A Three-Level Sieve Algorithm for the Shortest Vector Problem

Feng Zhang[(✉)], Yanbin Pan[(✉)], and Gengran Hu

Key Laboratory of Mathematics Mechanization,
Academy of Mathematics & NCMIS,
Chinese Academy of Sciences, 100190 Beijing, China
{zhangfeng,panyanbin}@amss.ac.cn, hudiran10@mails.ucas.ac.cn

Abstract. In AsiaCCS 2011, Wang *et al.* proposed a two-level heuristic sieve algorithm for the shortest vector problem in lattices, which improves the Nguyen-Vidick sieve algorithm. Inspired by their idea, we present a three-level sieve algorithm in this paper, which is shown to have better time complexity. More precisely, the time complexity of our algorithm is $2^{0.3778n+o(n)}$ polynomial-time operations and the corresponding space complexity is $2^{0.2833n+o(n)}$ polynomially many bits.

Keywords: Lattice · Shortest vector problem · Sieve algorithm · Sphere covering

1 Introduction

Lattices are discrete subgroups of $\mathbb{R}^n$ and have been widely used in cryptology. The *shortest vector problem*(SVP) refers the question to find a shortest non-zero vector in a given lattice, which is one of the most famous and widely studied computational problems on lattices.

It is well known that SVP is NP-hard under random reductions [2], so no polynomial time exact algorithms for it are expected to exist. Up to now, only approximation algorithms, such as [7,8,13,25], are efficient and all known exact algorithms are proven to cost exponential time. However, almost all known approximation algorithms (such as [8,25]) invoke some exact algorithm for solving SVP on some low dimensional lattices to improve the quantity of their outputs. Therefore, it is important to know how fast the best exact algorithm can be. What's more, algorithms for SVP play a very important role in cryptanalysis (see [19] for a survey). For example, nearly all knapsack-based public-key cryptosystems have been broken with a lattice algorithm (see [1,14,27]) and many lattice-based public-key cryptosystems can be broken by solving some

SVP, including the famous NTRU [10]. Hence, better exact algorithm for SVP can also help us to know the security of these lattice-based public-key cryptosystems better, and choose more appropriate parameters for these cryptosystems.

The exact algorithms for SVP can be classified into two classes by now: deterministic algorithms and randomized sieve algorithms.

The first deterministic algorithm to find the shortest vector in a given lattice was proposed by Fincke, Pohst [5,6] and Kannan [11], by enumerating all lattice vectors shorter than a prescribed bound. If the input is an LLL-reduced basis, the running time is $2^{O(n^2)}$ polynomial-time operations. Kannan [11] also showed the running time can reach $2^{O(nlogn)}$ polynomial-time operations by choosing a suitable preprocessing algorithm. Schnorr and Euchner [26] presented a zig-zag strategy for enumerating the lattice vectors to make the algorithm have a better performance in practice. In 2010, Gama, Nguyen and Regev [9] introduced an extreme pruning technique and improved the running time in both theory and practice. All enumeration algorithms above require a polynomial space complexity. Another deterministic algorithm for SVP was proposed by Micciancio and Voulgaris [15] in 2010. Different from the previous algorithms, it is based on Voronoi cell computation and is the first deterministic single exponential time exact algorithm for SVP. The time complexity is $2^{2n+o(n)}$ polynomial-time operations. One drawback of the algorithm is that its space requirement is not polynomial but $2^{O(n)}$.

The randomized sieve algorithm was discovered by Ajtai, Kumar and Sivakumar (AKS) [3] in 2001. The running time and space requirement were proven to be $2^{O(n)}$. Regev's alternative analysis [22] showed that the hidden constant in $O(n)$ was at most 16, and it was further decreased to 5.9 by Nguyen and Vidick [20]. Blömer and Naewe [4] generalized the results of AKS to l_p norms. Micciancio and Voulgaris [16] presented a provable sieving variant called the ListSieve algorithm, whose running time is $2^{3.199n+o(n)}$ polynomial-time operations and space requirement is $2^{1.325n+o(n)}$ polynomially many bits. Subsequently, Pujol and Stehlé [21] improved the theoretical bound of the ListSieve algorithm to running time $2^{2.465n+o(n)}$ and space $2^{1.233n+o(n)}$ by introducing the birthday attack strategy. In the same work [16], Micciancio and Voulgaris also presented a heuristic variant of the ListSieve algorithm, called the GaussSieve algorithm. However, no upper bound on the running time of the GaussSieve Algorithm is currently known and the space requirement is provably bounded by $2^{0.41n}$. In [23], Schneider analyzed the GaussSieve algorithm and showed its strengths and weakness. What's more, a parallel implementation of the GaussSieve algorithm was presented by Milde and Schneider [17]. Recently, Schneider [24] presented an IdealListSieve algorithm to improve the ListSieve algorithm for the shortest vector problem in ideal lattices and the practical speed up is linear in the degree of the field polynomial. He also proposed a variant of the heuristic GaussSieve algorithm for ideal lattice with the same speedup.

To give a correct analysis of its complexity, the AKS algorithm involves some perturbations. However, getting rid of the perturbations, Nguyen and Vidick [20] proposed the first heuristic variant of the AKS algorithm, which in

practice performs better and can solve SVP up to dimension 50. Its running time was proven to be $2^{0.415n+o(n)}$ polynomial-time operations under some nature heuristic assumption of uniform distribution of the sieved lattice vectors. By introducing a two-level technique, Wang *et al.* [30] gave an algorithm (WLTB) to improve the Nguyen-Vidick algorithm. Under a similar assumption of the distribution of sieved lattice vectors, the WLTB algorithm has the best theoretical time complexity so far, that is, $2^{0.3836n+o(n)}$. Both the heuristic assumptions can be supported by the experimental results on low dimensional lattices.

Our Contribution. Observing that the WLTB algorithm involves some data structure like skip list to reduce the time complexity, we present a three-level sieve algorithm in this paper. To estimate the complexity of the algorithm, it needs to compute the volume of some irregular spherical cap, which is a very complicated and tough work. By involving a smart technique, we simplify the complicated computation and prove that the optimal time complexity is $2^{0.3778n+o(n)}$ polynomial-time operations and the corresponding space complexity is $2^{0.2833n+o(n)}$ polynomially many bits under a similar natural heuristic assumption.

Table 1 summarizes the complexities of the heuristic variants of AKS algorithm and the GaussSieve algorithm. It can be seen that the latter two algorithms employ the time-memory tradeoffs that decrease the running time complexity at the cost of space complexity.

Table 1. Complexities of some heuristic algorithms for SVP

Algorithm	Time complexity	Space complexity
GaussSieve algorithm	-	$2^{0.41n+o(n)}$
Nguyen-Vidick algorithm	$2^{0.415n+o(n)}$	$2^{0.2075n+o(n)}$
WLTB algorithm	$2^{0.3836n+o(n)}$	$2^{0.2557n+o(n)}$
Our three-level algorithm	$2^{0.3778n+o(n)}$	$2^{0.2883n+o(n)}$

A natural question is whether we can improve the time complexity by four-level or higher-level algorithm. It may have a positive answer. However, by our work, it seems that the improvements get smaller and smaller, whereas the analysis of the complexity becomes more and more difficult when the number of levels increases.

Road Map. The rest of the paper is organized as follows. In Sect. 2 we provide some notations and preliminaries. We present our three-level sieve algorithm and the detailed analysis of its complexity in Sect. 3. Some experimental results are described in Sect. 4. Finally, Sect. 5 gives a short conclusion.

2   Notations and Preliminaries

Notations. Bold lower-case letters are used to denote vectors in $\mathbb{R}^n$. Denote by v_i the i-th entry of a vector $\boldsymbol{v}$. Let $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ be the Euclidean norm and

inner product of $\mathbb{R}^n$. Matrices are written as bold capital letters and the i-th column vector of a matrix $\boldsymbol{B}$ is denoted by $\boldsymbol{b}_i$.

