfies **SA**. Then Enum with linear pruning finds such $\mathbf{b}'$ under the volume heuristic in polynomial time.

Proof. For simplicity we assume that λ_1 is known. Pruning all stages $(u_t, ..., u_n)$ that satisfy $||\pi_t(\sum_{i=t}^n u_i \mathbf{b}_i)||^2 > \frac{n-t+1}{n}\lambda_1^2 =: \varrho_t^2$ does not cut off any shortest lattice vector $\mathbf{b}'$ that satisfies **SA**. The volume heuristics approxinates the number $\mathcal{M}_t^\varrho$ of performed stages $(u_t, ..., u_n)$ to

$$\begin{aligned}\mathcal{M}_t^\varrho := \#\mathcal{B}_{n-t+1}(\mathbf{0}, \varrho_t) \cap \pi_t(\mathcal{L}) &\approx \big(\sqrt{\tfrac{n-t+1}{n}}\,\lambda_1\big)^{n-t+1} V_{n-t+1}/(r_{t,t}\cdots r_{n,n}) \\ &\approx \big(\sqrt{\tfrac{n-t+1}{n}}\,\lambda_1\big)^{n-t+1}\big(\tfrac{2e\pi}{n-t+1}\big)^{\frac{n-t+1}{2}}/(r_{t,t}\cdots r_{n,n}\,\sqrt{\pi(n-t+1)}\,) \\ &< \big(\lambda_1\sqrt{\tfrac{2e\pi}{n}}\big)^{n-t+1}/(r_{t,t}\cdots r_{n,n}). \end{aligned} \tag{5.1}$$

Here $\approx$ uses Stirling's approximation $V_n = \pi^{n/2}/(n/2)! \approx (\frac{2e\pi}{n})^{n/2}/\sqrt{\pi n}$. Obviously $\|\mathbf{b}_i^*\| = r_{1,1}q^{\frac{i-1}{2}}$ holds by **GSA** and thus

$$(r_{t,t}\cdots r_{n,n})/r_{1,1}^{n-t+1} = q^{\sum_{i=t-1}^{n-1} i/2} = q^{\frac{n(n-1)-(t-1)(t-2)}{4}}.$$

For $t = 1$ this yields $q^{\frac{n-1}{4}} = (\det \mathcal{L})^{1/n}/r_{1,1} = \lambda_1/(r_{1,1}\sqrt{\gamma_n} rd(\mathcal{L}))$. Combining (5.1) with this equation and $\gamma_n < \frac{n}{e\pi}$ which holds for $n > n_0$, we get

$$\mathcal{M}_t^\varrho \lesssim \big(\tfrac{\lambda_1}{r_{1,1}}\sqrt{\tfrac{2e\pi}{n}}\big)^{n-t+1}\big(\sqrt{\tfrac{n}{e\pi}}\,rd(\mathcal{L})\,\tfrac{r_{1,1}}{\lambda_1}\big)^{n-\frac{(t-1)(t-2)}{n-1}} \tag{5.2}$$

Evaluating this upper bound for $rd(\mathcal{L}) \le \big(\frac{\lambda_1}{r_{1,1}}\sqrt{\frac{e\,\pi}{2\,n}}\big)^{\frac{1}{2}}$ yields

$$\mathcal{M}_t^\varrho \lesssim \big(\sqrt{\tfrac{n}{2\,e\,\pi}}\tfrac{r_{1,1}}{\lambda_1}\big)^{-n+t-1}\big(\sqrt{\tfrac{n}{2\,e\,\pi}}\tfrac{r_{1,1}}{\lambda_1}\big)^{+\frac{n}{2}-\frac{1}{2}\frac{(t-1)(t-2)}{n-1}}.$$

This approximate upper bound has for $t \le n$ its maximum 1 at $t = n$. This proves Prop. 5.1. □

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

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

$$\mathcal{M}_t^\varrho \lesssim \big(\sqrt{\tfrac{n}{e\pi}}\,\tfrac{r_{1,1}}{\lambda_1}\big)^{n-\frac{(t-1)(t-2)}{n-1}-n+t-1}2^{\frac{n-t+1}{2}}.$$

