Solving the Shortest Lattice Vector Problem in Time $2^{2.465n}$

Xavier Pujol[1] and Damien Stehlé[2]

[1] Université de Lyon, Laboratoire LIP, CNRS-ENSL-INRIA-UCBL, 46 Allée d'Italie, 69364 Lyon Cedex 07, France
[2] CNRS, Macquarie University and University of Sydney,
Department of Mathematics and Statistics F07, University of Sydney NSW 2006, Australia
{xavier.pujol,damien.stehle}@ens-lyon.fr

Abstract. The Shortest lattice Vector Problem is central in lattice-based cryptography, as well as in many areas of computational mathematics and computer science, such as computational number theory and combinatorial optimisation. We present an algorithm for solving it in time $2^{2.465n+o(n)}$ and space $2^{1.233n+o(n)}$, where n is the lattice dimension. This improves the best previously known algorithm, by Micciancio and Voulgaris [SODA 2010], which runs in time $2^{3.199n+o(n)}$ and space $2^{1.325n+o(n)}$.

Keywords. Lattices, Shortest Vector Problem, sieve algorithms.

1 Introduction

A lattice L is a discrete subgroup of $\mathbb{R}^n$. The dimension of L is $d = \dim(\operatorname{span} L)$. Any lattice can be represented as the set of integer linear combinations of d linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$. These vectors form a basis of L and we write $L = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_d)$. Since a lattice is discrete, it has shortest non-zero vectors. Their norm $\lambda(L)$ is called the minimum of L. The *Shortest Vector Problem* (SVP) consists in finding such a vector. For the sake of simplicity, we consider only full rank integer lattices in this article, i.e., $d = n$ and $L \subseteq \mathbb{Z}^n$. SVP is known to be NP-hard under randomized reductions [1], and to remain so even if relaxed by arbitrary constant factors [13, 10].

SVP is of prime interest in cryptography for two reasons: first, the security of several lattice-based cryptosystems (see, e.g., [2, 19, 8] and the survey [15]) relies on the hardness of polynomially relaxed versions of the decisional variant of SVP (for [2], it is proved to be so in [14]); second, the main cryptanalytic tool against lattice-based cryptosystems, namely hierarchical reduction algorithms [20, 21, 7], relies on an algorithm that solves SVP in moderate dimensions. Note that SVP also occurs naturally in algorithmic number theory [4] and in combinatorial optimization [5].

The currently known algorithms for SVP can be separated in two categories. On one side, deterministic algorithms enumerate all lattice vectors shorter than a fixed bound $A \geq \lambda(L)$, by working on the Gram-Schmidt orthogonalization of the given lattice basis. They were introduced by Kannan [12] and Fincke and Pohst [6]. If given as input an LLL-reduced basis, the algorithm of Fincke and Pohst runs in time $2^{O(n^2)}$, while the worst-case complexity of Kannan's algorithm is $n^{\frac{n}{2e}+o(n)}$ (this complexity upper bound is proved in [9]). Note that for all complexity statements, we omit a multiplicative factor that is polynomial in the bitsize of the lattice basis. Enumeration algorithms require a polynomially bounded amount of space. On the other side, the algorithms with the best theorical complexity are probabilistic (Monte Carlo) sieve algorithms, the first of which was introduced by Ajtai et al. in [3]. The initial time and space complexity bounds of $2^{O(n)}$ were later improved by Regev [18], then decreased to $2^{5.90n+o(n)}$ and $2^{2.95n}$ respectively by Nguyen and Vidick [17] and recently decreased further to $2^{3.40n+o(n)}$ and $2^{1.97n+o(n)}$ by Micciancio and Voulgaris [16]. The authors of [16] also introduced `ListSieve`, another sieve algorithm which solves SVP in time $2^{3.199n+o(n)}$ and space $2^{1.325n+o(n)}$. Contrary to enumeration algorithms, sieve algorithms require an exponential amount of space.

Our result. We present an improved version of `ListSieve` which solves SVP in time $2^{2.465n+o(n)}$ and space $2^{1.233n+o(n)}$ (the constants are chosen to minimize the time complexity: a better space complexity can

be achieved at the expense of increasing the time complexity). The main new ingredient is the use of the birthday paradox to decrease the number of vectors that must be generated to ensure that the sieve succeeds.

