Improved Nguyen-Vidick Heuristic Sieve Algorithm for Shortest Vector Problem *

Xiaoyun Wang[1,2], Mingjie Liu[1], Chengliang Tian[2] and Jingguo Bi[2]

[1] Institute for Advanced Study, Tsinghua University, Beijing 100084, China
xiaoyunwang@mail.tsinghua.edu.cn,liu-mj07@mails.tsinghua.edu.cn
[2] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China
{chengliangtian,jguobi}@mail.sdu.edu.cn

Abstract. In this paper, we present an improvement of the Nguyen-Vidick heuristic sieve algorithm for shortest vector problem in general lattices, which time complexity is $\mathbf{2^{0.3836n}}$ polynomial computations, and space complexity is $\mathbf{2^{0.2557n}}$. In the new algorithm, we introduce a new sieve technique with two-level instead of the previous one-level sieve, and complete the complexity estimation by calculating the irregular spherical cap covering.

keywords: lattice, shortest vector, sieve, heuristic, sphere covering

1 Introduction

The n-dimensional lattice Λ is generated by the basis $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\} \subset \mathbb{R}^m$ which consists n linearly independent vectors.

$$\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{z} = \sum_{i=1}^{n} z_i \mathbf{b}_i : \mathbf{z} \in \mathbb{Z}^n\}.$$

The minimum distance $\lambda_1(\Lambda)$ of a lattice Λ is the length of its shortest nonzero vector:

$$\lambda_1(\Lambda) = \min_{0 \neq \mathbf{x} \in \Lambda} \|\mathbf{x}\|.$$

Here $\|\mathbf{x}\|$ is the Euclidean norm of the vector $\mathbf{x}$. The problem of finding a lattice point $\mathbf{v}$ with norm $\lambda_1(\Lambda)$ is called the Shortest Vector Problem (SVP). The γ-approximation to SVP denoted as SVP_γ is the problem of finding a lattice point $\mathbf{v}$ with the length $\|\mathbf{v}\| \leq \gamma\lambda_1(\Lambda)$.

SVP is a classical mathematical problem originated from geometry of numbers [15, 25], and it is also a NP-hard problem in computational complexity theory [4]. In the past thirty years, SVP has been widely used in public-key cryptanalysis and lattice-based cryptography. On one hand, the fast algorithm for searching SVP or SVP_γ [13, 33] is a fundamental tool in public-key cryptanalysis and lattice-based cryptanalysis [26]. The most successful polynomial algorithm to search a shorter vector with approximation $2^{(n-1)/2}$ is the LLL basis reduction algorithm [20] which has been successfully used in breaking most Knapsack encryptions [2, 21, 32] and resulted in various weak key attacks on RSA-like cryptosystems [7, 9]. On the other hand, many cryptographic functions corresponding to SVP variants are proposed to be acted as the trapdoor one-way functions so that various lattice-based cryptographic schemes

* Supported by the National "973" Program of China (Grant No.2007CB807902) and National Natural Science Foundation of China (Grant No.60931160442).

are easy to be constructed [3, 6, 12, 31]. There are two popular cryptographic functions which are derived from SIS (small integer solution) problem and LWE (learning with errors) problem respectively, and it is noted that SIS and LWE can be reduced to an SVP variant–SIVP_γ(Shortest Independent Vectors Problem).

Today, fast searching shortest vector has become the most important focus point both on the security assessment and cryptanalysis in lattice-based cryptography. Because SVP is NP-hard, the exact algorithm to search for SVP is not expected to be polynomial time. So far, there are essentially two different types of algorithms for exact SVP: deterministic algorithms and randomized sieve algorithms.

The first deterministic algorithm for SVP is originated from the work of Pohst [36] and Kannan [19], which is named as deterministic enumeration algorithm. Its main idea is to enumerate all lattice vectors shorter than a fixed bound $A \geq \lambda_1(\Lambda)$, with the help of the Gram-Schmidt orthogonalization of the given lattice basis. Given an LLL-reduced basis as input, the algorithm of Fincke and Pohst [11] runs in time $2^{O(n^2)}$, while the worst-case complexity of Kannan's algorithm is $n^{\frac{n}{2e}+o(n)}$ [16]. See the survey paper [1] for more details. Among the enumeration algorithms, the Schnorr-Euchner enumeration strategy [34] is the most important one used in practice, whose running time is $2^{O(n^2)}$ polynomial-time operations where the basis is either LLL-reduced or BKZ-reduced. Recently, Gama, Nguyen and Regev propose a new technique called extreme pruning in enumeration algorithm to achieve exponential speedups both in theory and in practice [14]. All enumeration algorithms we mentioned above only require a polynomial data complexity.

A completely different deterministic algorithm for SVP is based on Voronoi cell computation which originally aimed at solving the Closest Vector Problem (CVP) [35]. Recently, Micciancio and Voulgaris [24] proposed an improved algorithm which is applicable to most lattice problems, including SVP, CVP and SIVP. The running time is $\widetilde{O}(2^{2n+o(n)})$ polynomial-time operations, where $f = \widetilde{O}(g)$, means $f(n) \leq \log^c g(n) \cdot g(n)$ for some constant c and all sufficiently large n. This is so far the best known result in lattice computational complexity in the deterministic search setting.

Another type algorithm for exact SVP is the randomized sieve algorithm, which was first proposed in 2001 by Ajtai, Kumar and Sivakumar [5] (AKS sieve algorithm). The sieve method reduces upper bound of the time to $2^{O(n)}$ at the cost of $2^{O(n)}$ space. In [30] Regev got the first constant estimation with time $2^{16n+o(n)}$ and space $2^{8n+o(n)}$, and further decreased to time $2^{5.90n+o(n)}$ and space $2^{2.95n+o(n)}$ by Nguyen and Vidick [28]. Micciancio and Voulgaris utilized the bound estimation of sphere packing [17], and improved both the time and space complexity to $2^{3.40n+o(n)}$ and $2^{1.97n+o(n)}$ respectively [23], and further reduced to $2^{3.199n+o(n)}$ and space $2^{1.325n+o(n)}$ by combining with ListSieve technique. By implementing the birthday attack on the sieved shorter vectors in a small ball with the radius $3.01\lambda_1(\Lambda)$, Pujol and Stehlé give a sieve algorithm to search SVP with the time complexity $2^{2.465n+o(n)}$ [29].

Besides the above algorithms, there is a more practical searching algorithm which is heuristic under a natural random assumption. Nguyen and Vidick [28] presented the first heuristic variant of AKS sieve [5] with time $2^{0.415n}$ and space $2^{0.2075n}$, which is so far the fastest randomized sieve algorithm. It is remarked that, Micciancio and Voulgaris [23] also described a heuristic ListSieve called Gauss Sieve, which performed fairly well in practice but the upper bound time complexity of this sieve is still unknown.

In this paper, we present an improved heuristic randomized algorithm which solves SVP with time $2^{0.3836n}$ and space $2^{0.2557n}$. The main idea of the algorithm is to collect shorter vectors by two-level sieve. The estimation of the complexity is based on the computation of the irregular spherical cap covering which comes from the intersection of a spherical surface and two balls.