The exponent $n - \frac{(t-1)(t-2)}{n-1} - n + t - 1$ is maximal for $t = n/2 + 1$ with maximal value $\frac{1}{4}\frac{n^2}{n-1}$. This proves for $r_{1,1}/\lambda_1 = n^{o(1)}\sqrt{e\pi}$ the heuristic **SVP** time bound

$$O(n)\big(\sqrt{\tfrac{n}{e\pi}}\,\tfrac{r_{1,1}}{\lambda_1}\big)^{\frac{1}{4}\frac{n^2}{n-1}}2^{n/4} = n^{n/8+1.1}\ . \tag{5.3}$$

This beats under heuristics the proven **SVP** time bound $n^{\frac{n}{2e}+o(n)}$ of Hanrot, Stehle [HS07] which holds for a quasi-HKZ-basis $\mathbf{B}$ satisfying $||\mathbf{b}_1|| \le 2||\mathbf{b}_2^*||$ and having a HKZ-basis $\pi_2(\mathbf{B})$. In fact $\frac{1}{2e} \approx 0.159 > 0.125 = \frac{1}{8}$. The **SVP**-algorithm of Prop.1 can use fast BKZ for preprocessing and works even for $||\mathbf{b}_1|| \gg 2\lambda_1$ – see the attack on γ-unique **SVP** – whereas [HS07] requires quasy-HKZ-reduction for preprocessing. This eduction already guarantees $\|\mathbf{b}_1\| \leq 2\lambda_1$ and performs the main **SVP** work during preprocessing. Our **SVP** time bound $n^{n/8+o(n)}$ only assumes $\|\mathbf{b}_1\| \leq n^{o(1)}\sqrt{e\pi}\lambda_1$.

Theorem 5.4 Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m\times n}$satisfying **GSA** and $\|\mathbf{b}_1\| \leq \sqrt{e\pi}\, n^b\, \lambda_1$ for some $b \geq 0$, New Enum solves **SVP** and proves to have found a solution in time $2^{O(n)}(n^{\frac{1}{2}+b} rd(\mathcal{L}))^{\frac{n+1+o(1)}{4}}$.

Theorem 5.4 is proven in [S10], it does not assume **SA** and the vol. heuristic. Recall from remark 4 that $n^{\frac{1}{2}+b} rd(\mathcal{L}) \geq 1$ holds under **GSA**. For $b = o(1)$ Thm. 5.4 shows the **SVP**-time bound $n^{\frac{n}{8}+o(n)}$ which beats $n^{\frac{n}{2e}+o(n)}$ from Hanrot, Stehle [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 $\|\mathbf{b}\|$ for $\mathbf{b} \in \mathcal{L} \setminus \{\mathbf{0}\}$ and minimizing $\|\mathbf{t} - \mathbf{b}\|$ for $\mathbf{b} \in \mathcal{L}$ require nearly the same work if $\|\mathbf{t} - \mathcal{L}\| \approx \lambda_1$. In fact the proof of Theorem 1 yields

Enum with linear pruning solves **SVP** of $\mathcal{L}$ of $\dim \mathcal{L} = n$ 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 $\mathbf{B} = \mathbf{QR} \in \mathbb{R}^{m\times hk}$ be a lattice basis with $\mathbf{R}$ =GNF($\mathbf{B}$) $= [r_{i,j}]_{1\leq i,j\leq hk}$ with blocks $\mathbf{R}_\ell = [r_{i,j}]_{\ell k-k<i,j\leq \ell k}$, $\ell = 1, ..., h$ of size k. Then $\mathbf{B}$ is a *primal-dual basis* if

1. it is LLL-basis with **HKZ**-bases $\mathbf{R}_\ell$, $\ell = 1, ..., h$.
2. $\max_T r^2_{k\ell,k\ell} \leq \alpha\, r^2_{k\ell+1,k\ell+1}$ for $\ell = 1, ..., h-1$, where $r^2_{k\ell,k\ell}$ of GNF($\mathbf{R}_\ell\mathbf{T}$) is maximized over all $\mathbf{T} \in \mathrm{GL}_k(\mathbb{Z})$ for the $\alpha = 1/(\delta - \frac{1}{4})$ of LLL-reduction.

Theorem 6.2 [GHKN06]
Every primal-dual basis $\mathbf{B} = \mathbf{QR} \in \mathbb{R}^{m\times hk}$ of lattice $\mathcal{L}$ satisfies $||\mathbf{b}_1||^2 \leq \gamma_k(\alpha\gamma_k^2)^{\frac{h-1}{2}}(\det \mathbf{R})^{\frac{2}{hk}}$.