The improvement is most easily described with the Ajtai et al. algorithm (see the simplified description of [17]). The latter samples *iid* vectors in $L \cap \mathcal{B}_n(0, c\lambda_1(L))$ for a small constant c, which contains only a finite number N of lattice points. The proof of correctness requires that the same vector is sampled twice with high probability, and another technical constraint implies that only a small fraction $1/x$ of all the vectors is taken into account. In the previous analyses, the number of required vectors was Nx. However, the birthday paradox ensures that $O(\sqrt{N}x)$ vectors suffice. In the case of the Ajtai et al. algorithm, this leads to a time complexity bound of $2^{2.648n+o(n)}$. We omit the proof, as the improved variant of `ListSieve` provides a better complexity bound, although it requires more care to ensure that the sampled vectors are iid.

Notations. We write $\|\cdot\|$ for the euclidean norm and $\langle\cdot,\cdot\rangle$ for the dot product. If $\mathbf{u}$ and $\mathbf{v}$ are non-zero vectors, we define $\phi_{\mathbf{u},\mathbf{v}}$ as the angle between $\mathbf{u}$ and $\mathbf{v}$. We use the notation log for the natural logarithm. All balls $\mathcal{B}_n(x,r)$ are closed, and if x is omitted, it means that the ball is centred on 0. The bitsize $|B|$ of a basis B is sum of the bitsizes of its vectors. We let $\mathcal{P}(B)$ denote the fundamental parallelepiped spanned by the basis B. Finally, for any $\mathbf{u} = \sum_i u_i \mathbf{b}_i$, we write $\mathbf{u} \mod \mathcal{P}(B)$ for $\sum_i (u_i - \lfloor u_i \rfloor)\mathbf{b}_i$.

2 The SVP algorithm

We first recall the `ListSieve` algorithm from [16], in Figure 1. It builds a list T of lattice vectors, reducing each randomly generated vector with vectors previously added to the list. `ListSieve` keeps adding vectors to T until it finds a lattice vector whose norm corresponds the guessed value μ for the lattice minimum λ. It makes use of sampling and reduction functions, described in Figures 3 and 4. The reduction is done on perturbed vectors $\mathbf{u}' = \mathbf{u} + \mathbf{x}$ instead of lattice vectors $\mathbf{u} \in L$, with randomly chosen $\mathbf{x}$'s. If the perturbations are large enough, a given perturbed vector can sometimes be obtained from several lattice vectors. The fact that the reduction function is oblivious to the lattice vector is crucial for the proof of correctness.

Input: A basis B, $\mu \simeq \lambda(\mathcal{L}(B))$, $\xi > \frac{1}{2}$, N_1.
Output: A shortest non-zero vector of $\mathcal{L}(B)$.
Choose $(\mathbf{x}_1, \ldots, \mathbf{x}_{N_1})$ randomly in $\mathcal{B}_n(0, \xi\mu)$.
$T \leftarrow \{0\}$.
For $i = 1$ to N_1, do
  $(\mathbf{t}_i, \mathbf{t}'_i) \leftarrow$ `Reduction`(`NewPair`$(B, \mathbf{x}_i), T)$,
  If $\exists \mathbf{t}_j \in T$, $0 < \|\mathbf{t}_i - \mathbf{t}_j\| \leq \mu$, then return $\mathbf{t}_i - \mathbf{t}_j$;
  ElseIf $\mathbf{t}_i \notin T$, then $T \leftarrow T \cup \{\mathbf{t}_i\}$.

Fig. 1. The `ListSieve` algorithm

The new algorithm `ListSieve-Birthday` is described in Figure 2. It runs `ListSieve` for a while, and then adds to a second list U the vectors reduced with respect to the `ListSieve` list T. Hence the vectors of the second list U are both short (with high probability) and iid.

In Section 3, we will prove the following result.

Theorem 1 *Let $L \subseteq \mathbb{Z}^n$ be an n-dimensional lattice and $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of L. With suitable choices for the parameters μ, ξ, r_0, N_1 and N_2, the algorithm* `ListSieve-Birthday` *can be used to solve SVP on B with probability $1 - 2^{-\Omega(n)}$ in time $2^{2.465n+o(n)} \cdot \mathrm{Poly}(|B|)$ and space $2^{1.233n+o(n)} \cdot \mathrm{Poly}(|B|)$.*

3 Analysis of `ListSieve-Birthday`

In this section we set $\lambda = \lambda(L)$ and fix the parameters $\xi > 1/2$ and $r_0 > 2\xi$. Wlog, we assume that:

Input: A basis B, $\mu \simeq \lambda(\mathcal{L}(B))$, $\xi > \frac{1}{2}$, $r_0 > 2\xi$, N_1, N_2.
Output: A shortest non-zero vector of $\mathcal{L}(B)$.
Choose $(\mathbf{x}_1, \ldots, \mathbf{x}_{N_1}, \mathbf{y}_1, \ldots, \mathbf{y}_{N_2})$ randomly in $\mathcal{B}_n(0, \xi\mu)$.
$T \leftarrow \emptyset$, $U \leftarrow \emptyset$.
For $i = 1$ to N_1, do
  $(\mathbf{t}_i, \mathbf{t}'_i) \leftarrow$ `Reduction`(`NewPair`$(B, \mathbf{x}_i), T)$,
  If $\|\mathbf{t}_i\| \geq r_0\mu$ then $T \leftarrow T \cup \{\mathbf{t}_i\}$.
For $i = 1$ to N_2, do
  $(\mathbf{u}_i, \mathbf{u}'_i) \leftarrow$ `Reduction`(`NewPair`$(B, \mathbf{y}_i), T)$,
  $U \leftarrow U \cup \{\mathbf{u}_i\}$.
Find closest distinct points $(\mathbf{s}_1, \mathbf{s}_2)$ in U (fail if they do not exist).
Return $\mathbf{s}_1 - \mathbf{s}_2$.

Fig. 2. The SVP algorithm: `ListSieve-Birthday`

Input: A basis B and a perturbation $\mathbf{x}$.
Output: A lattice vector $\mathbf{u}$ and a perturbed vector $\mathbf{u}'$.
$\mathbf{u}' \leftarrow (-\mathbf{x}) \mod \mathcal{P}(B)$.
$\mathbf{u} \leftarrow \mathbf{u}' + \mathbf{x}$.
Return $(\mathbf{u}, \mathbf{u}')$.

Fig. 3. The `NewPair` algorithm

Input: A pair $(\mathbf{u}, \mathbf{u}')$ generated by `NewPair` and a list $T \subseteq L$.
Output: A reduced pair $(\mathbf{u}, \mathbf{u}')$.
While $\exists \mathbf{w} \in T : \|\mathbf{u}' - \mathbf{w}\| < \left(1 - \frac{1}{n}\right) \|\mathbf{u}'\|$,
  $(\mathbf{u}, \mathbf{u}') \leftarrow (\mathbf{u} - \mathbf{w}, \mathbf{u}' - \mathbf{w})$.
Return $(\mathbf{u}, \mathbf{u}')$.

Fig. 4. The `Reduction` algorithm

- The integer basis B is LLL-reduced. This can be done in time $\mathrm{Poly}(|B|)$.
- We have $\max_i \|\mathbf{b}_i\| = 2^{O(n)}\lambda$ (if the basis is LLL-reduced then the basis vectors that are too long cannot come into play, see [17, Lemma 3.3]).
- We know μ such that $\lambda \leq \mu < \left(1 + \frac{1}{n}\right)\lambda$. This can be ensured by trying a polynomial number of values for μ.

3.1 Known results

The following lemmas are variants of those given in [16]. Theorem 2, which is the main tool for Lemmas 3 and 4, is proven in [11]. For the sake of completeness, we give proofs of Lemmas 3, 4 and 5 in the appendix.

Theorem 2 (Kabatiansky and Levenshtein) *Let $E \subseteq \mathbb{R}^n \backslash \{0\}$. If there exists $\phi_0 > 0$ such that for any $\mathbf{u}, \mathbf{v} \in E$, we have $\phi_{\mathbf{u},\mathbf{v}} \geq \phi_0$, then $|E| \leq 2^{cn+o(n)}$ with $c = -\frac{1}{2}\log_2\left[1 - \cos(\min(\phi_0, 62.99°))\right] - 0.099$.*

Lemma 3 *Let $c_b = \log_2 r_0 + 0.401$. For any lattice L, we have $|\mathcal{B}_n(0, r_0\mu) \cap L| \leq N_B(n) = 2^{c_b n+o(n)}$.*

Lemma 4 *Let $c_t = -\frac{1}{2}\log_2\left(1 - \frac{2\xi}{r_0}\right) + 0.401$. At any moment during the execution of* `ListSieve-Birthday`*, the list T contains at most $N_T(n) = 2^{c_t n+o(n)}$ vectors.*

Lemma 5 *Let $c_g = -\frac{1}{2}\log_2\left(1 - \frac{1}{4\xi^2}\right)$ and $\mathbf{s}$ be a shortest non-zero vector of L. Let $I_\mathbf{s} = \mathcal{B}_n(0, \xi\mu) \cap \mathcal{B}_n(-\mathbf{s}, \xi\mu)$. If $\mathbf{x}$ is chosen uniformly in $\mathcal{B}_n(0, \xi\mu)$, then $\Pr(\mathbf{x} \in I_\mathbf{s}) \geq \frac{1}{N_G(n)}$ with $N_G(n) = 2^{c_g n+o(n)}$.*

3.2 Proof of Theorem 1

Let $N_1^{\max} = \lceil 4N_G N_T \rceil$ and $N_2 = \lceil 8N_G \rceil \lceil \sqrt{N_B} \rceil$. We sample N_1 uniformly in the interval $[0, N_1^{\max} - 1]$.