Let $B_n(\boldsymbol{x}, R) = \{\boldsymbol{y} \in \mathbb{R}^n \mid \|\boldsymbol{y} - \boldsymbol{x}\| \leq R\}$ be the n-dimensional ball centered at $\boldsymbol{x}$ with radius R. Let $B_n(R) = B_n(\mathbf{O}, R)$. Let $C_n(\gamma, R) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \gamma R \leq \|\boldsymbol{x}\| \leq R\}$ be a spherical shell in $B_n(R)$, and $S^n = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\| = 1\}$ be the unit sphere in $\mathbb{R}^n$. Denote by $|S^n|$ the area of S^n.

2.1   Lattices

Let $\boldsymbol{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n\} \subset \mathbb{R}^m$ be a set of n linearly independent vectors. The lattice $\mathcal{L}$ generated by the basis $\boldsymbol{B}$ is defined as $\mathcal{L}(\boldsymbol{B}) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$. n is called the rank of the lattice. Denote by $\lambda_1(\mathcal{L})$ the norm of a shortest non-zero vector of $\mathcal{L}$.

2.2   The Basic Framework of Some Heuristic Sieve Algorithms

The Nguyen-Vidick algorithm and the WLTB algorithm have a common basic framework, which can be described as Algorithm 1 [30].

Algorithm 1. Finding short lattice vectors based on sieving

Input: An LLL-reduced basis $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]$ of a lattice $\mathcal{L}$, sieve factors
         and a number N.
Output: A short non-zero vector of $\mathcal{L}$.
1: $S' \leftarrow \emptyset$
2: **for** $j = 1$ to N **do**
3:    $S' \leftarrow S' \cup$ sampling($\boldsymbol{B}$) using Klein's algorithm [12]
4: **end for**
5: Remove all zero vectors from S'
6: **Repeat**
7:    $S \leftarrow S'$
8:    $S' \leftarrow$ sieve(S, sieve factors) using **Sieve Algorithm**
9:    Remove all zero vectors from S'
10: **until** $S' = \emptyset$
11: Compute $\boldsymbol{v}_0 \in S$ such that $\|\boldsymbol{v}_0\| = \min\{\|\boldsymbol{v}\|, \boldsymbol{v} \in S\}$
12: **Return** $\boldsymbol{v}_0$

In general the **Sieve Algorithm** in Line 8 will output a set S' of shorter lattice vectors than those in S. When we repeat the sieve process enough times, a shortest vector is expected to be found.

Denote by R' (*resp.* R) the maximum length of those vectors in S' (*resp.* S). To find S', the sieve algorithm usually tries to find a set C of lattice vectors in S such that the balls centered at these vectors with radius R' can cover all the lattice points in some spherical shell $C_n(\gamma, R)$. By subtracting the corresponding center from every lattice point in every ball, shorter lattice vectors will be obtained, which form the set S'.

Different ways to find C lead to different algorithms. Roughly speaking,

– The Nguyen-Vidick algorithm checks every lattice point in S' sequentially to decide whether it is also in some existing ball or it is a new vector in C (see Fig. 1 for a geometric description).
– The WLTB algorithm involves a two-level strategy, that is, the big-ball-level and the small-ball-level. It first covers the spherical shell with big balls centered at some lattice vectors, then covers the intersection of every big ball and $C_n(\gamma, R)$ with small balls centered at some lattice points in the intersection. The centers of the small balls form C. It can be shown that it is faster to decide whether a lattice vector is in C or not. We first check whether it is in some big ball or not. If not, it must be a new point in C. If so, we just check whether it is in some small ball in the big ball it belongs to, regardless of those small balls of the other big balls (see Fig. 2 for a geometric description).

For either the Nguyen-Vidick algorithm or the WLTB algorithm, to analyze its complexity needs a natural assumption below.

Heuristic Assumption 1: At any stage in Algorithm 1, the lattice vectors in $S' \cap C_n(\gamma, R)$ are uniformly distributed in $C_n(\gamma, R)$.

(a)   (b)
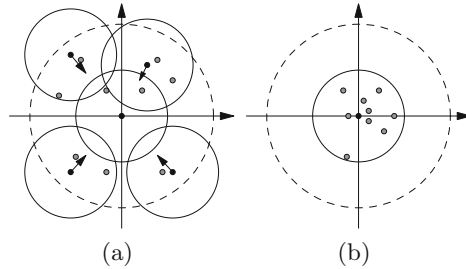
Fig. 1. Geometric description of Nguyen-Vidick's sieve algorithm

(a)   (b)   (c)
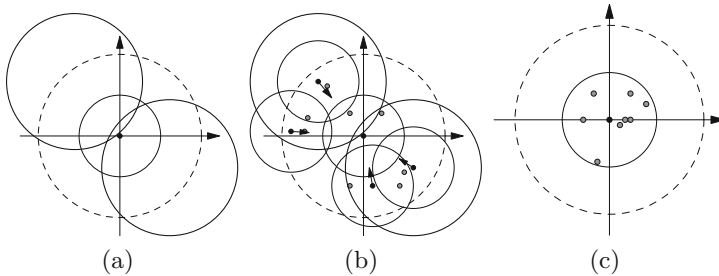
Fig. 2. Geometric description of WLTB's sieve algorithm

3   A Three-Level Sieve Algorithm

3.1   Description of the Three-Level Sieve Algorithm

Different from the two-level algorithm, our algorithm involves a medium-ball-level. Simply speaking, the algorithm first covers the spherical shell with big balls, then covers every big ball with medium balls, and at last covers every medium ball with small balls. Algorithm 2 gives a detailed description of the three-level sieve algorithm.

Algorithm 2. A three-level sieve algorithm

Input: A subset $S \subseteq B_n(R)$ of vectors in a lattice $\mathcal{L}$ where $R \leftarrow \max_{\boldsymbol{v} \in S} \|\boldsymbol{v}\|$
    and sieve factors $0.88 < \gamma_3 < 1 < \gamma_2 < \gamma_1 < \sqrt{2}\gamma_3$.
Output: A subset $S' \subseteq B_n(\gamma_3 R) \cap \mathcal{L}$.
1: $S' \leftarrow \emptyset, C_1 \leftarrow \emptyset$.
2: **for** $\boldsymbol{v} \in S$ **do**
3:    **if** $\|\boldsymbol{v}\| \leq \gamma_3 R$ **then**
4:        $S' \leftarrow S' \cup \{\boldsymbol{v}\}$
5:    **else**
6:        **if** $\exists \, \boldsymbol{c}_1 \in C_1, \|\boldsymbol{v} - \boldsymbol{c}_1\| \leq \gamma_1 R$ **then**
7:            **if** $\exists \, \boldsymbol{c}_2 \in C_2^{\boldsymbol{c}_1}, \|\boldsymbol{v} - \boldsymbol{c}_2\| \leq \gamma_2 R$ **then** $\backslash C_2^{\boldsymbol{c}_1}$ is initialized as $\emptyset \backslash$
8:                **if** $\exists \, \boldsymbol{c}_3 \in C_3^{\boldsymbol{c}_1, \boldsymbol{c}_2}, \|\boldsymbol{v} - \boldsymbol{c}_3\| \leq \gamma_3 R$ **then** $\backslash C_3^{\boldsymbol{c}_1, \boldsymbol{c}_2}$ is initialized as $\emptyset \backslash$
9:                    $S' \leftarrow S' \cup \{\boldsymbol{v} - \boldsymbol{c}_3\}$
10:               **else**
11:                   $C_3^{\boldsymbol{c}_1, \boldsymbol{c}_2} \leftarrow C_3^{\boldsymbol{c}_1, \boldsymbol{c}_2} \cup \{\boldsymbol{v}\}$ \ centers of small balls \
12:               **end if**
13:           **else**
14:               $C_2^{\boldsymbol{c}_1} \leftarrow C_2^{\boldsymbol{c}_1} \cup \{\boldsymbol{v}\}$ \ centers of medius balls \
15:           **end if**
16:       **else**
17:           $C_1 \leftarrow C_1 \cup \{\boldsymbol{v}\}$ \ centers of big balls \
18:       **end if**
19:    **end if**
20: **end for**
21: **return** S'

In Algorithm 2, $0.88 < \gamma_3 < 1 < \gamma_2 < \gamma_1 < \sqrt{2}\gamma_3$. The set C_1 is the collection of centers of big balls with radius $\gamma_1 R$ in the first level. For any $\boldsymbol{c}_1 \in C_1$, $C_2^{\boldsymbol{c}_1}$ is the set of centers of medium balls with radius $\gamma_2 R$ that cover the big spherical cap $B_n(\boldsymbol{c}_1, \gamma_1 R) \cap C_n(\gamma_3, R)$. It is clear that the elements of $C_2^{\boldsymbol{c}_1}$ are chosen from $B_n(\boldsymbol{c}_1, \gamma_1 R) \cap C_n(\gamma_3, R)$. For $\boldsymbol{c}_1 \in C_1, \boldsymbol{c}_2 \in C_2^{\boldsymbol{c}_1}$, $C_3^{\boldsymbol{c}_1, \boldsymbol{c}_2}$ is the set of centers of small balls with radius $\gamma_3 R$ that cover the small spherical cap $B_n(\boldsymbol{c}_2, \gamma_2 R) \cap B_n(\boldsymbol{c}_1, \gamma_1 R) \cap C_n(\gamma_3, R)$. Also the elements of $C_3^{\boldsymbol{c}_1, \boldsymbol{c}_2}$ are chosen from the small spherical cap.