Proof. Def. 6.1 shows for $\mathbf{R} = [\mathbf{r_{i,j}}]_{\mathbf{1\leq i,j\leq hk}}$ and $r_{k\ell,k\ell}$ of GNF($\mathbf{R}_\ell\mathbf{T}$) that $\max_T r^2_{k\ell,k\ell} \leq \alpha\, r^2_{k\ell+1,k\ell+1}$.
The inverse matrix $\mathbf{U}_k = \begin{bmatrix} & & 1 \\ & \cdot & \\ 1 & & \end{bmatrix} \in \mathbb{Z}^{k\times k}$ yields for the lower triangular matrix $\mathbf{R}_\ell^{-t} \in \mathbf{R}^{k\times k}$ the upper triangular matrix $\mathbf{R}^*_\ell = \mathbf{U}_k\mathbf{R}_\ell^{-t}\mathbf{U}_k$, where $\mathbf{R}_\ell^{-t}$ is the transpose of the matrix $\mathbf{R}_\ell$.
The Hermite inequality $\lambda_1^2(\mathcal{L}(\mathbf{R}^*_\ell)) \leq \gamma_k\mathcal{D}_\ell^{-1/k}$ for $\mathcal{L}(\mathbf{R}^*_\ell)$, $\mathcal{D}_\ell = (\det \mathbf{R}_\ell)^2$ and HKZ-reduction of $\mathbf{R}^*_\ell$ imply

$$\mathcal{D}_\ell^{1/k} \leq \gamma_k \max\nolimits_T r^2_{k\ell,k\ell} = \gamma_k/\lambda_1^2(\mathcal{L}(\mathbf{R}^*_\ell)).$$

The HKZ-basis $\mathbf{R}_{\ell+1}$ satisfies

$$\lambda_1^2(\mathcal{L}(\mathbf{R}_{\ell+1})) = r^2_{k\ell+1,k\ell+1} \leq \gamma_k\mathcal{D}_{\ell+1}^{1/k}.$$

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

$$\mathcal{D}_\ell^{1/k} \leq \gamma_k \max\nolimits_T r^2_{k\ell,k\ell} \leq \alpha\gamma_k r^2_{k\ell+1,k\ell+1} \leq \alpha\gamma_k^2\, \mathcal{D}_{\ell+1}^{1/k}. \tag{6.1}$$

For the HKZ-basis $\mathbf{R}_1$ follows by induction over ℓ that

$$||\mathbf{b}_1||^2 \leq \gamma_k\mathcal{D}_1^{1/k} \leq \gamma_k(\alpha\gamma_k^2)^\ell\mathcal{D}_{\ell+1}^{1/k} \text{ für } \ell = 0, ..., h-1.$$

The h-th root of the product of these h inequalities proves the claim, as $\sum_{\ell=0}^{h-1} \ell = \frac{h-1}{2}$, $\prod_{\ell=1}^h \mathcal{D}_\ell = (\det \mathbf{R})^2$. □

Algorithm 6.3 : for Primal-dual reduction

```
INPUT  LLL-basis $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_{n+1}] = \mathbf{QR} \in \mathbb{Z}^{m\times hk}$, $n+1 = hk$, for $\alpha = 1/(\delta - \frac{1}{4})$,
       with blocks $\mathbf{R}_1, ..., \mathbf{R}_h \subset \mathbf{R}$ of size $k$, $[\mathbf{b}_{k\ell-k+1}, ..., \mathbf{b}_{k\ell}] = \mathbf{B}_\ell = \mathbf{Q}_\ell \mathbf{R}_\ell$, $\ell = 1$
1. HKZ-reduce the block $\mathbf{R}_{\ell+1}$ to $\mathbf{R}_{\ell+1}\mathbf{T}_k$ with $\mathbf{T}_k \in \mathrm{GL}_k(\mathbb{Z})$,  $\mathbf{B}_{\ell+1} := \mathbf{B}_{\ell+1}\mathbf{T}_k$
2. HKZ-reduce $\mathbf{R}_\ell^*$ to $\mathbf{R}_\ell^*\mathbf{T}_\star$ with $\mathbf{T}_\star \in \mathrm{GL}_k(\mathbb{Z})$, $\mathbf{B}_\ell := \mathbf{B}_\ell \mathbf{U}_k \mathbf{T}_\star^{-t}\mathbf{U}_k$, size-reduce $\mathbf{B}_\ell, \mathbf{B}_{\ell+1}$,
   compute $\mathbf{R}_{\ell,\ell+1} = \mathrm{GNF}([\mathbf{B}_\ell, \mathbf{B}_{\ell+1}])$, LLL-reduce $\mathbf{R}_{\ell,\ell+1}$ with $\delta, \alpha$ to $\mathbf{R}_{\ell,\ell+1}\mathbf{T}_{2k}$
3. IF  step 2 exchanged the columns $k$ und $k+1$ of $\mathbf{R}_{\ell,\ell+1}$
   THEN  $[\mathbf{B}_\ell, \mathbf{B}_{\ell+1}] := [\mathbf{B}_\ell, \mathbf{B}_{\ell+1}]\mathbf{T}_{2k}$,  size-reduce $[\mathbf{B}_\ell, \mathbf{B}_{\ell+1}]$,  $\ell := \max(\ell - 1, 1)$
   ELSE $\ell := \ell + 1$
4. IF  $\ell < h$  THEN  GO TO 1
OUTPUT  primal-dual basis $\mathbf{B}$
```