This paper is organized as follows: Section 2 gives some notations and preliminaries. The new algorithm is introduced in Section 3. We present a proof of the algorithm complexity in Section 4. Conclusions are given in Section 5.

2 Notations and Preliminaries

- $\omega(f(n))$ represents a function growing faster than $cf(n)$ for any $c > 0$.
- $\Theta(f(n))$ is a function that has the same order as $f(n)$, when $n \to \infty$.
- Let $S^n = \{\mathbf{x} \in \mathbb{R}^n | \ \ \|\mathbf{x}\| = 1\}$ be the unit sphere in $\mathbb{R}^n$.
- $B_n(\mathbf{x}, r)$ denotes the n-dimensional ball centered at $\mathbf{x}$ with radius r, and is simplified as $B_n(r)$ when center is the origin.
- κ_n is the volume of the unit Euclidean n-dimensional ball.
- $C_n(\gamma R) = \{\mathbf{x} \in \mathbb{R}^n \mid \gamma R \le \|\mathbf{x}\| \le R\}$ is a spherical shell in the ball $B_n(R)$.
- $B(\varphi, \mathbf{x}) = \{\mathbf{y} \mid \langle \mathbf{x}, \mathbf{y} \rangle > \cos\varphi, \mathbf{y} \in S^n\}$ is the spherical cap with angle φ in S^n,$\varphi \in (0, \frac{\pi}{2})$.
- $\tilde{B}(\varphi, \mathbf{x}, \gamma) = \{\mathbf{y} \mid \langle \mathbf{x}, \mathbf{y} \rangle > \cos\varphi, \mathbf{y} \in C_n(\gamma)\}$ is the spherical cap with height and angle φ in $C_n(\gamma)$,$\varphi \in (0, \frac{\pi}{2})$.
- $|A|$ represents its volume if A is a geometric body or its cardinality if A is a finite set.

We note that $|S^n| = n\kappa_n$. it is well-known that

$$\kappa_n = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} = \begin{cases} \frac{\pi^k}{k!}, & n = 2k \\ \frac{2^{2k+1}k!\pi^k}{(2k+1)!}, & n = 2k+1 \end{cases}$$

where $\Gamma(z) = \int_0^\infty t^{z-1}\mathrm{e}^{-t}\mathrm{d}t$ is the gamma function.

Lemma 1. *[8] Let $\varphi \in (0, \frac{\pi}{2})$, and $\mathbf{x} \in S^n$, if $\varphi \le \arccos\frac{1}{\sqrt{n}}$, then*

$$\frac{\kappa_{n-1}}{3\cos\varphi}(\sin\varphi)^{n-1} < |B(\varphi, \mathbf{x})| < \frac{\kappa_{n-1}}{\cos\varphi}(\sin\varphi)^{n-1}.$$

Define $\Omega_n(\varphi) = \frac{|B(\varphi,\mathbf{x})|}{|S^n|} = \frac{|\tilde{B}(\varphi,\mathbf{x},\gamma)|}{|C_n(\gamma)|}$. From the facts that $\sqrt{\frac{n}{2\pi}} < \frac{\kappa_{n-1}}{\kappa_n} < \sqrt{\frac{n+1}{2\pi}}$, the following corollary holds.

Corollary 1. *[8] Let $\varphi \in (0, \frac{\pi}{2})$, if $\varphi \le \arccos\frac{1}{\sqrt{n}}$, then*

$$\frac{1}{3\sqrt{2\pi n}}\frac{1}{\cos\varphi}(\sin\varphi)^{n-1} < \Omega_n(\varphi) < \frac{1}{\sqrt{2\pi(n-1)}}\frac{1}{\cos\varphi}(\sin\varphi)^{n-1}.$$

For any real $s > 0$, the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with parameter s is given as follows.

$$\forall \mathbf{x} \in \mathbb{R}^n, \rho_{\mathbf{s},\mathbf{c}}(\mathbf{x}) = e^{-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2}.$$

The subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$ (respectively) when omitted.

For any $\mathbf{c} \in \mathbb{R}^n$, real $s > 0$, and n-dimensional lattice Λ, define the discrete Gaussian distribution over Λ as:

$$\forall \mathbf{x} \in \Lambda, D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\mathbf{s},\mathbf{c}}(\mathbf{x})}{\rho_{\mathbf{s},\mathbf{c}}(\Lambda)},$$

where $\rho_{\mathbf{s},\mathbf{c}}(A) = \sum_{x \in A} \rho_{\mathbf{s},\mathbf{c}}(\mathbf{x})$ for any countable set A.

A function $\varepsilon(n)$ is negligible if $\varepsilon(n) < 1/n^c$ for any $c > 0$ and all sufficiently large n. Statistical distance between two distributions $\mathbf{X}$ and $\mathbf{Y}$ over a countable domain D is defined as $\frac{1}{2}\sum_{d \in D} |X(d) - Y(d)|$. We say two distributions (indexed by n) are statistically close if their statistical distance is negligible in n.

Lemma 2. *[12] There is a probabilistic polynomial-time algorithm that, given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, a parameter $s \geq \|\tilde{\mathbf{B}}\|\omega(\sqrt{\log n})$, and a center $\mathbf{c} \in \mathbb{R}^n$, outputs a sample from a distribution that is statistically close to $D_{\Lambda,s,\mathbf{c}}$.*

Similar to NV heuristic algorithm, our algorithm requires that the lattice points distribute in $C_n(\gamma_2 R)$ uniformly at any stage of the algorithm. We need to select the sample by applying Klein randomized variant of nearest plane algorithm [18] so that the initial chosen sample is indistinguishable from Gauss distribution. The distribution proof of Klein algorithm can be found in [12].

3 Algorithm

Section 3.1 gives a brief description of Nguyen-Vidick heuristic sieve algorithm (NV algorithm). Then in section 3.2, under the same natural heuristic assumption as NV algorithm, we present a new algorithm with two-level sieve, which can output the shortest vector in time $2^{0.3836n}$.

3.1 Nguyen and Vidick's heuristic sieve algorithm

We start with the randomized algorithm proposed by Ajtai, Kummar and Sivakumar(AKS sieve)[5]. The main idea of the algorithm is as follows: sample $2^{O(n)}$ lattice vectors in a ball $B_n(R)$ for $R = 2^{O(n)}\lambda_1$, then implement a partition and a sieve method to search enough shorter vectors within $B_n(\gamma R)$, for $\gamma < 1$ without losing many vectors. R gets updated every time which is close to λ_1 by a polynomial iterations, while the short vectors left are enough to get the shortest vector. Until now, the perturbation technique in sampling procedure is necessary to prove the successful probability of finding the shortest vector in all randomized algorithms. But the effect of perturbation in practice is unclear. In [28], Nguyen and Vidick presented a fast heuristic algorithm (NV algorithm) which collects the short vectors by directly sieving the chosen lattice vectors instead of sieving the lattice vectors derived from perturbed points. The NV algorithm has $2^{0.415n}$ time and $2^{0.2075n}$ space complexity.