3.2   Complexity of the Algorithm

Denote by N_1, N_2 and N_3 the corresponding upper bound on the expected number of lattice points in C_1, $C_2^{c_1}$ (for any $c_1 \in C_1$) and $C_3^{c_1,c_2}$ (for any $c_1 \in C_1, c_2 \in C_2^{c_1}$).

The Space Complexity. Notice that the total number of big, medium and small balls can be bounded by N_1, $N_1 N_2$ and $N_1 N_2 N_3$ respectively. As in [20] and [30], if we sample $\mathrm{poly}(n) N_1 N_2 N_3$ vectors, after a polynomial number of iterations in Algorithm 1, it is expected that a shortest non-zero lattice vector can be obtained with the left vectors. So the space complexity is bounded by $O(N_1 N_2 N_3)$.

The Time Complexity. The initial size of S is $\mathrm{poly}(n) N_1 N_2 N_3$. In each iteration in Algorithm 1, steps 3–19 in Algorithm 2 repeat $\mathrm{poly}(n) N_1 N_2 N_3$ times, and in each repeat, at most $N_1 + N_2 + N_3$ comparisons are needed. Therefore, the total time complexity can be bounded by $O(N_1 N_2 N_3 (N_1 + N_2 + N_3))$ polynomial-time operations.

   We next give the estimation of N_1, N_2 and N_3. Without loss of generality, we restrict $R = 1$ and let $C_n(\gamma) = C_n(\gamma, 1) = \{ x \in \mathbb{R}^n \mid \gamma R \le \| x \| \le 1 \}$ through our proofs for simplicity.

The Upper Bound of N_1. Nguyen and Vidick [20] first gave a proof of the upper bound N_1, and a more refined proof was given by Wang *et al.* [30].

Theorem 1. (Wang *et al.* [30]) *Let n be a non-negative integer, N be an integer and $0.88 < \gamma_3 < 1 < \gamma_1 < \sqrt{2}\gamma_3$. Let*

$$N_1 = c_{\mathcal{H}_1}^n \lceil 3\sqrt{2\pi} n^{\frac{3}{2}} \rceil,$$

where $c_{\mathcal{H}_1} = 1/(\gamma_1 \sqrt{1 - \frac{\gamma_1^2}{4}})$ and S a subset of $C_n(\gamma_3 R)$ of cardinality N whose points are picked independently at random with uniform distribution. If $N_1 < N < 2^n$, then for any subset $C \subseteq S$ of size at least N_1 whose points are picked independently at random with uniform distribution, with overwhelming probability, for all $v \in S$, there exists a $c \in C$ such that $\| v - c \| \le \gamma_1 R$.

The Upper Bound of N_2. Let

- $\Omega_n(\gamma_1)$ be the fraction of $C_n(\gamma_3)$ that is covered by a ball of radius γ_1 centered at a point of $C_n(\gamma_3)$,
- $\Gamma_n(\gamma_1, \gamma_2)$ be the fraction of $C_n(\gamma_3)$ covered by a big spherical cap $B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$,
- $\Omega_n(\gamma_1, \gamma_2)$ be the fraction of $B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$ covered by $B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$, where $c_2 \in C_2^{c_1}, c_1 \in C_1$.

Clearly, $\Omega_n(\gamma_1, \gamma_2) = \frac{\Gamma_n(\gamma_1, \gamma_2)}{\Omega_n(\gamma_1)}$. To compute N_2, we need the minimal value of $\Omega_n(\gamma_1, \gamma_2)$. We estimate $\Omega_n(\gamma_1)$ and $\Gamma_n(\gamma_1, \gamma_2)$ respectively.

Lemma 1. (Wang *et al.* [30]) *Let* $0.88 < \gamma_3 < 1 < \gamma_1 < \sqrt{2}\gamma_3$, *then*

$$\frac{1}{3\sqrt{2\pi n}}\frac{(\sin\theta_2)^{n-1}}{\cos\theta_2} < \Omega_n(\gamma_1) < \frac{1}{\sqrt{2\pi(n-1)}}\frac{(\sin\theta_1)^{n-1}}{\cos\theta_1},$$

where $\theta_1 = \arccos(1 - \frac{\gamma_1^2}{2\gamma_3^2}), \theta_2 = \arccos(1 - \frac{\gamma_1^2}{2})$.

Note that the proportion $\Gamma_n(\gamma_1, \gamma_2)$ is different from that of Lemma 4 in [30], as the radius of $B_n(c_2, \gamma_2)$ is larger than the inside radius of the shell $C_n(\gamma_3)$. Thus, it leads to the slightly different bounds of $\Gamma_n(\gamma_1, \gamma_2)$ from that of Lemma 4 in [30]. If c_2 lies on the sphere of a big ball $B_n(c_1, \gamma_1)$, the fraction $\Gamma_n(\gamma_1, \gamma_2)$ is minimal. Lemma 2 gives the minimal and maximal value of $\Gamma_n(\gamma_1, \gamma_2)$ when c_2 lies on the sphere of a big ball $B_n(c_1, \gamma_1)$.

Lemma 2. *Let* $0.88 < \gamma_3 < 1 < \gamma_2 < \gamma_1 < \sqrt{2}\gamma_3$, *where* γ_3 *is very close to* 1. *Then*

$$\frac{cd_{\min}^{n-2}}{2\pi n} \le \Gamma_n(\gamma_1, \gamma_2) \le \frac{c'd_{\max}^{n-2}}{2\pi},$$

where $d_{\max} = \sqrt{1 - \left(\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3}\right)^2 - \left(\frac{1}{c_{\mathcal{H}_2}}\left(\frac{\gamma_3^2 + 1 - \gamma_2^2}{2} - \frac{(2\gamma_3^2 - \gamma_1^2)(\gamma_3^2 - \gamma_1^2 + 1)}{4\gamma_3^2}\right)\right)^2}$, $d_{\min}$

$= \gamma_2\sqrt{1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}}$, $c_{\mathcal{H}_1} = 1/(\gamma_1\sqrt{1 - \frac{\gamma_1^2}{4}})$, $c_{\mathcal{H}_2} = \frac{\gamma_1}{\gamma_3}\sqrt{1 - \frac{\gamma_1^2}{4\gamma_3^2}}$, c *and* c' *are constants unrelated to* n.
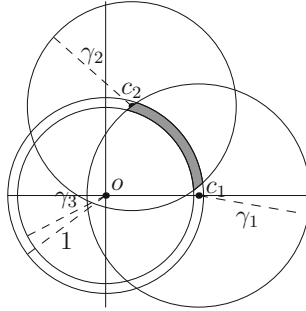
Fig. 3. The region of $B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$.

Proof. Note that γ_3 is very close to 1. We just consider the proportion on the sphere covering as in [30].

