Modular Form Approach to Solving Lattice Problems[*]

Tian Yuan[†], Zhu Xueyong[†] and Sun Rongxin[†]

Abstract We construct new randomized algorithms to find the exact solutions to the shortest and closest vector problems (SVP and CVP) in Euclidean norm (ℓ^2) for integral lattices. Not only the minimal ℓ^2-norm of non-zero lattice vectors in SVP and the minimal ℓ^2-distance in CVP, but also how many lattice vectors reach those minimums can be simultaneously computed by the algorithms. Our approach is based on special properties of the generating function of lattice vectors' ℓ^2-norms, the lattice-associated theta function, which is used in prior works mainly for hardness analysis on lattice problems but rarely for computational purposes. Such function's modular properties are exploited to develop our SVP and CVP solvers. In computational complexity perspective and take our SVP solver as an example, for the integral lattice family $\{\Lambda_n\}$ of dimension $dim\Lambda_n = n$ and level $h_n = l(\Lambda_n)$ (the minimal positive integer such that the dual lattice Λ_n^* scaled by $h_n^{1/2}$ is integral) polynomial in n, this algorithm can find the minimal ℓ^2-norm of non-zero lattice vectors and the number of such shortest vectors in Λ_n with success probability 1-ε in the asymptotic space-complexity of polynomial in n and asymptotic time-complexity of $n^{O(n)}\log(1/\varepsilon)$. In addition, the only contribution to the algorithm's exponential time complexity $n^{O(n)}\log(1/\varepsilon)$ comes from independently repeating a randomized lattice vector sampler $n^{O(n)}\log(1/\varepsilon)$ times. All the rest of operations contribute to the algorithm's time-complexity only with an additive polynomial in n. Similar situations occur when solving the exact CVP by our algorithm. As a result, our solvers can be easily parallelized to be polynomial in time complexity, and a variant of our CVP solver can solve the closest vector problem with preprocessing (CVPP) in polynomial time and $n^{O(n)}\log(1/\varepsilon)$ space complexity.

Keywords Lattice Algorithms, SVP, CVP, Modular Forms, Theta Function

1 Introduction

Lattice problems take important roles in combinatorial optimization, public-key cryptography and many other fields in computer science [5, 7, 9–12, 17, 19]. In the shortest lattice vector problem (SVP), a

[*] This work is supported by China NSF 61272173

[†] Network Department, Software School, Dalian University of Technology, P.R.China. Email:tianyuan_ca@sina.com xueyongzhu409@gmail.com and sunrongxin7666@163.com

non-zero lattice vector $\boldsymbol{x}$ in $\mathbf{B}\mathbb{Z}^n$ is to be found to minimize $|\boldsymbol{x}|$ on input the lattice basis matrix $\mathbf{B}$ with respect to some specific norm $||$ in $\mathbb{R}^n$. In the closest lattice vector problem (CVP), a lattice vector $\boldsymbol{x}$ is to be found to minimize $|\boldsymbol{u}-\boldsymbol{x}|$ on input the basis matrix $\mathbf{B}$ and a target vector $\boldsymbol{u}$ in $\mathbb{R}^n$. In recent years, lots of cryptographic schemes and protocols have been devised with proofs of security under the assumption that there is no (probabilistic and sometimes quantum) polynomial-time algorithm to solve arbitrary instances of variants of SVP and CVP.

From a computational hardness perspective, SVP, CVP and other related variants are NP-hard under deterministic (e. g., CVP) or randomized (e. g., SVP) reductions [1,10,17,22]. Even some approximation variants of these problems are proven to be NP-hard if the approximation factor is within some specific range. Despite of these facts, finding new algorithms to solve lattice problems exactly are still interesting and meaningful both because many applications (e. g., in mathematics and communication theory) involve lattices in relatively small dimensions, and because approximation algorithms for high dimensional lattices for which the exact solution is infeasible typically involve the exact solution of low dimensional sub-problems. In this paper we develop randomized algorithms to find the exact solutions to SVP and CVP.