In every sieve iteration of NV algorithm, the input is a set S of lattice vectors with maximal norm R which can be regarded as having the random distribution in $C_n(\gamma R)$. The main purpose of NV algorithm is to randomly select a subset C of S as the center points which are located in $C_n(\gamma R)$. The set C has enough points such that for any vector $\mathbf{a}$ in S, there is at least a point $\mathbf{c} \in C$ with $\mathbf{a} - \mathbf{c}$ shorter than γR. $\mathbf{a} - \mathbf{c}$ is an output shorter vector in the iteration. In every iteration, the sieve captures a new set of vectors within the ball $B_n(\gamma R)$ without losing many vectors by selecting available γ and the size of C, i.e. the upper norm bound of the set shrinks by γ. Then after a polynomial iterations, the shortest vector will be included in the sieved short vectors, and can be found by searching. The core of NV algorithm is the sieve in Algorithm 1.

The main part of the data complexity is determined by the upper size of the point centers C which should guarantee that after polynomial number of iterations the set S is not empty. The estimation of $|C|$ is based on a natural assumption, and the experiments shows the assumption is rational.

Algorithm 1 The NV sieve

Input: An subset $S \subseteq B_n(R)$ of vectors in a lattice L, sieve factors $\sqrt{\frac{2}{3}} < \gamma < 1$.
Output: A subset $S' \subseteq B_n(\gamma R) \cap L$.

$R \leftarrow \max_{\mathbf{v}\in S} \|\mathbf{v}\|$.
$C \leftarrow \emptyset, S' \leftarrow \emptyset$
for $\mathbf{v} \in S$ do
  if $\|\mathbf{v}\| \leq \gamma R$ then
    $S' \leftarrow S' \cup \{\mathbf{v}\}$.
  else
    if $\exists \mathbf{c} \in C, \|\mathbf{v} - \mathbf{c}\| \leq \gamma R$ then
      $S' \leftarrow S' \cup \{\mathbf{v} - \mathbf{c}\}$.
    else
      $C \longleftarrow C \cup \{\mathbf{v}\}$
    end if
  end if
end for

Heuristic Assumption: At any stage in the Algorithm, the vectors in $S \cap C_n(\gamma R)$ are uniformly distributed in $C_n(\gamma R) = \{\mathbf{x} \in \mathbb{R}^n : \gamma R \leq \|\mathbf{x}\| \leq R\}$.

3.2 New Sieve Algorithm

In NV algorithm, the time complexity is the square of space complexity. In order to achieve some balance between time and space, we try a new sieve method which in fact uses a two-level sieve. (Algorithm 3). Our algorithm also includes polynomial iterations, and each iteration consists of two level sieves. At the first level, we partition the lattice points in the spherical shell $C_n(\gamma_2 R)$ into different big balls rather than small ones in Algorithm 1 (See Fig.1). In the second level, we cover every spherical cap (intersection of the big ball and $C_n(\gamma_2 R)$) using small balls which are centered at a lattice point in the same spherical cap. By comparing all the lattice points in the spherical cap with centers of every small ball, we can get some shorter vectors. Merging all the short vectors calculated in every spherical cap, we obtain the required short lattice vectors for the next iteration. It is clear that, the first level sieve needs less number of big balls which saves the comparing time. At the second level, each shorter vector is obtained by pair-wise difference among the lattice points in a spherical cap. This means that more data is required in order to get enough shorter vectors. In particular, the Heuristic Assumption in NV sieve guarantees the uniform distribution in $S \cap C_n(\gamma R)$ which also supports our algorithm.

The frame of our algorithm is given in Algorithm 2. Instead of the shrink factor γ in NV sieve, we use two pivotal input parameters γ_1, γ_2. These parameters will be used to determine N and estimate the complexity and efficiency. In steps 1-5, we generate N lattice vectors within a proper length similar to NV sieve. Steps 6-11 are the key parts of our algorithm which reduce the norm of lattice vectors in S by a factor γ_2 without significantly decreasing

Fig. 1.

the size of S. This new sieve is different from any known sieve in which we put longer lattice vectors in different big balls firstly, then perform sieve again in separate big balls. The details will be given in Algorithm 3. This main loop repeats until the set S is empty, and the shortest vector in S_0 is returned. The size of S decreases in two ways: firstly in Algorithm 3 the vector used as center vector is removed from S; secondly, in step 10 the appearance of zero vector vanishes some vectors. We will provide a detailed analysis to estimate the number of vectors that are got rid of from the process.

Algorithm 2 Finding short lattice vectors based on sieving

Input: An LLL-reduced basis $B = [b_1, \cdots, b_n]$ of a lattice L, sieve factors γ_1, γ_2 such that $\sqrt{\frac{2}{3}} < \gamma_2 < 1 < \gamma_1 < \sqrt{2}\gamma_2$, and a number N.
Output: A short non-zero vector of L.
1: $S \leftarrow \emptyset$.
2: for $j = 1$ to N do
3: $\quad S \leftarrow S \bigcup$ sampling(B) using Klein's algorithm
4: end for
5: Remove zero vector from S.
6: $S_0 \leftarrow S$
7: repeat
8: $\quad S_0 \leftarrow S$
9: $\quad S \leftarrow$ latticesieve(S, γ_1, γ_2) using algorithm 3.
10: $\quad$ Remove zero vector from S.
11: Until $S = \emptyset$
12: Compute $\mathbf{v_0} \in S_0$ such that $\|\mathbf{v_0}\| = \min\{\|\mathbf{v}\|, \mathbf{v} \in S_0\}$
13: return $\mathbf{v_0}$

Under the heuristic assumption, the collisions in step 10 of algorithm 2 are negligible until $\sqrt{|B_n(R) \cap L|} \le |S \cap C_n(\gamma_2 R)|$ which means the upper bound of the norm get very close to the shortest length in lattice.

Two levels of center points are used in our sieve. Let C_1 be the set of centers of big balls with radius $\gamma_1 R$ in the first level, where $\gamma_1 > 1$. Since we can not get the short vectors in a

Algorithm 3 The lattice sieve

Input: An subset $S \subseteq B_n(R)$ of vectors in a lattice L, sieve factors $\sqrt{\frac{2}{3}} < \gamma_2 < 1 < \gamma_1 < \sqrt{2}\gamma_2$.
Output: A subset $S' \subseteq B_n(\gamma_2 R) \cap L$.