Without loss of generality, we assume the center of $B_n(c_1, \gamma_1)$ is $c_1 = (\alpha_1, 0, \ldots, 0)$, and the center of $B_n(c_2, \gamma_2)$ is $c_2 = (\beta_1, \beta_2, 0 \ldots, 0)$, where $\alpha, \beta_1, \beta_2 > 0$. The spherical cap $B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$ is

$$\begin{cases} x_1^2 + x_2^2 + \ldots + x_n^2 = 1 \\ (x_1 - \alpha_1)^2 + x_2^2 + \ldots + x_n^2 < \gamma_1^2 \\ (x_1 - \beta_1)^2 + (x_2 - \beta_2)^2 + \ldots + x_n^2 < \gamma_2^2 \end{cases}$$

where $\gamma_3 \leq \alpha_1 \leq 1$, $(\beta_1 - \alpha_1)^2 + \beta_2^2 = \gamma_1^2$ and $\gamma_3^2 \leq \beta_1^2 + \beta_2^2 \leq 1$. The region is as the shadow of the Fig. 3. Denote by Q the volume of the region. By projecting the target region to the hyperplane orthogonal to x_1 and by sphere coordinate transformation (for details see the proof of Lemma 4 in [30]), we get

$$\frac{cd^{n-2}}{2\pi n} \leq \Gamma_n(\gamma_1, \gamma_2) = \frac{Q}{|S^n|} \leq \frac{c'd^{n-2}}{2\pi}$$

where $d = \sqrt{1 - \left(\frac{\alpha_1^2 - \gamma_1^2 + 1}{2\alpha_1}\right)^2 - \left(\frac{1}{\beta_2}\left(\frac{\beta_1^2 + \beta_2^2 + 1 - \gamma_2^2}{2} - \beta_1\frac{\alpha_1^2 - \gamma_1^2 + 1}{2\alpha_1}\right)\right)^2}$ and c, c' are constants unrelated to n. Let $\alpha_2 = \sqrt{\beta_1^2 + \beta_2^2}$. From the equation $(\beta_1 - \alpha_1)^2 + \beta_2^2 = \gamma_1^2$, we obtain

$$\beta_1 = \frac{\alpha_2^2 + \alpha_1^2 - \gamma_1^2}{2\alpha_1}, \beta_2 = \sqrt{\alpha_2^2 - \left(\frac{\alpha_2^2 + \alpha_1^2 - \gamma_1^2}{2\alpha_1}\right)^2}.$$

Therefore, d can be regarded as a function with respect to α_1, α_2, where $\gamma_3 \leq \alpha_1 \leq 1, \gamma_3 \leq \alpha_2 \leq 1$. Since $0.88 < \gamma_3 < 1 < \gamma_2 < \gamma_1 < \sqrt{2}\gamma_3$, it can be proven that d decreases with α_1, α_2 increasing. Then $d_{\min}$ can be obtained by letting $\alpha_1 = 1, \alpha_2 = 1$ and $d_{\max}$ can be obtained by letting $\alpha_1 = \gamma_3, \alpha_2 = \gamma_3$. Hence, the lemma follows.

Theorem 2. *Let n be a non-negative integer, N be an integer and $0.88 < \gamma_3 < 1 < \gamma_2 < \gamma_1 < \sqrt{2}\gamma_3$, where γ_3 is very close to 1. Let*

$$N_2 = c_2\left(\frac{c_{\mathcal{H}_2}}{d_{\min}}\right)^n \lceil n^{\frac{3}{2}}\rceil,$$

where $c_{\mathcal{H}_2} = \frac{\gamma_1}{\gamma_3}\sqrt{1 - \frac{\gamma_1^2}{4\gamma_3^2}}, d_{\min} = \gamma_2\sqrt{1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}}, c_{\mathcal{H}_1} = 1/(\gamma_1\sqrt{1 - \frac{\gamma_1^2}{4}})$, and c_2 is a positive constant unrelated to n. Let S be a subset of $C_n(\gamma_3 R) \cap B_n(\boldsymbol{c}_1, \gamma_1 R) \cap B_n(\boldsymbol{c}_2, \gamma_2 R)$ of cardinality N whose points are picked independently at random with uniform distribution. If $N_2 < N < 2^n$, then for any subset $C \subseteq S$ of size at least N_2 whose points are picked independently at random with uniform distribution, with overwhelming probability, for all $\boldsymbol{v} \in S$, there exists a $\boldsymbol{c} \in C$ such that $\|\boldsymbol{v} - \boldsymbol{c}\| \leq \gamma_2 R$.

Proof. Combining Lemmas 1 and 2, we have $\Omega_n(\gamma_1, \gamma_2) = \frac{\Gamma_n(\gamma_1, \gamma_2)}{\Omega_n(\gamma_1)} \geq \frac{c}{\sqrt{2\pi n}} \cdot \left(1 - \frac{\gamma_1^2}{2\gamma_2^2}\right)\left(\frac{d_{\min}}{c_{\mathcal{H}_2}}\right)^n$. The expected fraction of $B_n(\boldsymbol{c}_1, \gamma_1) \cap C_n(\gamma_3)$ that is not covered by N_2 balls of radius γ_2 centered at randomly chosen points of $B_n(\boldsymbol{c}_1, \gamma_1) \cap C_n(\gamma_3)$ is $(1 - \Omega_n(\gamma_1, \gamma_2))^{N_2}$. So,

$$N_2\log(1 - \Omega_n(\gamma_1, \gamma_2)) \leq N_2(-\Omega_n(\gamma_1, \gamma_2))$$
$$< c_2 n^{3/2}\left(\frac{c_{\mathcal{H}_2}}{d_{\min}}\right)^n \cdot \frac{1}{c_2\sqrt{n}}\left(\frac{d_{\min}}{c_{\mathcal{H}_2}}\right)^n \leq -n < -\log N.$$

which implies $(1 - \Omega_n(\gamma_1, \gamma_2))^{N_2} < e^{-n} < \frac{1}{N}$. The expected number of uncovered points is smaller than 1. It means that any point in $B_n(\boldsymbol{c}_1, \gamma_1) \cap C_n(\gamma_3)$ is

covered by a ball centered at a vector in $B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$ with radius γ_2 with probability $1 - e^{-n}$.

The Upper Bound of N_3. Let

- $\Gamma_n(\gamma_1, \gamma_2, \gamma_3)$ be the fraction of $C_n(\gamma_3)$ that is covered by a small spherical cap $B_n(c_3, \gamma_3) \cap B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$,
- $\Omega_n(\gamma_1, \gamma_2, \gamma_3)$ the fraction of $B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$ covered by $B_n(c_3, \gamma_3) \cap B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$, where $c_3 \in C_3^{c_1, c_2}, c_2 \in C_2^{c_1}$, $c_1 \in C_1$.

Clearly, $\Omega_n(\gamma_1, \gamma_2, \gamma_3) = \frac{\Gamma_n(\gamma_1, \gamma_2, \gamma_3)}{\Gamma_n(\gamma_1, \gamma_2)}$. To estimate N_3, we need to compute the lower bound of $\Omega_n(\gamma_1, \gamma_2, \gamma_3)$. To obtain the lower bound of $\Gamma_n(\gamma_1, \gamma_2, \gamma_3)$, we need to compute the volume of some irregular convex region, which is very complicated. However, using the inscribed triangle of the region, we get a reasonable lower bound of the volume successfully.

Lemma 3. *Let* $0.88 < \gamma_3 < 1 < \gamma_2 < \gamma_1 < \sqrt{2}\gamma_3$, *where* γ_3 *is very close to* 1. *We have*

$$\Gamma_n(\gamma_1, \gamma_2, \gamma_3) \geq \frac{c'' r_{\min}^{n-3}}{2\pi^{3/2} n^2},$$

where $r_{\min} = \sqrt{c_{\mathcal{H}_3} - \left(1 - \frac{\gamma_3^2}{2c_{\mathcal{H}_3}}\right)^2}$, $c_{\mathcal{H}_3} = \gamma_2^2 \left(1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}\right)$, c'' *is a constant unrelated to* n.