The purpose of Lemmas 6 and 7 is to prove that with high probability, there are sufficiently many vectors $\mathbf{u}_i$ in U such that $\mathbf{u}_i$ is short (i.e., $\|\mathbf{u}_i\| < r_0\mu$) and $\mathbf{y}_i \in I_{\mathbf{s}}$ (in that case, the perturbed vector $\mathbf{u}'_i$ could be associated to another lattice vector, namely $\mathbf{u}'_i + \mathbf{s}$ with the perturbation $\mathbf{y}_i + \mathbf{s}$).

Lemma 6 *Consider* `ListSieve-Birthday` *with $N_1 = N_1^{max}$. For $i \le N_1^{max}$, we define the event $E_i : \|\mathbf{t}_i\| < r_0\mu$. We let $p_i = \Pr(E_i \mid \mathbf{x}_i \in I_{\mathbf{s}})$, where the probability is taken over the randomness of $x_1, \ldots, x_i$, and $J = \{i \le N_1^{max} : p_i \le \frac{1}{2}\}$. Then $|J| \le N_1^{max}/2$.*

Proof. Assume (for contradiction) that $|J| > N_1^{\max}/2$. Then by Lemma 5 we have

$$\sum_{i \in J} (1 - p_i) \Pr(\mathbf{x}_i \in I_{\mathbf{s}}) \ge \frac{|J|}{2N_G} > N_T.$$

This contradicts the following inequalities. The last one derives from Lemma 4.

$$\sum_{i \in J} (1 - p_i) \Pr(\mathbf{x}_i \in I_{\mathbf{s}}) = \sum_{i \in J} \Pr((\neg E_i) \cap (\mathbf{x}_i \in I_{\mathbf{s}})) \le \sum_{i \ge 1} \Pr(\neg E_i) \le N_T.$$

In the second loop of `ListSieve-Birthday`, we do not add any point to T. Therefore, the points that are added to U are iid. The procedure to reduce points being the same in both loops, we have that for any $i \le N_2$ such that $\mathbf{y}_i \in I_{\mathbf{s}}$, the probability that $\|\mathbf{u}_i\| < r_0\mu$ is p_{N_1+1}. Since N_1 is sampled uniformly in $[0, N_1^{\max} - 1]$, we have $p_{N_1+1} \ge \frac{1}{2}$ with probability $\ge \frac{1}{2}$, by Lemma 6.

Lemma 7 *If n is sufficiently large, then with probability $\ge 1/4$ (taken over the randomness of N_1, the x_k's and the y_k's), there exist two distinct indices $i, j \le N_2$ such that $\mathbf{u}_i = \mathbf{u}_j$ and $\mathbf{y}_i, \mathbf{y}_j \in I_{\mathbf{s}}$.*

Proof. Let $N = 2\lceil \sqrt{N_B} \rceil$. Until the end of the current proof, we assume that $p_{N_1+1} \ge \frac{1}{2}$, which occurs with probability $\ge \frac{1}{2}$ and implies that $\Pr(\|\mathbf{u}_i\| \le r_0\mu \mid \mathbf{y}_i \in I_{\mathbf{s}}) \ge \frac{1}{2}$ for all $i \le N_2$. Let $X = |\{i \le N_2 : (\|\mathbf{u}_i\| \le r_0\mu) \cap (\mathbf{y}_i \in I_{\mathbf{s}})\}|$. Lemma 5 gives

$$\Pr((\|\mathbf{u}_i\| \le r_0\mu) \cap (\mathbf{y}_i \in I_{\mathbf{s}})) = \Pr(\|\mathbf{u}_i\| \le r_0\mu \mid \mathbf{y}_i \in I_{\mathbf{s}}) \Pr(\mathbf{y}_i \in I_{\mathbf{s}}) \ge \frac{1}{2N_G}.$$