$R \leftarrow \max_{\mathbf{v} \in S} \|\mathbf{v}\|$.
$C_1 \leftarrow \emptyset, C_2 \leftarrow \{\emptyset\}, S' \leftarrow \emptyset$
for $\mathbf{v} \in S$ do
  if $\|\mathbf{v}\| \leq \gamma_2 R$ then
    $S' \leftarrow S' \cup \{\mathbf{v}\}$.
  else
    if $\exists \mathbf{c} \in C_1, \|\mathbf{v} - \mathbf{c}\| \leq \gamma_1 R$ then
      if $\exists \mathbf{c}' \in C_2^{\mathbf{c}}, \|\mathbf{c}' - \mathbf{v}\| \leq \gamma_2 R$ \ $C_2^{\mathbf{c}}$ is initialized as empty set \
        $S' \leftarrow S' \cup \{\mathbf{v} - \mathbf{c}'\}$.
      else
        $C_2^{\mathbf{c}} \longleftarrow C_2^{\mathbf{c}} \cup \{\mathbf{v}\}$
      end if
    else
      $C_1 \longleftarrow C_1 \cup \{\mathbf{v}\}, C_2 \longleftarrow C_2 \cup \{C_2^{\mathbf{v}} = \{v\}\}$
    end if
  end if
end for

big ball by subtracting its center directly, a second level covering is needed. $C_2^{\mathbf{c}}$ consists of the centers of small balls in the second level that cover a big ball with center $\mathbf{c}$. It is clear that, $C_2^{\mathbf{c}}$ is selected in the regular spherical cap $C_n(\gamma_2 R) \cap B_n(\mathbf{c}, \gamma_1 R)$. All $C_2^{\mathbf{c}}$ are merging into one set C_2, i.e., $C_2 = \bigcup_{c \in C_1} C_2^{\mathbf{c}}$. Denote the expected number of lattice vectors in C_1 as N_{C_1} and the expected size of every $C_2^{\mathbf{c}}$ as $N_{C_2^{\mathbf{c}}}$. To estimate N_{C_1}, we have to calculate the fraction of the spherical cap $C_n(\gamma_2 R) \cap B_n(\mathbf{c}, \gamma_1 R)$ in $C_n(\gamma_2 R)$. The right part of Fig. 1 illustrates the covering of first level. $N_{C_2^{\mathbf{c}}}$ is the number of small balls centered in $C_2^{\mathbf{c}}$ with radius $\gamma_2 R$, which cover the spherical cap $C_n(\gamma_2 R) \cap B_n(\mathbf{c}, \gamma_1 R)$ with probability close to 1. The region of $C_n(\gamma_2 R) \cap B_n(\mathbf{c}, \gamma_1 R) \cap B_n(\mathbf{c}', \gamma_2 R)$, $\mathbf{c}' \in C_2^{\mathbf{c}}$ is a regular or irregular spherical cap whose volume determines the number of $N_{C_2^{\mathbf{c}}}$. Fig. 2 shows the second covering. O_b denotes a center $\mathbf{c}$ of the first-level big ball, and O_s is a center $\mathbf{c}'$ of second-level small ball. We give the estimations for N_{C_1} and $N_{C_2^{\mathbf{c}}}$ in the following theorems.

The purpose of every iteration is to compress a large number of lattice points in $B_n(R)$ to $B_n(\gamma_2 R)$ without losing many points. So we just consider the covering of the spherical shell $C_n(\gamma_2 R)$. Applying LLL reduced input basis and the Klein's sample algorithm with proper parameter, we can choose the initial R smaller than $2^{O(n)}\lambda_1$. After every two-level sieve of the algorithm, the upper bound of the norm shrinks by γ_2. If the number of sampled vectors is not less than $poly(n) N_{C_1} N_{C_2^{\mathbf{c}}}$, then it is expected that the shortest vector remains after a polynomial many time iterations,.

The upper bound of C_1 and $C_2^{\mathbf{c}}$ are given in Theorem 1 and Theorem 2 respectively.

Theorem 1. *Let n be a non-negative integer, and $\frac{1}{2} < \gamma_2 < 1 < \gamma_1 < \sqrt{2}\gamma_2$,*

$$N_{C_1} = c_{\mathcal{H}_1}^n \lceil 3\sqrt{2\pi} n^{\frac{3}{2}} \rceil,$$

where $c_{\mathcal{H}_1} = \dfrac{1}{\gamma_1 \sqrt{1 - \frac{\gamma_1^2}{4}}}$. S is a subset of $C_n(\gamma_2 R)$ of cardinality N whose points are picked independently at random with uniform distribution. If $N_{C_1} < N < 2^n$, then for any subset

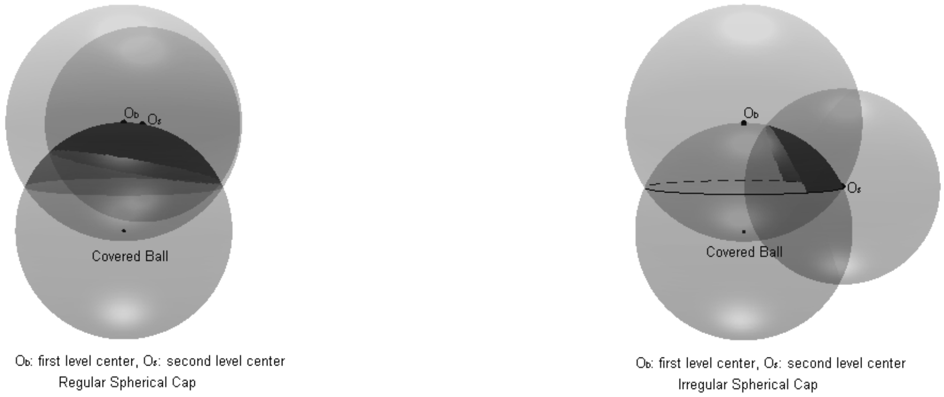

Fig. 2. Second Level Covering in the New Algorithm

$C \subseteq S$ of size at least N_{C_1} whose points are picked independently at random with uniform distribution, with overwhelming probability, for all $\mathbf{v} \in S$, there exists a $\mathbf{c} \in C$ such that $\|\mathbf{v} - \mathbf{c}\| \leq \gamma_1 R$.

Theorem 2. *Let n be a non-negative integer, $\gamma_2 < 1 < \gamma_1 < \sqrt{2}\gamma_2$, γ_2 is very close to 1,*

$$N_{C_2^c} = c\left(\frac{c_{\mathcal{H}_2}}{d_{\min}}\right)^n \lceil n^{\frac{3}{2}} \rceil,$$

where $c_{\mathcal{H}_2} = \frac{\gamma_1}{\gamma_2}\sqrt{1 - \frac{\gamma_1^2}{4\gamma_2^2}}, d_{\min} = \gamma_2\sqrt{1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}}$, c is a positive constant unrelated to n. S is a subset of $\{\mathbf{x} \in C_n(\gamma_2 R) \mid \|\mathbf{x} - \mathbf{c}_1\| \leq \gamma_1 R\}$ of cardinality N whose points are picked independently at random with uniform distribution. If $N_{C_2^c} < N < 2^n$, then for any subset $C \subseteq S$ of size at least $N_{C_2^c}$ whose points are picked independently at random with uniform distribution, with overwhelming probability, for all $\mathbf{v} \in S$, there exists a $\mathbf{c} \in C$ such that $\|\mathbf{v} - \mathbf{c}\| \leq \gamma_2 R$.