Proof. We consider the proportion on the sphere covering. W.l.o.g., we assume the centers of $B_n(c_1, \gamma_1), B_n(c_2, \gamma_2), B_n(c_3, \gamma_3)$ are, respectively,

$$c_1 = (\alpha_1, 0, \ldots, 0), \ \alpha_1 > 0,$$
$$c_2 = (\beta_1, \beta_2, 0 \ldots, 0), \ \beta_1, \beta_2 > 0,$$
$$c_3 = (\delta_1, \delta_2, \delta_3, 0 \ldots, 0), \ \delta_1, \delta_2, \delta_3 > 0.$$

The spherical cap $B_n(c_3, \gamma_3) \cap B_n(c_2, \gamma_2) \cap B_n(c_1, \gamma_1) \cap C_n(\gamma_3)$ is

$$\begin{cases} x_1^2 + x_2^2 + \ldots + x_n^2 = 1 & (E_1) \\ (x_1 - \alpha_1)^2 + x_2^2 + \ldots + x_n^2 < \gamma_1^2 & (E_2) \\ (x_1 - \beta_1)^2 + (x_2 - \beta_2)^2 + x_3^2 + \ldots + x_n^2 < \gamma_2^2 & (E_3) \\ (x_1 - \delta_1)^2 + (x_2 - \delta_2)^2 + (x_3 - \delta_3)^2 + \ldots + x_n^2 < \gamma_3^2 & (E_4) \end{cases}$$

where $\gamma_3 \leq \alpha_1 \leq 1, \gamma_3^2 \leq \beta_1^2 + \beta_2^2 \leq 1, (\beta_1 - \alpha_1)^2 + \beta_2^2 = \gamma_1^2, \gamma_3^2 \leq \delta_1^2 + \delta_2^2 + \delta_3^2 \leq 1, (\delta_1 - \alpha_1)^2 + \delta_2^2 + \delta_3^2 = \gamma_1^2, (\delta_1 - \beta_1)^2 + (\delta_2 - \beta_2)^2 + \delta_3^2 = \gamma_2^2$.

Denote by Q the volume of the region, and project the target region to the hyperplane orthogonal to x_1. Denote by D the projection region. Therefore, the volume of the target region is

$$Q = \iint \cdots \int_D \sqrt{1 + \sum_{i=2}^{n} \left(\frac{\partial x_1}{\partial x_i}\right)^2} \, dx_2 dx_3 \cdots dx_n = \iint \cdots \int_D \frac{dx_2 dx_3 \cdots dx_n}{\sqrt{1 - \sum_{i=2}^{n} x_i^2}}.$$

Now we determine the projection region D. To simplify the expression, we let $\alpha_2 = \sqrt{\beta_1^2 + \beta_2^2}, \alpha_3 = \sqrt{\delta_1^2 + \delta_2^2 + \delta_3^2}, a = \frac{\alpha_1^2 + 1 - \gamma_1^2}{2\alpha_1}, b = \frac{\alpha_2^2 + 1 - \gamma_2^2}{2}, f = \frac{\alpha_3^2 + 1 - \gamma_3^2}{2}$. From the equations $(\beta_1 - \alpha_1)^2 + \beta_2^2 = \gamma_1^2, (\delta_1 - \alpha_1)^2 + \delta_2^2 + \delta_3^2 = \gamma_1^2, (\delta_1 - \beta_1)^2 + (\delta_2 - \beta_2)^2 + \delta_3^2 = \gamma_2^2$, it is easy to write $\beta_1, \beta_2, \delta_1, \delta_2, \delta_3$ as the expressions of $\alpha_i, \gamma_i, i = 1, 2, 3$, i.e.,

$$\beta_1 = \frac{\alpha_1^2 + \alpha_2^2 - \gamma_1^2}{2\alpha_1}, \quad \beta_2 = \sqrt{\alpha_2^2 - \left(\frac{\alpha_2^2 + \alpha_1^2 - \gamma_1^2}{2\alpha_1}\right)^2},$$

$$\delta_1 = \frac{\alpha_1^2 + \alpha_3^2 - \gamma_1^2}{2\alpha_1}, \quad \delta_2 = \frac{\alpha_2^2 + \alpha_3^2 - \gamma_2^2 - \frac{(\alpha_1^2 + \alpha_2^2 - \gamma_1^2)(\alpha_1^2 + \alpha_3^2 - \gamma_1^2)}{2\alpha_1^2}}{2\sqrt{\alpha_2^2 - \left(\frac{\alpha_1^2 + \alpha_2^2 - \gamma_1^2}{2\alpha_1}\right)^2}},$$

$$\delta_3 = \left(\alpha_3^2 - \left(\frac{\alpha_1^2 + \alpha_3^2 - \gamma_1^2}{2\alpha_1}\right)^2 - \frac{\left(\alpha_2^2 + \alpha_3^2 - \gamma_2^2 - \frac{(\alpha_1^2 + \alpha_2^2 - \gamma_1^2)(\alpha_1^2 + \alpha_3^2 - \gamma_1^2)}{2\alpha_1^2}\right)^2}{4\left(\alpha_2^2 - \left(\frac{\alpha_1^2 + \alpha_2^2 - \gamma_1^2}{2\alpha_1}\right)^2\right)}\right)^{\frac{1}{2}}.$$

We project the intersection of equation (E_1) and (E_i) to the hyperplane orthogonal to x_1 and suppose the projection region is $D_{i-1}, i = 2, 3, 4$. Then $D = D_1 \cap D_2 \cap D_3$, where

$$D_1 = \{(x_2, x_3, \ldots, x_n) \in \mathbb{R}^{n-1} | x_2^2 + x_3^2 + \cdots + x_n^2 < 1 - a^2\}.$$

$$D_2^1 = \left\{(x_2, x_3, \ldots, x_n) \in \mathbb{R}^{n-1} | x_2^2 + x_3^2 + \cdots + x_n^2 < 1 - \left(\frac{b - \beta_2 x_2}{\beta_1}\right)^2, x_2 < \frac{b}{\beta_2}\right\},$$

$$D_2^2 = \left\{(x_2, x_3, \ldots, x_n) \in \mathbb{R}^{n-1} | x_2^2 + x_3^2 + \cdots + x_n^2 < 1, x_2 \geq \frac{b}{\beta_2}\right\},$$

$$D_2 = D_2^1 \cup D_2^2.$$

$$D_3^1 = \left\{(x_2, x_3, \ldots, x_n) \in \mathbb{R}^{n-1} | x_2^2 + x_3^2 + \cdots + x_n^2 < 1 - \left(\frac{f - \delta_2 x_2 - \delta_2 x_3}{\delta_1}\right)^2,\right.$$
$$\left. f - \delta_2 x_2 - \delta_2 x_3 > 0\right\},$$

$$D_3^2 = \{(x_2, x_3, \ldots, x_n) \in \mathbb{R}^{n-1} | x_2^2 + x_3^2 + \cdots + x_n^2 < 1, f - \delta_2 x_2 - \delta_2 x_3 \leq 0\},$$

$$D_3 = D_3^1 \cup D_3^2.$$

The region of (x_2, x_3) for D is the shadow of Fig. 4, and that of $(x_4, \ldots, x_n)$ is an $(n-3)$-dimensional ball with radius $r = \sqrt{1 - a^2 - \left(\frac{b - a\beta_1}{\beta_2}\right)^2 - \left(\frac{f - \delta_1 a - \delta_2 \frac{b - a\beta_1}{\beta_2}}{\delta_3}\right)^2}$.
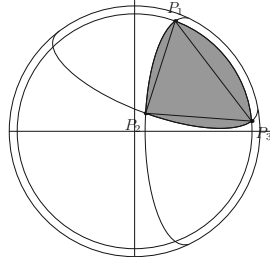
Fig. 4. The region of (x_2, x_3) for D.

For $(x_4, \ldots, x_n)$, we adopt hyper sphere coordinate transformation. Let

$$
\begin{cases}
x_4 = t \cos \varphi_1 \\
x_5 = t \sin \varphi_1 \cos \varphi_2 \\
\quad \vdots \\
x_{n-1} = t \sin \varphi_1 \cdots \sin \varphi_{n-5} \cos \varphi_{n-4} \\
x_n = t \sin \varphi_1 \cdots \sin \varphi_{n-5} \sin \varphi_{n-4}
\end{cases}
$$

where $0 \le t \le r, 0 \le \varphi_k \le \pi, k = 1, \ldots, n-5, 0 \le \varphi_{n-4} \le 2\pi$.