Comments on Alg. 6.3. Step 2 maximizes the last diagonal entry $r_{k,k}$ of $\mathrm{GNF}(\mathbf{R}_\ell\mathbf{T}) \in \mathbb{R}^{k\times k}$ for $\mathbf{T} \in \mathrm{GL}_k(\mathbb{Z})$. The reversal matrix $\mathbf{U}_k = \begin{bmatrix} & & 1 \\ & \cdot & \\ 1 & & \end{bmatrix} \in \mathbb{Z}^{k\times k}$ yields for $\mathbf{R}_\ell^{-t} \in \mathbf{R}^{k\times k}$, the transpose of $\mathbf{R}_\ell$, an upper triangular matrix $\mathbf{R}_\ell^* = \mathbf{U}_k\mathbf{R}_\ell^{-t}\mathbf{U}_k$, $\mathbf{R}_\ell^{-t}$, $\mathbf{R}_\ell^{-t}$ is the transpose of $\mathbf{R}_\ell$. HKZ-reduction of $\mathbf{R}_\ell^*$ to $\mathbf{R}_\ell^*\mathbf{T}_*$ with $\mathbf{T}_* \in \mathrm{GL}_k(\mathbb{Z})$ minimizes the first diagonal entry $r_{1,1}$ of $\mathrm{GNF}(\mathbf{R}_\ell^*\mathbf{T}_*)$ and maximizes $1/r_{1,1}$ the last diagonal entry of $\mathrm{GNF}(\mathbf{R}_\ell\mathbf{U}_k\mathbf{T}_*^{-t}\mathbf{U}_k)$. The transformation $\mathbf{T}_*$ of $\mathbf{R}_\ell^*$ yields the transformation $\mathbf{U}_k\mathbf{T}_*^{-t}\mathbf{U}_k$ for $\mathbf{R}_\ell$ and $\mathbf{B}_\ell$.

The LLL-reduction of $\mathbf{R}_{\ell,\ell+1}$ in step 2 either starts by exchanging the columns k and $k+1$ of $\mathbf{R}_{\ell,\ell+1}$ or $\mathbf{R}_{\ell,\ell+1}$ is already LLL-reduced, because $r_{k,k}$ of $\mathbf{R}_\ell$ is maximal and $r_{k+1,k+1}$ of $\mathbf{R}_{\ell+1}$ ist minimal.

The primal-dual output $\mathbf{B}$ of Alg. 6.3 is of the form $\mathbf{BT}_1$ for the input $\mathbf{B}$ of Alg.6.3 and $\mathbf{T}_1 \in \mathrm{GL}_n(\mathbb{Z})$.

Theorem 6.4 : Alg. 6.3 performs at most $\frac{n^2h}{12}\log_{1/\delta}\alpha$ iterations before arriving at $\mathcal{D}_B \leq 1$.