4 Proof of the Complexity

This section contains the proofs of Theorem 1,2 and the complexity estimation of our algorithm.

Since R has no effect on the conclusion of Theorem 1 and Theorem 2, we prove following lemma for unit ball. Let $\Omega_n(\gamma_1)$ be the fraction of $C_n(\gamma_2)$ that is covered by a ball of radius γ_1 centered in a point of $C_n(\gamma_2)$.

Lemma 3. $\sqrt{\frac{2}{3}} < \gamma_2 < 1 < \gamma_1 < \sqrt{2}\gamma_2$,

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

where $\theta_1 = \arccos(1 - \frac{\gamma_1^2}{2\gamma_2^2})$, $\theta_2 = \arccos(1 - \frac{\gamma_1^2}{2})$.

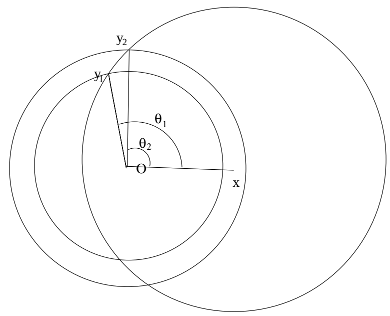

Fig. 3.

Proof. Let $\mathbf{x} \in C_n(\gamma_2)$, $\|\mathbf{x}\| = \alpha_1$ where $\gamma_2 \le \alpha_1 \le 1$. $\mathbf{y}_1$ and $\mathbf{y}_2$ are two points in the spherical cap $C_n(\gamma_2)$ which are at distance γ_1 from $\mathbf{x}$, and $\|\mathbf{y}_1\| = \gamma_2$, and $\|\mathbf{y}_2\| = 1$. Denote the angle of vertices $\mathbf{x}$, O and $\mathbf{y}_1$ as θ_1 and θ_2 is the angle of vertices $\mathbf{x}$, O and $\mathbf{y}_2$. We have $\cos\theta_1 = \frac{\alpha_1^2+\gamma_2^2-\gamma_1^2}{2\alpha_1\gamma_2}$, $\cos\theta_2 = \frac{\alpha_1^2+1-\gamma_1^2}{2\alpha_1}$. From $\gamma_1^2 > \alpha_1^2 - \gamma_2$ and $\gamma_2 < 1$, we know that $\cos\theta_1 < \cos\theta_2$. This implies $\theta_1 > \theta_2$. Then $\tilde{B}(\theta_2, \mathbf{x}, \gamma_2) \subset \Omega_n(\gamma_1) \subset \tilde{B}(\theta_1, \mathbf{x}, \gamma_2)$. By Corollary 1, we have

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

Furthermore, both $\cos\theta_1$ and $\cos\theta_2$ increases with α_1. So the lower bound is given by $\alpha_1 = 1$, where $\theta_2 = \arccos(1 - \frac{\gamma_1^2}{2})$. When $\alpha_1 = \gamma_2$, $\theta_1 = \arccos(1 - \frac{\gamma_1^2}{2\gamma_2^2})$, we get the upper bound for $\Omega_n(\gamma_1)$. □

Remark 1. It is noted that Lemma 3 is similar to lemma 4.2 in [28]. The difference is that we generalize the formula to that of reflecting the exact expressions of angles θ_1, θ_2 with parameters γ_1 and γ_2, which are important in the main complexity estimation of Theorem 1 and Theorem 2.

Now we are ready to prove Theorem 1.

Proof. By lemma 3, we have

$$\Omega_n(\gamma_1) > \frac{1}{3\sqrt{2\pi n}}\frac{1}{\cos\theta_2}(\sin\theta_2)^{n-1} > \frac{1}{3\sqrt{2\pi n}}(\sin\theta_2)^{n-1} > \frac{1}{3\sqrt{2\pi n}}c_{\mathcal{H}_1}^{-n}.$$

The expected proportion of $C_n(\gamma_2)$ that is not covered by N_{C_1} balls of radius γ_1 centered at randomly chosen points of $C_n(\gamma_2)$ is $(1-\Omega_n(\gamma_1))^{N_{C_1}}$. So,

$$N_{C_1}\log(1-\Omega_{\mathrm{n}}(\gamma_1)) \le N_{C_1}(-\Omega_n(\gamma_1)) < c_{\mathcal{H}_1}^{n}\lceil 3\sqrt{2\pi}n^{\frac{3}{2}}\rceil \cdot \frac{-1}{3\sqrt{2\pi n}}c_{\mathcal{H}_1}^{-n} \le -n < -\log N,$$

which implies

$$(1-\Omega_n(\gamma))^{N_C} < e^{-n} < \frac{1}{N}.$$

Therefore, the expected number of uncovered points is smaller than 1. In other words, any point in $C(\gamma_2)$ is covered by a ball of radius γ_1 with successful probability $1 - e^{-n}$. □

Without loss of generality, we denote the center of one big ball centered at $C_n(\gamma_2)$ as $(\alpha_1, 0, \ldots, 0)$, where $\gamma_2 \le \alpha_1 \le 1$. The region of the regular spherical cap $B_n(c, \gamma_1) \cap C_n(\gamma_2)$ is denoted as M, i.e.,

$$M = \{(x_1, x_2, x_3, \ldots, x_n) \in C_n(\gamma_2) \mid (x_1 - \alpha_1)^2 + x_2^2 + \ldots + x_n^2 < \gamma_1^2\},$$

where $\gamma_2 \le \alpha_1 \le 1$. To discuss the covering of the M by the small balls $B'_n(\mathbf{c}', \gamma_2)$, $c' \in C_2^c$, we need to calculate the minimum fraction $\Omega_n(\gamma_1, \gamma_2)$ of the spherical cap $B'_n(\mathbf{c}', \gamma_2) \cap B_n(\mathbf{c}, \gamma_1) \cap C_n(\gamma_2)$ as c' ranging over C_2^c.

We denote $B'_n(\mathbf{c}', \gamma_2) \cap B_n(\mathbf{c}, \gamma_1) \cap C_n(\gamma_2)$ as H. Before estimate the proportion of spherical cap H in M, we need to clarify its location. From Fig. 2, we know that, when the small ball completely fall into the big ball, H is a regular spherical cap, otherwise it is an irregular spherical cap. Especially, if $\mathbf{c}'$ slips along the sphere of the big ball (See right part of Fig. 2), the fraction of H in M is minimal. We noted that, only a little part of H is regular, and most centers $\mathbf{c}'$ are close to surface of big balls. So, we only compute the volume of minimum H, i.e., $\mathbf{c}'$ is located at the sphere of a big ball $B_n(\mathbf{c}, \gamma_1)$.