For a fixed t, denote by $D(t)$ the corresponding region of (x_2, x_3) and by $s(t)$ the area of $D(t)$. Let $f(t)$ be the area of triangular $\triangle_{P_1 P_2 P_3}$, then $s(t) \ge f(t)$. Thus,

$$
Q = \int_0^r \int_0^{2\pi} \int_0^\pi \cdots \int_0^\pi t^{n-4} \sin \varphi_{n-5} \cdots \sin^{n-4} \varphi_1 \iint_{D(t)} \frac{dx_2 dx_3}{\sqrt{1 - \sum_{i=2}^n x_i^2}} d\varphi_1 \cdots dt
$$

$$
\ge \int_0^r \int_0^{2\pi} \int_0^\pi \cdots \int_0^\pi t^{n-4} \sin \varphi_{n-5} \cdots \sin^{n-4} \varphi_1 \iint_{D(t)} dx_2 dx_3 d\varphi_1 \cdots d\varphi_{n-4} dt
$$

$$
= 2\pi \int_0^r t^{n-4} s(t) dt \prod_{k=1}^{n-5} \int_0^\pi \sin^k \varphi d\varphi \ge 2\pi \int_0^r t^{n-4} f(t) dt \prod_{k=1}^{n-5} \int_0^\pi \sin^k \varphi d\varphi,
$$

and

$$
\Gamma_n(\gamma_1, \gamma_2, \gamma_3) = \frac{Q}{|S^n|} \ge \frac{2\Gamma(\frac{n+2}{2}) \prod_{k=1}^{n-5} \int_0^\pi \sin^k \varphi d\varphi}{n\pi^{n/2-1}} \int_0^r t^{n-4} f(t) dt. \qquad (1)
$$

We next give the lower bounds of $\frac{2\Gamma(\frac{n+2}{2}) \prod_{k=1}^{n-5} \int_0^\pi \sin^k \varphi d\varphi}{n\pi^{n/2-1}}$ and $\int_0^r t^{n-4} f(t) dt$.

Since $\int_0^\pi \sin^k \varphi d\varphi = \sqrt{\pi} \frac{\Gamma((k+1)/2)}{\Gamma(k/2+1)}$ and $\Gamma(x)$ is increasing when $x > 2$, we obtain

$$\frac{2\Gamma(\frac{n+2}{2})\prod_{k=1}^{n-5}\int_0^\pi \sin^k \varphi d\varphi}{n\pi^{n/2-1}} \geq \frac{2\Gamma(\frac{n+2}{2})}{\pi^{3/2}n\Gamma(\frac{n-3}{2})} \geq \frac{n-3}{2\pi^{3/2}}. \tag{2}$$

For the lower bound of $\int_0^r t^{n-4}f(t)dt$, we first have the coordinate of P_1, P_2, P_3:

$$P_1 = \left(\frac{b-a\beta_1}{\beta_2}, \sqrt{1-a^2 - \left(\frac{b-a\beta_1}{\beta_2}\right)^2 - t^2}\right) \triangleq (a_1, b_1),$$

$$P_2 = \left(k_1(s_1 - \sqrt{t_1 - q_1 t^2}) + k_2, s_1 - \sqrt{t_1 - q_1 t^2}\right) \triangleq (a_2, b_2),$$

$$P_3 = \left(\frac{f - a\delta_1 - \delta_3(s_2 - \sqrt{t_2 - q_2 t^2})}{\delta_2}, s_2 - \sqrt{t_2 - q_2 t^2}\right) \triangleq (a_3, b_3).$$

where

$$h_1 = -\frac{\delta_3\beta_2}{\delta_1\beta_2 - \delta_2\beta_1}, \quad h_2 = \frac{f\beta_2 - b\delta_2}{\delta_1\beta_2 - \delta_2\beta_1}, \quad k_1 = \frac{\delta_3\beta_1}{\delta_1\beta_2 - \delta_2\beta_1}, \quad k_2 = \frac{b\delta_1 - f\beta_1}{\delta_1\beta_2 - \delta_2\beta_1},$$

$$s_1 = -\frac{h_1 h_2 + k_1 k_2}{1 + k_1^2 + h_1^2}, \quad q_1 = \frac{1}{1 + k_1^2 + h_1^2}, \quad t_1 = \frac{1 - h_2^2 - k_2^2}{1 + k_1^2 + h_1^2} + \left(\frac{h_1 h_2 + k_1 k_2}{1 + k_1^2 + h_1^2}\right)^2,$$

$$s_2 = \frac{(f - \delta_1 a)\delta_3}{\delta_2^2 + \delta_3^2}, \quad q_2 = \frac{\delta_2^2}{\delta_2^2 + \delta_3^2}, \quad t_2 = \left(1 - a^2 - \frac{(f - \delta_1 a)^2}{\delta_2^2 + \delta_3^2}\right)\frac{\delta_2^2}{\delta_2^2 + \delta_3^2}.$$

The area of $\triangle_{P_1 P_2 P_3}$ is $f(t) = \frac{1}{2}(a_1 b_2 + a_2 b_3 + a_3 b_1 - a_3 b_2 - a_2 b_1 - a_1 b_3)$. It can be verified that $f(t)$ is decreasing when $t \in [0, r]$ and $f(r) = f'(r) = 0, f''(r) > 0$. We have

$$\int_0^r t^{n-4}f(t)dt \geq \int_0^{r-\frac{r}{n}} t^{n-4}f(t)dt \geq \frac{r^{n-3}}{n-3}(1 - \frac{1}{n})^{n-3}f(r - \frac{r}{n}).$$

Notice that $\left(1 - \frac{1}{n}\right)^{n-3} \geq \left(1 - \frac{1}{n}\right)^n \approx e^{-1}$ when n is sufficiently large, and by Taylor series for $f(r - \frac{r}{n})$, $f(r - \frac{r}{n}) = \Theta(\frac{1}{n^2})$. We have for some constant c'' unrelated to n,

$$\int_0^r t^{n-4}f(t)dt \geq \frac{c'' r^{n-3}}{n^2(n-3)}. \tag{3}$$

Combining (1), (2) and (3), we have $\Gamma_n(\gamma_1, \gamma_2, \gamma_3) \geq \frac{c'' r^{n-3}}{2\pi^{3/2}n^2}$. Now r can be regarded as a function with respect to $\alpha_1, \alpha_2, \alpha_3$, where $\gamma_3 \leq \alpha_1, \alpha_2, \alpha_3 \leq 1$. It can be verified that r decreases with $\alpha_1, \alpha_2, \alpha_3$ increasing. Let $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1$, we get the minimal value of r. $r_{\min} = \sqrt{c_{\mathcal{H}_3} - \left(1 - \frac{\gamma_3^2}{2c_{\mathcal{H}_3}}\right)^2}$, $c_{\mathcal{H}_3} = \gamma_2^2\left(1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}\right)$. So, $\Gamma_n(\gamma_1, \gamma_2, \gamma_3) \geq \frac{c'' r_{\min}^{n-3}}{2\pi^{3/2}n^2}$.

Theorem 3. *Let n be a non-negative integer, N be an integer and $0.88 < \gamma_3 < 1 < \gamma_2 < \gamma_1 < \sqrt{2}\gamma_3$, where γ_3 is very close to 1. Let*

$$N_3 = c_3 n^3 (\frac{d_{\max}}{r_{\min}})^n,$$

where $d_{\max} = \sqrt{1 - \left(\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3}\right)^2 - \left(\frac{1}{c_{\mathcal{H}_2}}\left(\frac{\gamma_3^2 + 1 - \gamma_2^2}{2} - \frac{2\gamma_3^2 - \gamma_1^2}{2\gamma_3}\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3}\right)\right)^2}$, $r_{\min} = \sqrt{c_{\mathcal{H}_3} - \left(1 - \frac{\gamma_3^2}{2c_{\mathcal{H}_3}}\right)^2}$, $c_{\mathcal{H}_1} = \frac{1}{\gamma_1\sqrt{1 - \frac{\gamma_1^2}{4}}}$, $c_{\mathcal{H}_2} = \frac{\gamma_1}{\gamma_3}\sqrt{1 - \frac{\gamma_1^2}{4\gamma_3^2}}$, $c_{\mathcal{H}_3} = \gamma_2^2\left(1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}\right)$,

and c_3 is a positive constant unrelated to n. Let S be a subset of $C_n(\gamma_3 R) \cap B_n(\boldsymbol{c}_1, \gamma_1 R) \cap B_n(\boldsymbol{c}_2, \gamma_2 R) \cap B_n(\boldsymbol{c}_3, \gamma_3 R)$ of cardinality N whose points are picked independently at random with uniform distribution. If $N_3 < N < 2^n$, then for any subset $C \subseteq S$ of size at least N_3 whose points are picked independently at random with uniform distribution, with overwhelming probability, for all $\boldsymbol{v} \in S$, there exists a $\boldsymbol{c} \in C$ such that $\|\boldsymbol{v} - \boldsymbol{c}\| \leq \gamma_3 R$.