1.1 Basic Results

We develop new randomized algorithms to find the exact solutions to SVP and CVP in Euclideans norm (ℓ^2) for any integral lattice. Not only the minimal ℓ^2-norm of non-zero lattice vectors in SVP and the minimal ℓ^2-distance in CVP, but also how many lattice vectors reach those minimums(e. g., the kissing number in SVP) can be simultaneously computed by the algorithms. More concretely and take SVP as an example, for the integral lattice family $\{\Lambda_n\}$ which dimension $dim\Lambda_n = n$ and level $h_n = l(\Lambda_n)$ (the minimal positive integer such that the dual lattice $\Lambda_n{}^*$ scaled by $h_n^{1/2}$ is integral) is polynomial in n, the case frequently occurring in applications, this algorithm can find the minimal ℓ^2-norm of non-zero lattice vectors and the number of such shortest vectors in Λ_n with success probability $1-\varepsilon$ in the asymptotic space-complexity of polynomial in n and asymptotic time-complexity of $n^{O(n)}\log(1/\varepsilon)$. Interestingly, the only contribution to the algorithm's exponential time complexity $n^{O(n)}\log(1/\varepsilon)$ comes from independently repeating a randomized lattice vector sampler $n^{O(n)}\log(1/\varepsilon)$ times. All the rest of operations contribute to the time-complexity with only an additive polynomial in n. Similar situations occur when solving the exact CVP by our algorithm. As a result, our solvers can be (very easily) parallelized to be polynomial in time-complexity. Due to the same feature, a variant of our CVP solver can solve the closet lattice vector problem with preprocessing (CVPP) in polynomial time and $n^{O(n)}\log(1/\varepsilon)$ space complexity.

1.2 A Sketch on Our Approach

Our approach is based on some special properties of the generating function of lattice vectors' ℓ^2-norms. This function is a measure used in previous works mainly for hardness analysis on lattice and related problems [1, 4, 18] but rarely for computational purposes. For SVP, such function is defined as:

$$\vartheta(\tau;\Lambda) \equiv \sum_{\vec{x}\in\Lambda} \exp(2\pi i\tau|\vec{x}|^2)$$

where $|\boldsymbol{x}|$ denotes the vector $\boldsymbol{x}$'s ℓ^2-norm and $\tau = \sigma + it$ is a complex variable on the upper-half complex plane(i. e., $t > 0$). If Λ is integral, i. e., all $|\vec{x}|^2$s are integers for any $\vec{x}$ in Λ (an assumption without any loss in generality when we only deal with rational lattices), this function can be equivalently represented as a Fourier expansion (with complex variable τ)

$$\vartheta(\tau;\Lambda) = \sum_{m\geq 0} a(m)\exp(2\pi i\tau m)$$

where $a(0) = 1$ and $a(m)$ is the number of lattice vectors in Λ which squared ℓ^2-norms equal m. From this viewpoint, solving SVP on Λ reduces to finding its theta function's first non-zero Fourier coefficient $a(m)$ among its non-constant items.

The technical support to the above idea comes from the fact that, as a function of complex variable $\tau(Im\tau > 0)$, $\theta(\tau;\Lambda)$ is a so-called modular form of weight $n/2$ (details in section 2.2) and therefore has a series of special properties. The modularity comes from its transformation law

$$\vartheta(\tau+1;\Lambda) = \vartheta(\tau;\Lambda)$$
$$\vartheta(\tau/(4h\tau+1);\Lambda) = (4h\tau+1)^{n/2}\vartheta(\tau;\Lambda)$$

where h is some positive integer, the level of Λ. As a result, the theta function can be expanded on a polynomial (in the lattice's level h and dimension n) number of base functions and then its Fourier coefficients $a(m)$ can be efficiently computed from the linear combination of a set of the basis' Fourier coefficients.