The variable X has a binomial distribution of parameter $p \ge \frac{1}{2N_G}$. We have $\mathbb{E}(X) = pN_2 \ge 2N$ and $\mathrm{Var}(X) = p(1-p)N_2 \le \mathbb{E}(X)$. Therefore, by using Chebyshev's inequality, we have (since $N_B \ge 25$ holds for n large enough, we have $N \ge 10$):

$$\begin{aligned}\Pr(X \le N) \le \Pr(|X - \mathbb{E}(X)| \ge \mathbb{E}(X) - N) &\le \frac{\mathrm{Var}(X)}{(\mathbb{E}(X) - N)^2} \\ &\le \frac{\mathbb{E}(X)}{(\mathbb{E}(X) - N)^2} \le \frac{2}{N} \le \frac{1}{5}.\end{aligned}$$

So with high probability `ListSieve-Birthday` samples at least N iid points in $S_0 = \mathcal{B}_n(r_0\mu) \cap L$. The probability that a collision occurs is minimized when the distribution is uniform, i.e., the probability of each point is $1/|S_0|$. Since we have chosen $N \ge \sqrt{|S_0|}$ (by Lemma 3), the birthday paradox implies that the probability will be large. More precisely it is greater than

$$\frac{4}{5}\left(1 - \prod_{i<N}\left(1 - \frac{i}{|S_0|}\right)\right) \ge \frac{4}{5}\left(1 - \exp\left(-\frac{N(N-1)}{2N_B}\right)\right) \ge \frac{4}{5}\left(1 - \frac{1}{\mathrm{e}}\right),$$

where we used the fact that $|S_0| \le N_B$ (by Lemma 3).

In order to prove that `ListSieve-Birthday` returns a shortest non-zero vector with high probability, we introduce a modified version `ListSieve-Birthday2`. Recall that in Lemma 5, we have fixed a shortest vector $\mathbf{s}$ and defined $I_{\mathbf{s}} = \mathcal{B}_n(0, \xi\mu) \cap \mathcal{B}_n(-\mathbf{s}, \xi\mu)$. For $\mathbf{x}$ in $\mathcal{B}_n(0, \xi\mu)$, let $\tau(\mathbf{x}) = \mathbf{x} + \mathbf{s}$ if $\mathbf{x} \in I_{\mathbf{s}}$ and $\tau(\mathbf{x}) = -\mathbf{x}$ if $\mathbf{x} \notin I_{\mathbf{s}}$. The difference between `ListSieve-Birthday` and `ListSieve-Birthday2` is that in the latter the function τ is applied to each $\mathbf{y}_i$ with probability $\frac{1}{2}$ immediately after it is chosen. If $\mathbf{x}$ is sampled uniformly in $\mathcal{B}_n(0, \xi\mu)$, then so is $\tau(\mathbf{x})$. As a consequence, the outputs of `ListSieve-Birthday` and `ListSieve-Birthday2` follow the same distribution. For $\mathbf{x} \in I_{\mathbf{s}}$, let $(\mathbf{u}, \mathbf{u}') = \texttt{Reduction}(\texttt{NewPair}(\mathbf{x}), T)$ and $(\mathbf{v}, \mathbf{v}') = \texttt{Reduction}(\texttt{NewPair}(\tau(\mathbf{x})), T)$. The fact that $\mathbf{x} \in I_{\mathbf{s}}$ implies that $\mathbf{x} = \tau(\mathbf{x}) \mod \mathcal{P}(B)$. The actions of `Reduction` depend only on the perturbed vector, so we have $\mathbf{v}' = \mathbf{u}'$ and $\mathbf{v} = \mathbf{u} + \mathbf{s}$.

Lemma 8 *Let $c_{time} = \max(c_g + 2c_t, 2c_g + c_b)$ and $c_{space} = \max(c_t, c_g + c_b/2)$. Then with probability $\geq \frac{1}{16}$ and for sufficiently large n,* `ListSieve-Birthday` *returns a shortest non-zero vector of L in time $2^{c_{time}n+o(n)}$ and space $2^{c_{space}n+o(n)}$.*

Proof. We start with the correctness property. Assume that we run the algorithms `ListSieve-Birthday` and `ListSieve-Birthday2` on the same input and that they make the same random choices for N_1 and the perturbations. By Lemma 7, with probability $\geq \frac{1}{4}$, there exist two distinct indices i and j such that $\mathbf{u}_i = \mathbf{u}_j$ and $\mathbf{y}_i, \mathbf{y}_j \in I_{\mathbf{s}}$ in `ListSieve-Birthday`. With probability $\geq \frac{1}{4}$, `ListSieve-Birthday2` applies τ to $\mathbf{y}_i$ but not to $\mathbf{y}_j$. Therefore it outputs $\mathbf{u}_i + \mathbf{s}$ and $\mathbf{u}_j = \mathbf{u}_i$, because it chooses the same perturbations as `ListSieve-Birthday`. Thus, with probability $\geq \frac{1}{16}$, there exist two vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ in the second list of `ListSieve-Birthday2` such that $\|\mathbf{s}_1 - \mathbf{s}_2\| = \lambda(L)$. This also holds for `ListSieve-Birthday`, since it has the same output distribution.