Proof. Combining Lemmas 2 and 3, we have

$$\Omega_n(\gamma_1, \gamma_2, \gamma_3) = \frac{\Gamma_n(\gamma_1, \gamma_2, \gamma_3)}{\Gamma_n(\gamma_1, \gamma_2)} \geq \frac{c''}{\sqrt{\pi}n^2}\left(\frac{r_{\min}}{d_{\max}}\right)^n.$$

Let $N_3 = c_3 n^3 (\frac{d_{\max}}{r_{\min}})^n$, the remaining proof is similar to that of Theorem 2.

The Optimal Time Complexity. It can be proved that $N_1 N_2 N_3 (N_1 + N_2 + N_3)$ decreases with γ_3. In fact,

- $N_1 = (\frac{1}{\gamma_1\sqrt{1 - \gamma_1^2/4}})^n \lceil 3\sqrt{2\pi}n^{3/2}\rceil$ is unrelated to γ_3.
- $N_2 = c_2(\frac{c_{\mathcal{H}_2}}{d_{\min}})^n \lceil n^{\frac{3}{2}}\rceil$. Only $c_{\mathcal{H}_2} = \frac{\gamma_1}{\gamma_3}\sqrt{1 - \frac{\gamma_1^2}{4\gamma_3^2}} = \sqrt{1 - (1 - \frac{\gamma_1^2}{2\gamma_3^2})^2}$ is related to γ_3, and it is easy to see that $c_{\mathcal{H}_2}$ decreases with respect to γ_3, which implies that N_2 is a monotonically decreasing function of γ_3.
- $N_3 = c_3 n^3 \left(\frac{\sqrt{1 - \left(\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3}\right)^2 - \left(\frac{1}{c_{\mathcal{H}_2}}\left(\frac{\gamma_3^2 + 1 - \gamma_2^2}{2} - \frac{2\gamma_3^2 - \gamma_1^2}{2\gamma_3}\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3}\right)\right)^2}}{\sqrt{c_{\mathcal{H}_3} - \left(1 - \frac{\gamma_3^2}{2c_{\mathcal{H}_3}}\right)^2}}\right)^n$. First, the denominator of N_3 increases with γ_3, since $c_{\mathcal{H}_3}$ is unrelated to γ_3. By $\gamma_1 > 1$, we have $\left(\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3}\right)' = \frac{\gamma_3^2 + \gamma_1^2 - 1}{2\gamma_3^2} > 0$, and $\left(\frac{\gamma_3^2 + 1 - \gamma_2^2}{2} - \frac{2\gamma_3^2 - \gamma_1^2}{2\gamma_3}\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3}\right)' = \gamma_3 - \frac{2\gamma_3^2 + \gamma_1^2}{2\gamma_3^2}\frac{\gamma_3^2 - \gamma_1^2 + 1}{2\gamma_3} - \frac{2\gamma_3^2 - \gamma_1^2}{2\gamma_3}\frac{\gamma_3^2 + \gamma_1^2 - 1}{2\gamma_3^2} = \frac{\gamma_1^2(\gamma_1^2 - 1)}{2\gamma_3^3} > 0$. Together with $\frac{1}{c_{\mathcal{H}_2}}$ increases with γ_3, then we have the numerator of N_3 decreases with γ_3. Thus, N_3 decreases with respect to γ_3.

Therefore, $N_1 N_2 N_3 (N_1 + N_2 + N_3)$ decreases with γ_3.

Since the expression of the time complexity is complicated, we solve a numerical optimal solution. Take $\gamma_3 = 1$. Let γ_1 go through from 1 to 1.414 by 0.0001 and for a fixed γ_1, let γ_2 go through from 1 to γ_1 by 0.0001, then we can easily find the minimal value of the exponential constant for the running time. Thus, we obtain the numerical optimal time complexity of our three-level sieve algorithm.

Theorem 4. *The optimal time complexity of the algorithm is $2^{0.3778n+o(n)}$ poly-nomial-time operations with $\gamma_3 \to 1, \gamma_1 = 1.1399, \gamma_2 = 1.0677$, and the corresponding space complexity is $2^{0.2833n+o(n)}$ polynomially many bits under Heuristic Assumption 1.*

Remark 1. As in [20], the number of iterations is usually linear in the dimension of lattices. Regardless of the number of iterations, the polynomial factors hidden in the time complexity in NV algorithm and WLTB algorithm are respectively n^3 and $n^{4.5}$. In our three level sieve algorithm, the polynomial parts of N_1, N_2 and N_3 given by Theorem 1, 2, and 3 are $n^{3/2}, n^{3/2}$ and n^3 respectively. So the hidden polynomial factor in our algorithm is n^9 without the number of iterations.

Remark 2. It is natural to extend the three-level sieve algorithm to multiple-level, such as four-level algorithm. However, the number of small balls will increase as the number of the levels increases. Therefore, we conjecture that the time complexity may be decreased with small number levels, but will increase if the number of levels is greater than some positive integer.

4   Experimental Results

4.1   Comparison with the Other Heuristic Sieve Algorithms

We implemented the NV algorithm, the WLTB algorithm and our three-level sieve algorithm on a PC with Windows 7 system, 3.00 GHz Intel 4 processor and 2 GByte RAM using Shoup's NTL library version 5.4.1 [28]. Instead of implementing the GaussSieve algorithm, we directly applied the GaussSieve Alpha V.01 published by Voulgaris [29] on a PC with Fedora 15 system, 3.00 GHz Intel 4 processor and 2 GByte RAM.

   We performed experiments to compare our three-level sieve algorithm with the other three algorithms. For every dimension n, we first used the method in [18] to pick some random n-dimensional lattice and computed the LLL-reduced basis, then we sampled the same number of lattice vectors, and performed the NV algorithm with $\gamma = 0.97$, the WLTB algorithm with $\gamma_1 = 1.0927, \gamma_2 = 0.97$ and our three-level sieve algorithm with $\gamma_1 = 1.1399, \gamma_2 = 1.0667, \gamma_3 = 0.97$ using these samples. We performed one experiments on lattices with dimension 10, 20 with more than 100000 samples, but about fifty experiments with fewer samples, and two experiments on dimension 25, 30, 40, 50. Instead of using our samples, we just performed the GaussSieve Alpha V.01 with the selected lattices as its inputs. The experimental results of the four algorithms are shown in Table 2, where $\boldsymbol{v}$ is the output vector of the corresponding algorithm.

   In our experiments, the GaussSieve algorithm is much faster than the others and succeeds to find the shortest vectors for all the lattices we picked. Besides of the major reason that the GaussSieve algorithm performs better in practice (it has been reported that the GaussSieve algorithm is more efficient than the NV algorithm), another possible reason is that our implementation is a little poor.

Table 2. Experimental results.

	10	20	25	30	40	50	60
Dimension							
Number of sample	150000	100000	8000	5000	5000	3000	2000
Time of sample (s)	301	810	87833	73375	147445	120607	167916
Time (s) NV algorithm	25005	64351	120	220	625	254	187
WLTB algorithm	23760	18034	35	42	93	46	47
Our algorithm	20942	13947	27	27	57	29	30
GaussSieve algorithm	0.003	0.013	0.068	0.098	0.421	3.181	42.696
$\frac{\|v\|}{\lambda_1}$ NV algorithm	1	1	23.8	38.3	170.1	323	347.7
WLTB algorithm	1	1	25.9	35.1	170.1	323	347.7
Our three-level algorithm	1	1	21.2	38.3	170.1	323	347.7
GaussSieve algorithm	1	1	1	1	1	1	1

Compared with the NV and WLTB algorithms, it seems that our algorithm may be slower for low dimensional lattices due to the larger hidden polynomial factor. However, on one hand, the number of sieved vectors in each iteration of our algorithm decreases faster because the number of small balls is larger, which implies that the number of iterations is smaller and the number of the vectors to be sieved in the next iteration is smaller as well. On the other hand, the time complexity is for the worst case. In practice, we need not to check all the big balls, medium balls and small balls to decide which small ball the sieved vector belongs to. Thus, with the same number of samples in our experiments, our algorithm runs faster than the NV and WLTB algorithms. Since the sample procedure is very fast when the dimension n is not greater than twenty, we can sample enough lattice vectors to ensure that the three algorithms can find a shortest nonzero lattice vector. In such case, the time of sieving overwhelms the time of sampling, so our algorithm usually costs the least total time.