For CVP, when restricting the target vector $\boldsymbol{u}$ to be the integral vector (without any loss in generality when we only work in the rational number field), the same idea applies to the non-homogenous theta function

$$\vartheta(\tau;\Lambda,\vec{u}) \equiv \sum_{\vec{x}\in\Lambda} exp(2\pi i\tau|\vec{x}-\vec{u}|^2) = \sum_{m\geq 1} b(m)exp(2\pi i\tau m)$$

which is also a modular form, where $b(0) = 0$ (except for the trivial case that $\boldsymbol{u} \in \Lambda$) and $b(m)$ is the number of lattice vectors in Λ which squared ℓ^2-distance to $\boldsymbol{u}$ is m. From this viewpoint, solving CVP on input Λ and $\boldsymbol{u}$ reduces to finding the non-homogenous theta function's first non-zero Fourier coefficient $b(m)$.

1.3 Related Works

To find the exact solutions to lattice problems, so far three main families of SVP and CVP solvers exist which are listed in Table 1 together with our algorithms developed in this paper in comparison.

Among these solvers, MV and Kannan algorithms are deterministic while AKS (and our) algorithms are randomized. All algorithms work in ℓ^2-norm (only AKS algorithm can work in other norms, e. g., ℓ_∞). The core of MV algorithm [16] is to compute the Voronoi cell of the lattice [5], whose knowledge facilitates the tasks to solve SVP and CVP. Kannan algorithm [11,12] relies on a deterministic procedure to enumerate all lattice vectors below a prescribed norm, or within a prescribed distance to the target vector. This procedure uses the Grahm-Schmidt orthogonalization of the input lattice basis to recursively bound the integer coordinates of the candidate solutions.

The AKS algorithm [2] was the first single-exponential time algorithm for SVP which can be described as follows: Let $\gamma < 1$ be a constant and S be a set of N lattice vectors sampled in the ℓ^2-ball of radius $R = 2^{O(n)}\lambda_1(\Lambda)$ where $\lambda_1(\Lambda)$ is the minimal norm of non-zero lattice vectors in Λ. For sufficiently large N, there exists a pair of lattice vectors $\boldsymbol{u}$, $\boldsymbol{v}$ such that $|\boldsymbol{u}-\boldsymbol{v}| < \gamma R$, so $\boldsymbol{u}-\boldsymbol{v}$ is shorter in Λ. The core of the algorithm is to chose a subset C in S such that $|C|$ is not too large and for any $\boldsymbol{u}$ in $S\backslash C$ there exists $\boldsymbol{v}$ in C such that $|\boldsymbol{u}-\boldsymbol{v}| < \gamma R$. This is used to produce a set of lattice vectors S_1 in the ball $\gamma R B_2^n$ with $|S_1| = |S| - |C|$. This procedure can be applied a polynomial number of times to obtain lattice vectors of norms less than $a\lambda_1(\Lambda)$ for some constant a. Recently this algorithm has been significantly improved and the currently best time complexity is $2^{2.465n+o(n)}$ [9]. However, the AKS variant solver for CVP only finds the $(1+\varepsilon)$-approximate solution for arbitrary $\varepsilon > 0$ in time complexity bounded by $(2+1/\varepsilon)^{O(n)}$ [3,20].