The space complexity is $|T| + |U|$. By Lemma 4, we have $|T| \leq 2^{c_t n + o(n)}$, and, by definition of N_2, we have $|U| \leq 2^{(c_g + c_b/2)n + o(n)}$. Since $\|\mathbf{b}_i\| = 2^{O(n)}\mu$ for all i, the complexity of `Reduction` is $|T| \operatorname{Poly}(n, |B|)$. Omitting the polynomial factor, the time complexity of the first loop is $|T|N_1 \leq |T|N_1^{\max} \leq 2^{(c_g + 2c_t)n + o(n)}$. The time required to find a closest pair of points in U with the naive algorithm is $|U|^2$. Finally, the time complexity of the second loop is $|T| \cdot |U| \leq 2^{(c_t + c_g + c_b/2)n + o(n)}$, which is negligibly smaller than the cost of one of the other components.

Proof of Theorem 1. The time complexity is minimized when $2c_t = c_g + c_b$. By Lemmas 3, 4 and 5, this is equivalent to $r_0 = 2\xi + 2^{0.401}\sqrt{1 - \frac{1}{4\xi^2}}$. Optimizing with respect to ξ leads to $\xi \simeq 0.9476$, $r_0 \simeq 3.0169$, $c_{\text{time}} \leq 2.465$ and $c_{\text{space}} \leq 1.233$. Calling the algorithm n times ensures that it succeeds with probability exponentially close to 1. □

4 Open Problems

In [16] the authors drew a list of important open problems about `ListSieve`, in particular on the necessity of perturbing the initial lattice vectors. These carry over to `ListSieve-Birthday`. Another question that is specific to the latter is whether it is necessary to divide it into two steps to ensure that the vectors of the second list are iid. At first sight, it seems to be an artefact of the proof, but we did not manage to avoid it.

Acknowledgments This work is part of the Australian Research Council Discovery Project DP0880724 "Integral lattices and their theta series". We thank D. Micciancio and P. Voulgaris for helpful discussions.

References

1. M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the 30th Symposium on the Theory of Computing (STOC 1998)*, pages 284–293. ACM Press, 1998.
2. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th Symposium on the Theory of Computing (STOC 1997)*, pages 284–293. ACM Press, 1997.

3. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd Symposium on the Theory of Computing (STOC 2001)*, pages 601–610. ACM Press, 2001.
4. H. Cohen. *A Course in Computational Algebraic Number Theory, 2nd edition*. Springer-Verlag, 1995.
5. F. Eisenbrand. *50 Years of Integer Programming 1958-2008, From the Early Years to the State-of-the-Art*, chapter Integer Programming and Algorithmic Geometry of Numbers. Springer-Verlag, 2009.
6. U. Fincke and M. Pohst. A procedure for determining algebraic integers of given norm. In *Proceedings of EUROCAL*, volume 162 of *Lecture Notes in Computer Science*, pages 194–202. Springer-Verlag, 1983.
7. N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *Proceedings of the 40th Symposium on the Theory of Computing (STOC 2008)*. ACM, 2008.
8. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Symposium on the Theory of Computing (STOC 2008)*, pages 197–206. ACM Press, 2008.
9. G. Hanrot and D. Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In *Proceedings of Crypto 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 170–186. Springer-Verlag, 2007.
10. I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proceedings of the 39th Symposium on the Theory of Computing STOC 2007*, pages 469–477. ACM, 2007.
11. G. A. Kabatiansky and V. I. Levenshtein. Bounds for packings on a sphere and in space. *Problemy Peredachi Informatsii*, 14(1):3–25, 1978. Available in English in Problems of Information Transmission 14(1):1–17.
12. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the 15th Symposium on the Theory of Computing (STOC 1983)*, pages 99–108. ACM Press, 1983.
13. S. Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, 2005.
14. V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *Proceedings of Crypto 2009*, pages 577–594, 2009.
15. D. Micciancio and O. Regev. *Post-Quantum Cryptography*, chapter Lattice-based Cryptography. Springer-Verlag, 2008.
16. D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem, 2010. To appear in the proceedings of SODA'10, preliminary versions available at the URLs `http://eccc.hpi-web.de/report/2009/065` and `http://cseweb.ucsd.edu/~pvoulgar/`.
17. P. Nguyen and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2), 2008.
18. O. Regev. Lattices in computer science, 2004. Lecture notes of a course given at the Tel Aviv University. Available at the URL `http://www.cs.tau.ac.il/~odedr/teaching/lattices_fall_2004/`.
19. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Symposium on the Theory of Computing (STOC 2005)*, pages 84–93. ACM Press, 2005.
20. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theor. Comput. Sci*, 53:201–224, 1987.
21. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematics of Programming*, 66:181–199, 1994.