4.2   On Heuristic Assumption 1

To test the validity of the Heuristic Assumption 1 that the distribution of the sieved vectors remains uniform, we picked four random lattices of dimension 10, 25, 40 and 50, sampled 150000, 8000, 5000, 3000 lattice vectors and then sieved them respectively. As in [20], we plotted the number of sieved vectors in each iteration (see Fig. 5). It can be seen that the head and the tail of the curve change slightly, but most of the curve, the middle part, decreases *regularly*. The lost vectors in each iteration are those used as centers or reduced to zero which means collisions occur. So the curve shows that the numbers of centers and collisions in most of the iterations are nearly the same, which partially suggests that the distribution of the sieved vectors is close to uniform throughout the iterations.

(a) n=10

(b) n=25

(c) n=40

(d) n=50
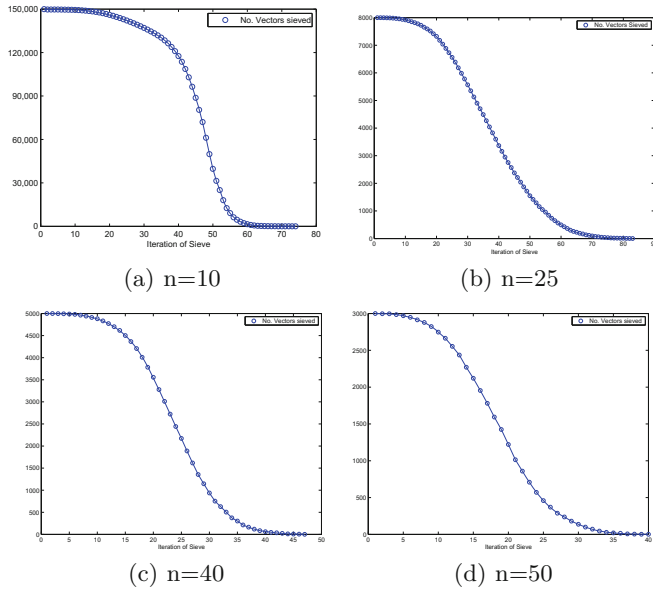
Fig. 5. Cardinality of the set of sieved vectors.

5   Conclusion

In this paper, we propose a three-level heuristic sieve algorithm to solve SVP and prove that the optimal running time is $2^{0.3778n+o(n)}$ polynomial-time operations and the space requirement is $2^{0.2833n+o(n)}$ polynomially many bits under Heuristic Assumption 1.

References

1. Adleman, L.M.: On breaking generalized knapsack public key cryptosystems. In: The 15th Annual ACM Symposium on Theory of Computing Proceedings, pp. 402–412. ACM, April 1983
2. Ajtai, M.: The shortest vector problem in l_2 is NP-hard for randomized reductions. In: Proceedings of the 30th STOC. ACM (1998)
3. Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proceedings of the 33rd STOC, pp. 601–610. ACM (2001)
4. Blömer, J., Naewe, S.: Sampling methods for shortest vectors, closest vectors and successive minima. Theor. Comput. Sci. **410**(18), 1648–1665 (2009)

5. Fincke, U., Pohst, M.: A procedure for determining algebraic integers of given norm. In: van Hulzen, J.A. (ed.) EUROCAL. LNCS, vol. 162, pp. 194–202. Springer, Heidelberg (1983)
6. Fincke, U., Pohst, M.: Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. Math. Comp. **44**(170), 463–471 (1985)
7. Gama, N., Howgrave-Graham, N., Koy, H., Nguyên, P.Q.: Rankin's constant and blockwise lattice reduction. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 112–130. Springer, Heidelberg (2006)
8. Gama, N., Nguyen, P.Q.: Finding short lattice vectors within Mordell's inequality. In: STOC '08-Proceedings of the 40th ACM Symposium on the Theory of Computing. ACM (2008)
9. Gama, N., Nguyen, P.Q., Regev, O.: Lattice enumeration using extreme pruning. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 257–278. Springer, Heidelberg (2010)
10. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
11. Kannan, R.: Improved algorithms for integer programming and related lattice problems. In: Proceedings of the 15th STOC, pp. 193–206. ACM (1983)
12. Klein, P.N.: Finding the closest lattice vector when it's unusually close. In: Proceedings of the SODA, pp. 937–941. ACM (2000)
13. Lenstra, A.K., Lenstra Jr, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**, 513–534 (1982)
14. Lagarias, J.C., Odlyzko, A.M.: Solving low-density subset sum problems. J. ACM **32**(1), 229–246 (1985)
15. Micciancio, D., Voulgaris, P.: A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In: Proceedings of the STOC, pp. 351–358. ACM (2010)
16. Micciancio, D., Voulgaris, P.; Faster exponential time algorithms for the shortest vector problem. In: The 21st Annual ACM-SIAM Symposium on Discrete Algorithms Proceedings, pp. 1468–1480. SIAM, January 2010
17. Milde, B., Schneider, M.: A parallel implementation of GaussSieve for the shortest vector problem in lattices. In: Malyshkin, V. (ed.) PaCT 2011. LNCS, vol. 6873, pp. 452–458. Springer, Heidelberg (2011)
18. Nguyên, P.Q., Stehlé, D.: LLL on the average. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 238–256. Springer, Heidelberg (2006)
19. Nguyên, P.Q., Stern, J.: The two faces of lattices in cryptology. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 146–180. Springer, Heidelberg (2001)
20. Nguyen, P.Q., Vidick, T.: Sieve algorithms for the shortest vector problem are practical. J. Math. Cryptology **2**(2), 181–207 (2008)
21. Pujol, X., Stehlé, D.: Solving the shortest lattice vector problem in time $2^{2.465n}$. Cryptology ePrint Archive, Report 2009/605 (2009)
22. Regev, O.: Lecture notes on lattices in computer science. http://www.cs.tau.ac.il/odedr/teaching/latticesfall2004/index.html (2004)
23. Schneider, M.: Analysis of Gauss-sieve for solving the shortest vector problem in lattices. In: Katoh, N., Kumar, A. (eds.) WALCOM 2011. LNCS, vol. 6552, pp. 89–97. Springer, Heidelberg (2011)
24. Schneider, M.: Sieving for shortest vectors in ideal lattices. In: Youssef, A., Nitaj, A., Hassanien, A.E. (eds.) AFRICACRYPT 2013. LNCS, vol. 7918, pp. 375–391. Springer, Heidelberg (2013)

25. Schnorr, C.P.: A hierarchy of polynomial lattice basis reduction algorithms. Theoret. Comput. Sci. **53**, 201–224 (1987)
26. Schnorr, C.P., Euchner, M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math. Program. **66**, 181–199 (1994)
27. Shamir, A.: A polynomial time algorithm for breading the basic Merkel-Hellman cryptosystem. In: The 23rd IEEE Symposium on Foundations of Computer Science Proceedings, pp. 145–152. IEEE (1982)
28. Shoup, V.: NTL: a library for doing number theory. http://www.shoup.net/ntl/
29. Voulgaris, P.: Gauss Sieve alpha V. 0.1 (2010). http://cseweb.ucsd.edu/pvoulgar/impl.html
30. Wang, X., Liu, M., Tian, C., Bi, J.: Improved Nguyen-Vidick Heuristic sieve algorithm for shortest vector problem. In: The 6th ACM Symposium on Information, Computer and Communications Security Proceedings, pp. 1–9. ACM (2011)