As a randomized algorithm, our solver outperforms the sieve algorithms in the aspects that it has space complexity only polynomial in n and can solve both SVP and CVP precisely. Another characteristic of our algorithm is its ability to be parallelized to be polynomial in time complexity. As noticed in section 4.1, when sampling the lattice by calling N independent Gaussian samplers in concurrency rather than in sequence, the whole algorithm to solve SVP or CVP becomes polynomial in time complexity (but exponential in parallelism). Another variant of our CVP solver can solve CVPP in polynomial time and $n^{O(n)}log(1/\varepsilon)$ space complexity with success probability 1-ε. Such characteristics will be valuable in practices, e. g., in solving lattice problems of moderately high dimensions. So far with our understanding no other solvers can be parallelized to be polynomial in time complexity. For example, the critical component in the elegant MV algorithm [16] is an iterative subroutine to operate at most 2^n times, which is hard to be parallelized due to its iterative nature. The core of AKS algorithm and its variants [2,3,20], the sieve subroutine which dominates the algorithm's time complexity, is also hard to be parallelized to be polynomial. Similar situations occur for Kannan algorithm. From this viewpoint, an interesting and importnat open question is: is there any parallel (deterministic or randomized) SVP/CVP solver which is of polynomial time complexity in n with $2^{O(n)}$ processors? The last (but not the least) important feature of our approach is its potential to apply to SVP and CVP for the ideal lattices in algebraic number field where the theta functions have more special properties to exploit. Table 1 summarizes the main features of these algorithms in comparison with each other.

1.4 Roadmap

In section 2 we give necessary backgrounds in lattice geometry and modular forms. In section 3, we give a sketch on our approach which technical details are elaborated in section 4. The complete algorithms to solve SVP and CVP are presented in section 4.3 and the complexity analysis is given in section 5. In section 6, we discuss some extensions from our approach to solve more generalized or specialized problems.

Table 1: Comparing the existed families of SVP and CVP solvers and our algorithms

Solvers	***Time complexity upper bound***	***Space complexity upper bound***	***Remarks***
Kannan [9, 11, 12]	$n^{O(n)}$	$poly(n)$	deterministic; the O-constant is improved as small as 1/2e
MV [9, 16]	$2^{2n+o(n)}$	$2^{O(n)}$	deterministic
AKS [2, 3, 9, 20]	SVP: $2^{2.465n+o(n)}$ CVP: $(2+1/\varepsilon)^{O(n)}$	SVP: $2^{1.325n+o(n)}$ CVP: $(1+1/\varepsilon)^{O(n)}$	randomized; solves $(1+\varepsilon)$-CVP only
Our algorithm	$n^{O(n)}$	$poly(n)$	for lattice level $h = poly(n)$
	$(nh)^{O(n)}$	$poly(n)$	for arbitrary lattice level h The O-constant is $2+\delta$,$\delta > 0$. Easy to be parallelized to be polynomial-time to solve SVP, CVP and CVPP.

2 Preliminaries

2.1 Lattices

General: The set of integers is denoted by $\mathbb{Z}$ and rational numbers by $\mathbb{Q}$. In the Euclidean space $\mathbb{R}^n$, a n-dimensional rational lattice, denoted $\Lambda(\mathbf{B})$ where $\mathbf{B}$ is a matrix with column vectors $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$, is the set of vectors $\{x_1\boldsymbol{b}_1 + \ldots + x_n\boldsymbol{b}_n : x_1, \ldots, x_n \in \mathbb{Z}\}$ where the scalar products $< \boldsymbol{b}_i, \boldsymbol{b}_j >$ are all rational numbers. The lattice with basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is also denoted $Z\boldsymbol{b}_1 + \ldots + Z\boldsymbol{b}_n$. Without loss of generality in computer science, in this work we only consider the integral lattice in which $< \boldsymbol{b}_i, \boldsymbol{b}_j >$ are all integers.

For any lattice $\Lambda = Z\boldsymbol{b}_1 + \ldots + Z\boldsymbol{b}_n$, the lattice $\Lambda^* \equiv Z\boldsymbol{b}_1^* + \ldots + Z\boldsymbol{b}_n^*$ where $< \boldsymbol{b}_i^*, \boldsymbol{b}_j >= \delta_{ij}$ for all $i, j = 1, \ldots, n$ is called Λ's dual lattice. Equivalently, Λ^* is a discrete set of vectors $\mathbf{y}$ such that $< \boldsymbol{x}, \boldsymbol{y} > \in Z$ for all $\boldsymbol{x}$'s in Λ. The dual Λ^* of a rational lattice Λ is always rational, but Λ^* may not be integral even when Λ is integral. When Λ has a base matrix $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$, its dual lattice Λ^* will have a base matrix $\Lambda^* = (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*) = \mathbf{B}^{-T}$ so both Λ and Λ^* are integral *iff* $det(\mathbf{B}) = det(\mathbf{B}^*) = \pm 1$. Another important property is that $\Lambda^{**} = \Lambda$.