Proof. We replace the Lovász invariante $\mathcal{D}$ by the following invariante $\mathcal{D}_B$ where $\mathcal{D}_\ell = (\det \mathbf{R}_\ell)^2$:

$$\mathcal{D}_B := \textstyle\prod_{\ell=1}^{h-1}(\mathcal{D}_\ell/\mathcal{D}_{\ell+1})^{h^2/4-(h/2-\ell)^2}.$$

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 $\mathcal{D}_\ell \leq \alpha^{k^2}\mathcal{D}_{\ell+1}$ and therefore her $\mathcal{D}_B$-value $\mathcal{D}_B^{inp}$ satisfies $\mathcal{D}_B^{inp} \leq \alpha^{k^2 s}$ for $s =_{def} \sum_{\ell=1}^{h-1} h^2/4 - (h/2-\ell)^2$.
The well known sum $\bar{s} := \sum_{\ell=1}^{h-1}\ell^2 = h(h-1)(h-1/2)/3$ yields

$$\textstyle\sum_{\ell=1}^{h-1}(h/2-\ell)^2 = -h^2(h-1)/4 + \bar{s} = h(h-1)(h-2)/12$$

and therefore $s = (h+1)h(h-1)/6 = (h^3-h)/6$.

Hence we have $\mathcal{D}_B^{inp} \leq \alpha^{k^2(h^3-h)/6}$. An active step **3** changes from $\mathcal{D}_B$ only the factor

$$\prod_{t=\ell-1,\ell,\ell+1}(\mathcal{D}_t/\mathcal{D}_{t+1})^{t(h-t)} = \mathcal{D}_{\ell-1}^{(\ell-1)(h-\ell+1)}(\mathcal{D}_\ell\mathcal{D}_{\ell+1})^{h-2\ell-1}\mathcal{D}_\ell^2\,\mathcal{D}_{\ell+2}^{-(\ell+1)(h-\ell-1)}.$$

Every iteration with $\mathcal{D}_\ell^{new} \leq \delta^2\,\mathcal{D}_\ell^{old}$ decreases as shown $\mathcal{D}_B$ to $\mathcal{D}_B^{new} \leq \delta^2\mathcal{D}_B^{old}$. For the number #**It** of iterations until arriving at $\mathcal{D}_B \leq 1$ we get from $s = \sum_{\ell=1}^{h}(h^2/4 - (h/2-\ell)^2) = (h^3-h)/6$ that

$$\#\mathbf{It} \leq \tfrac{1}{2}\log_{1/\delta}\mathcal{D}_B^{Ein} \leq \tfrac{1}{2}\log_{1/\delta}\alpha^{k^2 s} \leq \tfrac{k^2h^3}{12}\log_{1/\delta}\alpha = \tfrac{n^2h}{12}\log_{1/\delta}\alpha. \qquad \square$$

Part **2** of def. 6.1. has been hightend by Gama, Nguyen [GN08] to

2$^+$ $\max_{\mathbf{R}_\ell'\mathbf{T}} r_{k\ell+1,k\ell+1} \leq (1+\varepsilon)\,r_{k\ell+1,k\ell+1}$ for $\ell = 1, ..., h-1$, $0 < \varepsilon \approx 0$. [GN08 *slide-reduction*]

Let $\mathbf{R}_\ell' := [r_{i,j}]_{k\ell-k+2\leq i,j\leq k\ell+1} \in \mathbb{R}^{k\times k}$ denote the segment one unit to the right of $\mathbf{R}_\ell$. $\max_{\mathbf{R}_\ell'\mathbf{T}} r^2_{k\ell+1,k\ell+1}$ marks the maximum over $\bar{r}^2_{k\ell+1,k\ell+1}$ of $[\bar{r}_{i,j}] = \mathrm{GNF}(\mathbf{R}_\ell'\mathbf{T})$ over all $\mathbf{T} \in \mathrm{GL}_k(\mathbb{Z})$. $\mathbf{B}_\ell' = [\mathbf{b}_{k\ell-k+2}, ..., \mathbf{b}_{k\ell+1}]$

is the block one unit to the right of $\mathbf{B}_\ell$.

Definition 6.5 A size-reduced basis $\mathbf{B} = \mathbf{QR} \in \mathbb{R}^{m \times hk}$ is *strong primal-dual* if $\mathbf{R}_1, ..., \mathbf{R}_h$ are HKZ-bases satisfying $\mathbf{2}^+$.