Known proofs

Proof of Lemma 3. Let $\alpha = 1+\frac{1}{n}$. The ball $\mathcal{B}_n\left(\frac{\lambda}{2}\right)$ contains exactly one lattice point. We cover $\mathcal{B}_n(r_0\mu)\setminus\mathcal{B}_n\left(\frac{\lambda}{2}\right)$ with coronas $T_r = \mathcal{B}_n(\alpha r)\setminus\mathcal{B}_n(r)$ for $r = \frac{\lambda}{2}, \frac{\lambda}{2}\alpha, \ldots, \frac{\lambda}{2}\alpha^k$, with $k = \lceil n\log_2(2r_0)\rceil = O(n)$. It suffices to prove that any corona T_r contains at most $2^{c_b n+o(n)}$ lattice points.

Let $\mathbf{u}$ and $\mathbf{v}$ be two distinct lattice vectors in $T_r \cap \mathcal{B}_n(r_0\mu)$. We have $\langle \mathbf{u}-\mathbf{v}, \mathbf{u}-\mathbf{v}\rangle \geq \lambda^2$, so $\langle \mathbf{u},\mathbf{v}\rangle \leq \frac{1}{2}\left(\|\mathbf{u}\|^2+\|\mathbf{v}\|^2-\lambda^2\right)$. This implies that:

$$\begin{aligned}\cos\phi_{\mathbf{u},\mathbf{v}} = \frac{\langle \mathbf{u},\mathbf{v}\rangle}{\|\mathbf{u}\|\cdot\|\mathbf{v}\|} &\leq \frac{1}{2}\left(\frac{\|\mathbf{u}\|}{\|\mathbf{v}\|}+\frac{\|\mathbf{v}\|}{\|\mathbf{u}\|}-\frac{\lambda^2}{\|\mathbf{u}\|\cdot\|\mathbf{v}\|}\right)\\ &\leq 1+\frac{1}{n}-\frac{\lambda^2}{2r_0^2\mu^2} \leq 1+\frac{1}{n}-\frac{1}{2(1+\frac{1}{n})^2 r_0^2} \underset{n\to\infty}{\longrightarrow} 1-\frac{1}{2r_0^2}.\end{aligned}$$

For any $\varepsilon \in (0, \frac{1}{2r_0^2})$ and sufficiently large n we can apply Theorem 2 with $\phi_0 = \cos^{-1}\left(1-\frac{1}{2r_0^2}+\varepsilon\right) \leq 60°$. □

Proof of Lemma 4. First, we bound the norm of any vector of T. `NewPair` returns $(\mathbf{t}, \mathbf{t}')$ such that $\mathbf{t}' \in \mathcal{P}(B)$ and $\|\mathbf{t}' - \mathbf{t}\| \leq \xi\mu$. We have assumed that $\max_i \|\mathbf{b}_i\| = 2^{O(n)}\lambda$. Hence $\|\mathbf{t}'\| \leq n \max_i \|\mathbf{b}_i\| \leq 2^{O(n)}\mu$. After applying `Reduction`, the norm of $\mathbf{t}'$ does not increase and $\mathbf{t}' - \mathbf{t}$ is unchanged, so, for any $\mathbf{t}_i \in T$, we have $r_0\mu \leq \|\mathbf{t}_i\| \leq (2^{O(n)} + \xi)\mu$. It now suffices to prove that any $T_r = \{\mathbf{t}_i \in T \mid r\mu \leq \|\mathbf{t}_i\| \leq \left(1 + \frac{1}{n}\right) r\mu\}$ for $r \geq r_0$ contains at most $2^{c_t n + o(n)}$ points. Indeed, the list T is contained in a union of $O(n^2)$ sets T_r.

Let $i < j$ such that $\mathbf{t}_i, \mathbf{t}_j \in T_r$. The idea of the proof is that for large n, the angle between $\mathbf{t}'_j$ and $\mathbf{t}_i$ is not far from being above $\frac{\pi}{3}$ because $\mathbf{t}_i$ was already in T when $\mathbf{t}_j$ was reduced. We use the inequality $\|\mathbf{t}_j - \mathbf{t}'_j\| \leq \xi\mu$ to obtain a lower bound for $\phi_{\mathbf{t}_i, \mathbf{t}_j}$ and then apply Theorem 2.