For any vector $\boldsymbol{u} = (u_1, \ldots, u_n)$ in $\mathbb{R}^n$, its ℓ^2-norm $< \boldsymbol{u}, \boldsymbol{u} >^{1/2} = (u_1^2 + \ldots + u_n^2)^{1/2}$ is denoted $|\boldsymbol{u}|$. The squared ℓ^2-norm of any lattice vector in an integral lattice is always an integer.

Lattice Problems: Given a lattice $\Lambda(\boldsymbol{B}) = Z\boldsymbol{b}_1 + \ldots + Z\boldsymbol{b}_n$, let

$$\lambda_1(\Lambda) \equiv \min\{|\boldsymbol{x}| : \boldsymbol{x} \text{ } in \text{ } \Lambda \text{ and non-zero}\} \tag{2.1}$$

be the minimal value of ℓ^2-norms of non-zero lattice vectors in $\Lambda(\mathbf{B})$. The *optimization* (ℓ^2-) *shortest vector problem*, SVP(Λ) in brief, is to find $\lambda_1(\Lambda)$. The *search* (ℓ^2-) shortest vector problem, s-SVP(Λ) in brief, is to find a lattice vector $\boldsymbol{x}$ in Λ such that $|\boldsymbol{x}| = \lambda_1(\Lambda)$.

Given a lattice $\Lambda(\boldsymbol{B})$ and a rational target vector $\boldsymbol{u}$ in $\mathbb{Q}^n$, let

$$dist(\Lambda; \boldsymbol{u}) \equiv \min\{|\boldsymbol{x} - \boldsymbol{u}| : \boldsymbol{x} \text{ } in \text{ } \Lambda\} \tag{2.2}$$

be the minimum ℓ^2-distance between $\boldsymbol{u}$ and all lattice vectors in Λ. The *optimization* (ℓ^2-)*closest vector problem*, CVP(Λ,$\boldsymbol{u}$) in brief, is to find $dist(\Lambda;\boldsymbol{u})$. The *search* ($\ell^2$-)closest vector problem, s-CVP(Λ,$\boldsymbol{u}$) in brief, is to find a lattice vector $\boldsymbol{x}$ in Λ such that $|\boldsymbol{x} - \boldsymbol{u}| = dist(\Lambda; \boldsymbol{u})$.

The covering radius of a lattice, $\mu(\Lambda)$, is defined as the maximal distance between any vector and the lattice. The covering radius problem, CRP(Λ) in brief, is to find

$$\mu(\Lambda) \equiv \max\{dist(\Lambda; \boldsymbol{u}) : \boldsymbol{u} \text{ } in \text{ } \mathbb{Q}^n\} \tag{2.3}$$

In this paper we focus on the algorithm to solve SVP and CVP problems. It has been known that these problems are computationally hard [1,10,15,17,22]. However, there is:

Theorem 2.1. *[15, 17, 22] (1)s-SVP can be solved in polynomial time given the oracle to solve s-CVP. (2)s-CVP can be solved in polynomial time given the oracle to solve (optimization) CVP.* □

In consequence, the algorithm for optimization CVP can be used as the cornerstone to solve both search problems. In this paper we focus on constructing the randomized algorithms for optimization SVP and CVP with similar ideas and techniques.