Lemma 4. *Let $\gamma_2 < 1 < \gamma_1 < \sqrt{2}\gamma_2$, γ_2 is very close to 1, we have $\Omega_n(\gamma_1, \gamma_2) \ge c\frac{d_{\min}^{n-2}}{2\pi n}$, where $d_{\min} = \gamma_2\sqrt{1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}}$, $c_{\mathcal{H}_1} = \frac{1}{\gamma_1\sqrt{1-\frac{\gamma_1^2}{4}}}$, and c is a positive constant.*

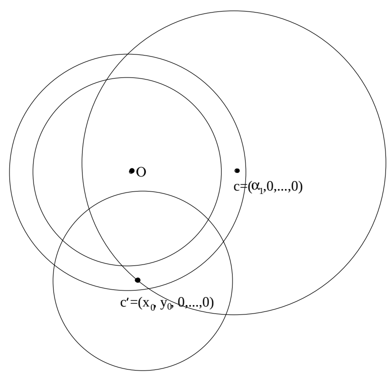

Fig. 4.

Proof. Note that γ_2 is selected close to 1 in our algorithm, to estimate the covering of the regular spherical cap M by irregular spherical cap, we just consider the proportion on the sphere covering rather than the shell covering.

Without loss of generality, we assume the center of $B_n(c, \gamma_1)$ as $(\alpha_1, 0, \ldots, 0)$, and the center $B'_n(c', \gamma_2)$ as $(x_0, y_0, 0, \ldots, 0)$ where $x_0 > 0$, $y_0 > 0$. According to the above description, the irregular spherical cap $B'_n(\mathbf{c}', \gamma_2) \cap B_n(\mathbf{c}, \gamma_1) \cap C_n(\gamma_2)$ with the minimum volume is expressed as:

$$\begin{cases} x_1^2 + x_2^2 + \ldots + x_n^2 = 1 \\ (x_1 - \alpha_1)^2 + x_2^2 + \ldots + x_n^2 < \gamma_1^2 \\ (x_1 - x_0)^2 + (x_2 - y_0)^2 + \ldots + x_n^2 < \gamma_2^2 \end{cases},$$

where $\gamma_2 \leq \alpha_1 \leq 1$, $(x_0-\alpha_1)^2+y_0^2=\gamma_1^2$, and $\gamma_2 \leq x_0^2+y_0^2 \leq 1$. In order to calculate this surface integral we project the target region to the hyperplane orthogonal to x_1, then this integral is changed to multiple integral. To simplify the expression, denote $A=x_0^2+y_0^2$ and $B=(A+1-\gamma_2^2)/2$. Let

$$D_1=\left\{(x_2,x_3,\ldots,x_n)\in\mathbb{R}^{n-1} \mid x_2^2+x_3^2+\ldots+x_n^2<1-\left(\frac{\alpha_1^2-\gamma_1^2+1}{2\alpha_1}\right)^2\right\}.$$

$$D_2^1=\left\{(x_2,x_3,\ldots,x_n) \mid \frac{A}{x_0^2}(x_2-\frac{By_0}{A})^2+x_3^2+\ldots+x_n^2<1-\frac{B^2}{x_0^2}(1-\frac{y_0^2}{A}), x_2<\frac{B}{y_0}\right\},$$

$$D_2^2=\left\{(x_2,x_3,\ldots,x_n) \mid x_2^2+x_3^2+\ldots+x_n^2<1, x_2\geq\frac{B}{y_0}\right\},\ D_2=D_2^1\cup D_2^2.$$

Let $R_1=\sqrt{1-\left(\frac{\alpha_1^2-\gamma_1^2+1}{2\alpha_1}\right)^2}$, $R_2=\sqrt{1-\frac{B^2}{x_0^2}\left(1-\frac{y_0^2}{A}\right)}$. The integral region is denoted as D which is the intersection of D_1 and D_2. By the equation $x_1^2+x_2^2+\ldots+x_n^2=1$, we have

$$x_1=\pm\sqrt{1-(x_2^2+\ldots+x_n^2)},\quad \frac{\partial x_1}{\partial x_i}=\frac{\mp x_i}{\sqrt{1-(x_2^2+\ldots+x_n^2)}}.$$

Now we can calculate the volume of the target region by computing

$$\begin{aligned}Q&=\int\int\cdots\int_D\sqrt{1+\left(\frac{\partial x_1}{\partial x_2}\right)^2+\ldots+\left(\frac{\partial x_1}{\partial x_n}\right)^2}\mathrm{d}x_2\mathrm{d}x_3\ldots\mathrm{d}x_n\\&=\int\int\cdots\int_D\frac{1}{\sqrt{1-(x_2^2+x_3^2+\ldots+x_n^2)}}\mathrm{d}x_2\mathrm{d}x_3\ldots\mathrm{d}x_n.\end{aligned}$$

We analysis the region D and first compute x_2 to simplify the above multiple integral. The upper bound and lower bound of x_2 is $\sqrt{R_1^2-(x_3^2+\ldots+x_n^2)}$ and $\frac{By_0}{A}-\frac{x_0}{\sqrt{A}}\sqrt{R_2^2-(x_3^2+\ldots+x_n^2)}$ respectively, while the region of $(x_3,\ldots,x_n)$ is a ball of dimension $n-2$ with radius

$$d=\sqrt{R_1^2-\left(\frac{1}{y_0}\left(B-x_0\left(\frac{\alpha_1^2-\gamma_1^2+1}{2\alpha_1}\right)\right)\right)^2}.$$

Therefore Q is expressed as,

$$\begin{aligned}Q&=\int_{\sum_{i=3}^n x_i^2<d^2}\cdots\int\left(\int_{\frac{By_0}{A}-\frac{x_0}{\sqrt{A}}\sqrt{R_2^2-\sum_{i=3}^n x_i^2}}^{\sqrt{R_1^2-\sum_{i=3}^n x_i^2}}\frac{1}{\sqrt{1-\sum_{i=3}^n x_i^2-x_2^2}}\mathrm{d}x_2\right)\mathrm{d}x_3\ldots\mathrm{d}x_n\\&=\int_{\sum_{i=3}^n x_i^2<d^2}\cdots\int\left(\arcsin\frac{\sqrt{R_1^2-\sum_{i=3}^n x_i^2}}{\sqrt{1-\sum_{i=3}^n x_i^2}}-\arcsin\frac{\frac{By_0}{A}-\frac{x_0}{\sqrt{A}}\sqrt{R_2^2-\sum_{i=3}^n x_i^2}}{\sqrt{1-\sum_{i=3}^n x_i^2}}\right)\mathrm{d}x_3\ldots\mathrm{d}x_n.\end{aligned}$$

Let

$$\begin{cases}x_3=t\cos\varphi_1\\x_4=t\sin\varphi_1\cos\varphi_2\\\qquad\vdots\\x_{n-1}=t\sin\varphi_1\ldots\sin\varphi_{n-4}\cos\varphi_{n-3}\\x_n=t\sin\varphi_1\ldots\sin\varphi_{n-4}\sin\varphi_{n-3}\end{cases},$$

then $0 \le t \le d, 0 \le \varphi_k \le \pi, k = 1, \dots, n-4, 0 \le \varphi_{n-3} \le 2\pi$. Furthermore, we get,

$$\frac{\partial(x_3, x_4, \dots, x_n)}{\partial(t, \varphi_1, \dots \varphi_{n-3})} = t^{n-3} \sin\varphi_{n-4} \dots (\sin\varphi_2)^{n-5} (\sin\varphi_1)^{n-4}.$$