Theorem 6.6 [GN08]
A strong primal-dual basis $\mathbf{B} = \mathbf{QR} \in \mathbb{R}^{m \times hk}$ with $0 \approx \varepsilon > 0$ in $\mathbf{2}^+$ of the lattice $\mathcal{L}$ satisfies

$$||\mathbf{b}_1|| < ((1+\varepsilon)\,\gamma_k)^{\frac{1}{2}\frac{hk-k}{k-1}} (\det \mathcal{L})^{1/hk}.$$

Proof. Hermite showed for an HKZ-basis $\mathbf{R}_\ell$ that $r^{2k}_{k\ell-k+1,k\ell-k+1} \le \gamma_k^k \, \mathcal{D}_\ell$. The dual of this inequality shows for $\mathcal{D}'_\ell := (\det R'_\ell)^2$ that $\max\limits_{\mathbf{R}'_\ell \mathbf{T}} r^{2k}_{k\ell+1,k\ell+1} \ge \mathcal{D}'_\ell / \gamma_k^k$ for $\mathbf{T} \in \mathrm{GL}_k(\mathbb{Z})$.

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

$$\mathcal{D}'_\ell \le (1+\varepsilon)^{2k} \gamma_k^k \, r^{2k}_{k\ell+1,k\ell+1}. \tag{6.2}$$

Combination of (6.2) with $r^{2k}_{k\ell-k+1,k\ell-k+1} \le \gamma_k^k \, \mathcal{D}_\ell$ and $\mathcal{D}'_\ell / r^2_{k\ell+1,k\ell+1} = \mathcal{D}_\ell / r^2_{k\ell-k+1,k\ell-k+1}$ limplies

$$r_{k\ell-k+1,k\ell-k+1} \le ((1+\varepsilon)\,\gamma_k)^{\frac{k}{k-1}} r_{k\ell+1,k\ell+1} \quad \text{for } \ell = \ell_{max} \text{ und } \ell = h-1. \tag{6.3}$$

For $\ell = \ell_{max}$ we get from (6.2) and $r^{2k}_{k\ell-k+1,k\ell-k+1} \le \gamma_k^k \, \mathcal{D}_\ell$ that
$\mathcal{D}'_\ell \le (1+\varepsilon)^{2k} \gamma_k^k r^{2k}_{k\ell+1,k\ell+1} \le (1+\varepsilon)^{2k} \gamma_k^{2k} \mathcal{D}_{\ell+1}$. (6.3) implies

$$\mathcal{D}_\ell = r^2_{k\ell-k+1,k\ell-k+1} \mathcal{D}'_\ell / r^2_{k\ell+1,k\ell+1} \le ((1+\varepsilon)\,\gamma_k)^{\frac{2k}{k-1}} \mathcal{D}'_\ell. \tag{6.4}$$

The combination of the two previous unequalities yields for $\ell = \ell_{max}$

$$(\mathcal{D}_\ell \le ((1+\varepsilon)\,\gamma_k)^{\frac{2k}{k-1}+2k} \mathcal{D}_{\ell+1} \le ((1+\varepsilon)\,\gamma_k)^{\frac{2k^2}{k-1}} \mathcal{D}_{\ell+1}. \tag{6.5}$$

Therefore we get for ℓ_{max} and also for all $\ell = 1, ..., h-1$ that $\mathcal{D}_\ell \le ((1+\varepsilon)\,\gamma_k)^{\frac{2k^2}{k-1}} \mathcal{D}_{\ell+1}$.

For the HKZ-basis $\mathbf{R}_1$ this implies for $\ell = 1, ..., h$ that

$$||\mathbf{b}_1||^2 \le \gamma_k \mathcal{D}_1^{1/k} \le \gamma_k ((1+\varepsilon)\,\gamma_k)^{\frac{2k(\ell-1)}{k-1}} \mathcal{D}_\ell^{1/k}.$$

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

$$||\mathbf{b}_1||^{2h} \le \gamma_k^h ((1+\varepsilon)\,\gamma_k)^{\frac{kh(h-1)}{k-1}} (\det \mathcal{L})^{2/k.}$$

This proves the claim $|||\mathbf{b}_1||^2 \le \gamma_k ((1+\varepsilon)\,\gamma_k)^{\frac{1}{2}\frac{hk-k}{k-1}} (\det \mathcal{L})^{2/hk} < ((1+\varepsilon)\,\gamma_k)^{\frac{hk-1}{k-1}} (\det \mathcal{L})^{2/hk}$. □