Note that $\|\mathbf{t}'_j\| \leq \|\mathbf{t}_j\| + \xi\mu \leq 3r\mu$. Since $\mathbf{t}_j$ was added after $\mathbf{t}_i$, we have:

$$\begin{aligned}
\|\mathbf{t}'_j - \mathbf{t}_i\| &> \left(1 - \frac{1}{n}\right) \|\mathbf{t}'_j\| \\
\langle \mathbf{t}'_j - \mathbf{t}_i, \mathbf{t}'_j - \mathbf{t}_i \rangle &> \left(1 - \frac{1}{n}\right)^2 \langle \mathbf{t}'_j, \mathbf{t}'_j \rangle \geq \left(1 - \frac{2}{n}\right) \langle \mathbf{t}'_j, \mathbf{t}'_j \rangle \\
\langle \mathbf{t}'_j, \mathbf{t}_i \rangle &< \frac{1}{2}\left[\|\mathbf{t}_i\|^2 + \frac{2}{n}\|\mathbf{t}'_j\|^2\right] \leq \frac{1}{2}\|\mathbf{t}_i\|^2 + \frac{1}{n}(3r\mu)^2.
\end{aligned}$$

Moreover, we have $\langle \mathbf{t}_j - \mathbf{t}'_j, \mathbf{t}_i \rangle \leq \xi\mu\|\mathbf{t}_i\|$. We can now bound $\cos(\phi_{\mathbf{t}_i, \mathbf{t}_j})$.

$$\begin{aligned}
\langle \mathbf{t}_j, \mathbf{t}_i \rangle = \langle \mathbf{t}'_j, \mathbf{t}_i \rangle + \langle \mathbf{t}_j - \mathbf{t}'_j, \mathbf{t}_i \rangle &\leq \frac{1}{2}\|\mathbf{t}_i\|^2 + \frac{1}{n}(3r\mu)^2 + \xi\mu\|\mathbf{t}_i\| \\
\cos(\phi_{\mathbf{t}_i, \mathbf{t}_j}) = \frac{\langle \mathbf{t}_j, \mathbf{t}_i \rangle}{\|\mathbf{t}_i\| \cdot \|\mathbf{t}_j\|} &\leq \frac{1}{2}\frac{\|\mathbf{t}_i\|}{\|\mathbf{t}_j\|} + \frac{1}{n} \cdot \frac{(3r\mu)^2}{\|\mathbf{t}_i\| \cdot \|\mathbf{t}_j\|} + \frac{\xi\mu}{\|\mathbf{t}_j\|} \\
&\leq \frac{1}{2}\left(1 + \frac{1}{n}\right) + \frac{9}{n} + \frac{\xi}{r} \\
&\leq \frac{1}{2} + \frac{\xi}{r_0} + O\left(\frac{1}{n}\right).
\end{aligned}$$

The bound on $|T_r|$ follows directly from Theorem 2. □

Proof of Lemma 5. The set $\mathcal{B}_n(0, \mu\xi) \cap \mathcal{B}_n(-\mathbf{s}, \mu\xi)$ is the union of two identical n-sphere caps of height $\mu\xi - \frac{\lambda}{2} \geq \mu\left(\xi - \frac{1}{2}\right)$. Let C be one of these. It contains a cone of height $h = \mu\left(\xi - \frac{1}{2}\right)$ whose basis is an $(n-1)$-sphere of radius $r = \mu\sqrt{\xi^2 - \frac{1}{4}}$. Moreover $\mathcal{B}_n(r)$ is included in a cylinder of basis $\mathcal{B}_{n-1}(r)$ and height $2r$ so we have $\operatorname{Vol}\mathcal{B}_n(r) \leq 2r \operatorname{Vol}\mathcal{B}_{n-1}(r)$. Then

$$\frac{\operatorname{Vol} C}{\operatorname{Vol}\mathcal{B}_n(\xi\mu)} \geq \frac{h}{n} \cdot \frac{\operatorname{Vol}\mathcal{B}_{n-1}(r)}{\operatorname{Vol}\mathcal{B}_n(\xi\mu)} \geq \frac{h}{2rn} \cdot \frac{\operatorname{Vol}\mathcal{B}_n(r)}{\operatorname{Vol}\mathcal{B}_n(\xi\mu)} \geq \frac{\xi - \frac{1}{2}}{2n\sqrt{\xi^2 - \frac{1}{4}}}\left(1 - \frac{1}{4\xi^2}\right)^{n/2}. \square$$