So,

$$\begin{aligned} Q = &\int_0^d \int_0^{2\pi} \int_0^{\pi} \cdots \int_0^{\pi} t^{n-3} \sin\varphi_{n-4} \dots (\sin\varphi_2)^{n-5} (\sin\varphi_1)^{n-4} \left(\arcsin \frac{\sqrt{R_1^2 - t^2}}{\sqrt{1-t^2}} \right. \\ &\left. - \arcsin \frac{\frac{By_0}{A} - \frac{x_0}{\sqrt{A}} \sqrt{R_2^2 - t^2}}{\sqrt{1-t^2}} \right) \mathrm{d}\varphi_1 \dots \mathrm{d}\varphi_{n-3} \mathrm{d}t \\ =&2\pi \int_0^d t^{n-3} \left(\arcsin \frac{\sqrt{R_1^2 - t^2}}{\sqrt{1-t^2}} - \arcsin \frac{\frac{By_0}{A} - \frac{x_0}{\sqrt{A}} \sqrt{R_2^2 - t^2}}{\sqrt{1-t^2}} \right) \mathrm{d}t \prod_{k=1}^{k=n-4} \int_0^{\pi} \sin^k \varphi \mathrm{d}\varphi. \end{aligned}$$

From $\int_0^{\pi} \sin^k \varphi \mathrm{d}\varphi = 2\int_0^{\pi/2} \sin^k \varphi \mathrm{d}\varphi = \sqrt{\pi} \frac{\Gamma(\frac{k+1}{2})}{\Gamma(\frac{k}{2}+1)}$, we obtain

$$Q = \frac{2\pi^{(n-2)/2} \int_0^d t^{n-3} \left(\arcsin \frac{\sqrt{R_1^2 - t^2}}{\sqrt{1-t^2}} - \arcsin \frac{\frac{By_0}{A} - \frac{x_0}{\sqrt{A}} \sqrt{R_2^2 - t^2}}{\sqrt{1-t^2}} \right) \mathrm{d}t}{\Gamma(\frac{n-2}{2})},$$

and

$$\Omega_{\mathrm{n}}(\gamma_1, \gamma_2) = \frac{Q}{|S^n|} = \frac{n-2}{2\pi} \int_0^d t^{n-3} \left(\arcsin \frac{\sqrt{R_1^2 - t^2}}{\sqrt{1-t^2}} - \arcsin \frac{\frac{By_0}{A} - \frac{x_0}{\sqrt{A}} \sqrt{R_2^2 - t^2}}{\sqrt{1-t^2}} \right) \mathrm{d}t.$$

When $t \in [0, d]$, the difference of the two anti-trigonometric function is bounded and positive which is independent of n. More precisely, the function

$$f(t) = \arcsin \frac{\sqrt{R_1^2 - t^2}}{\sqrt{1-t^2}} - \arcsin \frac{\frac{By_0}{A} - \frac{x_0}{\sqrt{A}} \sqrt{R_2^2 - t^2}}{\sqrt{1-t^2}}$$

is decreasing when $t \in [0, d]$. We have

$$\Omega_n(\gamma_1, \gamma_2) \ge \frac{n-2}{2\pi} \int_0^{d-\varepsilon} t^{n-3} f(t) \mathrm{d}t \ge \frac{d^{n-2}}{2\pi} (1 - \frac{\varepsilon}{d})^{n-2} f(d - \varepsilon).$$

And since the omitted part which the integral region is from $d-\varepsilon$ to d is negligible compared to that from 0 to $d-\varepsilon$, our estimate is tight. Let $\varepsilon = \frac{d}{n}$, using Taylor series to estimate $f(d-\varepsilon)$, we have $f(d-\varepsilon) = \Theta(\frac{1}{n})$. Also $(1-\frac{1}{n})^{n-2} \ge (1-\frac{1}{n})^n \approx e^{-1}$ when n is sufficient large. Based on the above discussion, $\Omega_n(\gamma_1, \gamma_2) \ge \frac{cd^{n-2}}{2\pi n}$.

Next, given γ_1 and γ_2, we compute the minimum d with the variables α_1, x_0, y_0. Because x_0,y_0 satisfy the equation $(x_0 - \alpha_1)^2 + y_0^2 = \gamma_1^2$, let $\alpha_2 = \sqrt{x_0^2 + y_0^2}$, then

$$x_0 = \frac{\alpha_2^2 + \alpha_1^2 - \gamma_1^2}{2\alpha_1}, \quad y_0 = \sqrt{\alpha_2^2 - \left(\frac{\alpha_2^2 + \alpha_1^2 - \gamma_1^2}{2\alpha_1} \right)^2}.$$

So d can be regarded as a function with two variables α_1 and α_2, and $\gamma_2 \le \alpha_1 \le 1$, $\gamma_2 \le \alpha_2 \le 1$. By calculating the partial derivative $\frac{\partial d(\alpha_1,\alpha_2)}{\partial \alpha_2}$, from $\sqrt{\frac{2}{3}} < \gamma_2 \le \alpha_1 < 1 < \gamma_1 < \sqrt{2}\gamma_2$, it can be proven that, d is a decreasing function with respect to α_2. Let $\alpha_2 = 1$, we get

$$d = \gamma_2 \sqrt{1 - \frac{\gamma_2^2}{4T^2}}, \quad T = \sqrt{1 - \left(\frac{1 + \alpha_1^2 - \gamma_1^2}{2\alpha_1}\right)^2}.$$

It is obvious that d decreases with α_1. Let $\alpha_1 = 1$, we achieve the minimum d.

$$d_{\min} = \gamma_2 \sqrt{1 - \frac{\gamma_2^2 c_{\mathcal{H}_1}^2}{4}}, \quad c_{\mathcal{H}_1} = \frac{1}{T} = \frac{1}{\gamma_1 \sqrt{1 - \frac{\gamma_1^2}{4}}}.$$

The proof of Lemma 4 is completed. □

Now we prove Theorem 2.

Proof. Combing the Lemma 3 and Lemma 4, we get

$$\frac{\Omega_n(\gamma_1, \gamma_2)}{\Omega_n(\gamma_1)} \ge \frac{c}{\sqrt{2\pi n}} \left(1 - \frac{\gamma_1^2}{2\gamma_2^2}\right) \left(\frac{d_{\min}}{c_{\mathcal{H}_2}}\right)^n,$$

which reflects the fraction of M covered by a small ball with radius γ_2 centered in M.