Algorithm 6.7 : Accelerated, strong primal-dual reduction

```
Input  LLL-basis $\mathbf{B} = \mathbf{QR} \in \mathbb{Z}^{m \times n}$, $n = hk$, $0 \approx \varepsilon > 0$.
1. Choose $\ell$, $1 \le \ell < n$ where $\mathcal{D}_\ell / \mathcal{D}_{\ell+1}$ is maximal.
2. HKZ-reducee $\mathbf{R}_{\ell+1}$ to $\mathbf{R}_{\ell+1}\mathbf{T}$ with $\mathbf{T} \in \mathrm{GL}_k(\mathbb{Z})$, $\mathbf{B}_{\ell+1} := \mathbf{B}_{\ell+1}\mathbf{T}$, size-reduce $\mathbf{B}_{\ell+1}$,
   renew $\mathbf{R}_{\ell+1}$. HKZ-reduce $(\mathbf{R}'_\ell)^*$ to $(\mathbf{R}'_\ell)^* \mathbf{T}_\star$ with $\mathbf{T}_\star \in \mathrm{GL}_k(\mathbb{Z})$,
   $[r^{new}_{k\ell+i,k\ell+j}]_{2 \le i,j \le 1+k} := \mathrm{GNF}(\mathbf{R}'_\ell \mathbf{T}_\star^{-1} \mathbf{U}_k)$ for the reversal matrix $\mathbf{U}_k = \begin{bmatrix} & & 1 \\ & \cdot & \\ 1 & & \end{bmatrix} \in \mathbb{Z}^{k \times k}$.
3. IF $r^{new}_{k\ell+1,k\ell+1} > (1+\varepsilon)\, r^{old}_{k\ell+1,k\ell+1}$ THEN $\mathbf{B}'_\ell := \mathbf{B}'_\ell T_\star^{-1} \mathbf{U}_k$, size-reduce $\mathbf{B}'_\ell$,
   renew $\mathbf{R}_{\ell,\ell+1}$, GO TO 1
Output  strong primal-dual basis $\mathbf{B}$.
```

Theorem 6.8 Alg. 6.7 performs at most $\frac{n^2 h}{24} \log_{1+\varepsilon} \alpha$ iterationen until arriving at $\mathcal{D}_B \le 1$.

Proof. An active step **3** implayes $\mathcal{D}^{new}_\ell \le \mathcal{D}^{old}_\ell / (1+\varepsilon)^2$ and thus $\mathcal{D}^{new}_B \le \mathcal{D'}^{old}_B / (1+\varepsilon)^4$. The input-LLL-basis $\mathbf{B}$ satisfies $\mathcal{D}^{inp}_B \le \alpha^{k^2 \, \frac{h^3-h}{6}}$. Hence $\#It \le \frac{k^2 h^3}{24} \log_{1+\varepsilon} \alpha$ until $\mathcal{D}_B \le 1$.

[GN08] replaces $(1+\varepsilon)$ in $\mathbf{2}^{+}$ by $\sqrt{1+\varepsilon}$ for all $\ell \leq h-1$; [GN08] nearly proves Theorem 6.6 for slide reduced bases. Since we require $\mathbf{2}^{+}$ only for ℓ_{max} then Alg. 6.7 for strong primal-dual reduction performes 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

[AWHTT16] *Y.Aono, Y. Wang, T. Hayashi and T. Takagi*, Improved Progressive BKZ Algorithm and their Precise Cost Estimatian by Sharp Simulator. Eurocrypt 2016.

[BL05] *J. Buchmann and C. Ludwig*, Practical lattice basis sampling reduction. eprint.iacr.org, TR 072, 2005.

[Ch13] *M.Charlet*, Faktorisierung ganzer Zahlen mit dem NEW ENUM-Gitteralgorithmus. Diplomarbeit, University Frankfurt 2013.

[CP01] *R. Crandall and C. Pomerance*, Prime Numbers, A Computational Perspective. Springer-Verlag, New York, 2001.

[D30] *K. Dickman*, On the frequency of numbers containing prime factors of a certain relative magnitude. *Ark. Math. Astr. Fys.* **22**, pp. 1–14, 1930.

[D81] *J.D. Dixon*, Asymptotically Fast Factorization of Integers. *Mathematics of Computation* **36**(153), pp. 255–260, 1981.

[FP85] *U. Fincke and M. Pohst*, Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Math. of Comput.*, **44**, pp. 463–471, 1985.

[GHKN06] *N. Gama, N. How-Grave-Graham, H. Koy and P. Nguyen*, Rankin's Constant and Blockwise Lattice Reduction, In Proc. CRYPTO 2006, LNCS 4117, Springer-Verlag, Berlin/Heidelberg, pp. 112–139, 2006.