Similar to Theorem 1, it is easy to know the center points C_2^c of the second level in every big ball are less than $c' n^{\frac{3}{2}} (\frac{c_{\mathcal{H}_2}}{d_{\min}})^n$. □

Theorem 3. *The time complexity of our algorithm is $N_{C_1}^2 N_{C_2^c} + N_{C_1} N_{C_2^c}^2$, while the space complexity is $N_{C_1} N_{C_2^c}$. When $\gamma_2 \longrightarrow 1, \gamma_1 = 1.0927$, we get the optimal time complexity $2^{0.3836n}$, and the space complexity $2^{0.2557n}$.*

Proof. The total number of point centers in C_2 is about $N_{C_1} N_{C_2^c}$. If sampling $poly(n) N_{C_1} N_{C_2^c}$ vectors, after a polynomial iterations, we expect the vector left is enough to include the shortest vector. So the space complexity is $\text{poly}(n) N_{C_1} N_{C_2^c}$.

The initial sampling size S is $poly(n) N_{C_1} N_{C_2^c}$. In each iteration, steps 3-17 in algorithm 3 repeat $N_{C_1} N_{C_2^c}$ times, in every repeat, at most $N_{C_1} + N_{C_2^c}$ comparisons are needed. So the total time complexity is $N_{C_1}^2 N_{C_2^c} + N_{C_1} N_{C_2^c}^2$ polynomial computations.

Because N_{C_1} only depends on γ_1, and $N_{C_2^c}$ decreases with γ_2, we obtain the minimum time complexity by selecting $\gamma_2 \longrightarrow 1$ and $N_{C_1} = N_{C_2^c}$ which leads to $\gamma_1 = 1.0927$ and $N_{C_1} = N_{C_2^c} = 2^{0.1278n}$. □

5 Conclusion

In this paper, we describe a new algorithm of heuristic sieve for solving the shortest vector problem with $2^{0.3836n}$ polynomial time operations and $2^{0.2557n}$ space. Although our algorithm decreases the index of the time complexity from 0.415 to 0.3836, the polynomial part of the time complexity increases to $n^{4.5}$ from that of n^3 in NV algorithm. So our algorithm performs better than NV algorithm for large n.

Acknowledgments

We thank Guangwu Xu for revising the paper during his stay in Tsinghua University.

REFERENCES

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory,* 48(8):2201-2214. August 2002.

[2] L.M.Adleman. On breaking generalized knapsack public key cryptosystems. In *the 15th Annual ACM Symposium on Theory of Computing Proceedings*, pages 402-412. ACM, April 1983.

[3] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *the 29th Annual ACM Symposium on Theory of Computing,* pages 284-293. ACM, May 1997.

[4] M. Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *the 30th Annual ACM Symposium on Theory of Computing Proceedings,* pages 10-19. ACM, May 1998.

[5] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *the 33th Annual ACM Symposium on Theory of Computing Proceedings,* pages 266-275. July 2001.

[6] M. Ajtai. Generating hard instances of lattice problems. *Complexity of Computations and Proofs, Quaderni di Matematica,* 13:1-32, 2004. Preliminary version in STOC 1996.

[7] D.Boneh, G.Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. In *Advances in Cryptology - EUROCRYPT 1999 Proceedings,* pages 1-11. Springer, May 1999.

[8] K. Böröczky and G. Wintsche. Covering the sphere by equal spherical balls. *Discrete and Computational Geometry, The Goodman-Pollack Festschrift,* 237-253,2003.

[9] D. Coppersmith. Finding a small root of a univariate modular equation, In *Advances in Cryptology - EUROCRYPT 1996 Proceedings*, pages155-165. Springer, May 1996.

[10] H. Cohen. *A Course in Computational Algebraic Number Theory, 2nd edition.* Springer-Verlag, 1995.

[11] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation,* 44(170):463-471, 1985.

[12] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *the 40th Annual ACM Symposium on Theory of Computing Proceedings,* pages 197-206. ACM, May 2008.

[13] N. Gama and P. Q. Nguyen. Finding short lattice vectors within mordell's inequality. In *the 40th Annual ACM Symposium on Theory of Computing Proceedings,* pages 207-216. ACM, May 2008.

[14] N. Gama, P. Q. Nguyen and O. Regev. Lattice enumeration using extreme pruning. In *Advances in Cryptology - EUROCRYPT 2010 Proceedings,* pages 257-278. Springer, May 2010.

[15] C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objects de la théorie des nombres, deuxième lettre. *Journal fur die reine und angewandte mathematik,* 40:279-290,1850. Also available in the first volume of Hermite's complete works, published by Gauthier-Villars.

[16] G. Hanrot and D. Stehlé. Improved analysis of kannan's shortest lattice vector algorithm. In *Advances in Cryptology - CRYPTO 2007 Proceedings,* pages 170-186. Springer, August 2007.

[17] G. Kabatiansky and V. Levenshtein. Bounds for packings on a sphere and in space. Problemy Peredachi Informatsii, 14(1):3-25, 1978.

[18] P.Klein. Finding the closest lattice vector when it's unusually close. In *the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*,pages 937-941. SIAM, January 2000.

[19] R. Kannan. Improved algorithms for integer programming and related lattice problems. In *the 15th Annual ACM Symposium on Theory of Computing Proceedings,* pages 193-206. ACM, April 1983.

[20] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen,* 261:513-534, 1982.

[21] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the ACM,* 32(1): 229-246, 1985.

[22] D. Micciancio and O. Regev. *Post-Quantum Cryptography,* chapter Lattice-based Cryptography. Springer-Verlag, 2008.

[23] D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *the 21th Annual ACM-SIAM Symposium on Discrete Algorithms Proceedings,* pages 1468-1480. SIAM, January 2010.

[24] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In *the 42th Annual ACM Symposium on Theory of Computing Proceedings,* pages 351-358. ACM, June 2010.

[25] H. Minkowski. *Geometrie der Zahlen.* Teubner, Leipzig, 1896.

[26] P. Q. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto 1997. In *Advances in Cryptology - CRYPTO 1999 Proceedings,* pages 288-304. Springer, August 1999.

[27] P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Cryptography and Lattices Conference 2001,* pages 146-180. Springer, March 2001.

[28] P. Q. Nguyen and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology,* 2(2):181-207, July 2008.

[29] X. Pujol and D. Stehlé. Solving the shortest lattice vector problem in time $2^{2.465n}$. *Cryptology ePrint Archive, Report 2009/605,* 2009.

[30] O. Regev. Lecture notes on lattices in computer science, 2004. *Available at http://www.cs.tau.ac.il/odedr/teaching/lattices fall 2004/index. html.*

[31] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *the 37th Annual ACM Symposium on Theory of Computing Proceedings,* pages 84-93. ACM, May 2005.

[32] A.Shamir. A polynomial time algorithm for breading the basic Merkel-Hellman cryptosystem. In *the 23rd IEEE Symposium On Foundations of Computer Science Proceedings*, pages 145-152. IEEE,1982.

[33] C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science,* 53:201-224, 1987.

[34] C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematics of Programming,* 66:181-199, 1994.

[35] N. Sommer, M. Feder, and O. Shalvi. Finding the closest lattice point by iterative slicing. *SIAM Journal on Discrete Mathematics,* 23(2):715-731, April 2009.

[36] M. Pohst. On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *ACM SIGSAM Bulletin,* 15(1):37-44, 1981.