[GN08] *N. Gama and P.Q. Nguyen*, Finding Short Lattice Vectors within Mordell's Inequality. Proc. of the 2008 ACM Symposiumon on Theory of Computing, pp. 208-216, 2008.

[GNR10] *N. Gama, P.Q. Nguyen and O. Regev*, Lattice enumeration using extreme pruning, Proc. EUROCRYPT 2010, LNCS 6110, Springer-Verlag, pp. 257–278, 2010.

[G08] *A. Granville*, Smooth numbers: computational number theory and beyond. in Algorithmic Number Theory, MSRI Publications, **44**, pp. 267–323, 2008.

[HS07] *G. Hanrot and D. Stehlé*, Improved Analysis of Kannans Shortest Lattice Vector Algorithm, In Proc. CRYPTO 2007, LNCS 4622, Springer Verlag, pp. 170-186, 2007.

[H84] *A. Hildebrand*, Integers free of large prime factors and the Riemann hypothesis. *Mathematika* **31**, pp. 258–271, 1984.

[HHHW09] *P. Hirschhorn, J. Hoffstein, N. Howgrave-Graham and W. Whyte*, Choosing NTRU-Encrypt parameters in light of combined lattice reduction and MITM approaches. In Proc. ACNS 2009, LNCS 5536, Springer-Verlag, pp. 437–455, 2009.

[H07] *N. Howgrave-Graham*, A hybrid lattice–reduction and meet-in-the-middle attiack against NTRU. In Proc. CRYPTO 2007, LNCS 4622, Springer-Verlag, pp. 150–169, 2007.

[KaLe78] *G.A. Kabatiansky und V.I. Levenshtein* , Bounds for Packings on a Sphere and in Space, Problems of Information Transmission, Band 14, Seiten 1–17, 1978.

[Ka87] *R. Kannan*, Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, **12**, pp. 415–440, 1987.

[LLL82] *H.W. Lenstra Jr., A.K. Lenstra and L. Lovász*, Factoring polynomials with rational coefficients, Mathematische Annalen 261, pp. 515–534, 1982.

[Mar03] *J. Martinet*, Perfect Lattices in Euclidean Spaces. Springer-Verlag 2003.

[MG02] *D. Micciancio and S. Goldwasser*, Complexity of Lattice Problems: A Cryptographic Perspective. Kluwer Academic Publishers, Boston, London, 2002.

[MW16] *D. Micciancio and Walter*, Practical, Predictable Lattice Basis Reduktion, Eurocrypt 2016.

[MO90] *J. Mazo and A. Odlydzko*, Lattice points in high-dimensional spheres. Monatsh. Math. 110, pp. 47–61, 1990.

[MR05] *D. Micciancio and O. Regev*, Worst-case to Average-case Reductions based on Gaussian Measures, Siam J. on Computing 37(1), pp. 267-302. 2007.

[MV09] *D. Micciancio and P. Voulgaris* Faster exponential time algorithms for the shortest vector problem. ECCC Report No. 65, 2009
bibitem[MW09]MW16 *D. Micciancio and Walter* Practical, Predictitable Lattice Basis Reduktion. Eurocrypt 2016.

[MB75] *M.A. Morrison and J. Brillhart*: *A Method of Factoring and the Factorization of F_7*, Mathematics of Computation **29**(129), pp. 183 –205, 1975.

[N10] *P.Q. Nguyen*, Hermite's Constant and Lattice Algorithms. in The LLL Algorithm, Eds. P.Q. Nguyen, B. Vallée, Springer-Verlag, Jan. 2010.

[Reg04] *O. Regev*, New lattice-based cryptographic constructions, J. ACM 51 (6), pp. 899-942, 2004.

[S16] *A. Schickedanz*, Faktorisierung ganzer Zahlen durch Gitteralgorithmen. Masterarbeit, University Frankfurt, 2016.

[SE94] *C.P. Schnorr and M. Euchner*, Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming **66**, pp. 181–199, 1994.

[S03] *C.P. Schnorr*, Lattice reduction by sampling and birthday methods. Proc. STACS 2003: 20th Annual Symposium on Theoretical Aspects of Computer Science, LNCS 2007, Springer-Verlag, pp. 146–156, 2003.

[S10] *C.P. Schnorr*, Progress on LLL and lattice reduction, Proceedings LLL+25, Caen, France, June 29–July 1, 2007, The LLL Algorithm, Eds. P.Q. Phong, B. Vallée, Springer Verlag, Jan. 2010.