General Bounds: For any n-dimensional lattice Λ, one of the most important general fact is the Mincowski's inequality [5,22]:

$$Vol(B_2^n)\lambda_1(\Lambda)^n \le 2^n|det(\Lambda)|$$

where $Vol(B_2^n)$ is the n-dimensional volume of the unit Euclidean ball B_2^n , e.g., $\pi^{n/2}/(n/2)!$, and $|det(\Lambda)|$ is the determinant of the lattice's base matrix $\mathbf{B}$, numerically equal to the lattice's elementary parallelotope's volume. It follows that

$$\lambda_1(\Lambda) \le cn^{1/2}|det(\Lambda)|^{1/n} \tag{2.4}$$

where $c(\le 1)$ is some absolute constant.

Another important general property is the transference theorem [4,22]

$$\lambda_1(\Lambda^*)\mu(\Lambda) \le dn \tag{2.5}$$

where $d(\le 1/2)$ is some absolute constant. In particular, let h be some positive integer such that the lattice $h^{1/2}\Lambda^*$ is integral, then due to $\lambda_1(h^{1/2}\Lambda^*) \ge 1$ we have

$$\mu(\Lambda) \le dn/\lambda_1(\Lambda^*) \le dnh^{1/2}/\lambda_1(h^{1/2}\Lambda^*) \le dnh^{1/2} \tag{2.6}$$

Lattice Level: Let Λ be an integral lattice. In this case the dual lattice Λ^* is rational so there exists a positive integer h such that $h^{1/2}\Lambda^*$ is integral.

Definition 2.1. *Given an integral lattice Λ, the level of this lattice, denoted $l(\Lambda)$, is defined as the minimal positive integer* h *such that $h^{1/2}\Lambda^*$ is integral.* □

It's easy to see that $l(\Lambda)$ is an invariant of Λ, i. e., independent of Λ's basis choice.

Let $\mathbf{B}$ and $\mathbf{B}^*$ be Λ's and Λ^*'s base matrix respectively (so $\mathbf{B}^* = \mathbf{B}^{-T}$), so the dual lattice Λ^*'s Grahm matrix $\mathbf{A}^* = \mathbf{B}^{*T}\mathbf{B}^* = \mathbf{B}^{-1}\mathbf{B}^{-T} = \mathbf{A}^{-1}$, the inverse of the lattice Λ's Grahm matrix. Notice that $h^{1/2}\Lambda^*$ is integral means that $h\mathbf{A}^*$ is an integral matrix, and since $(det\mathbf{A})\mathbf{A}^* = \mathbf{A}^{adj}$ is always integral (because the adjoint matrix $\mathbf{A}_{adj}$'s entries are all integers), it follows that $h|det(\mathbf{A})$. On the other hand, the fact that $\mathbf{M} = h\mathbf{A}^*$ is an integral matrix deduces that $h\mathbf{I} = \mathbf{MA}$ and then we have $det\mathbf{A}|h^n$. In summary, the level h satisfies $h|(det\Lambda)^2|h^n$.

Moreover, the level h can be computed by $h = det(\mathbf{A})/g$ where $g = \gcd(\mathbf{A}^{adj})$=the greatest common divisor of all the entries in $\mathbf{A}$'s adjoint matrix $\mathbf{A}^{adj}$.

2.2 Modular Forms

In this section we present a very brief description about its concepts and facts of one-variable modular forms needed in our work.

General: Let

$$SL_2(\mathbb{Z}) \equiv \{\gamma = \begin{bmatrix} a & b \\ c & d \end{bmatrix} : a, b, c, d \in \mathbb{Z} \ and \ ad - bc = 1\}$$

be the group of 2×2 integer matrices with determinant 1. For any γ in $SL_2(\mathbb{Z})$ there is an related action on the upper-half complex plane $H \equiv \{\sigma + it : t > 0\}$ defined as:

$$\gamma(\tau) \equiv (a\tau + b)/(c\tau + d) : H \to H$$

Notice that $\pm\gamma$ induces the same action $\gamma(\tau)$. $SL_2(\mathbb{Z})$ is a finitely generated group with two generators [6]

$$\gamma_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} , \ \gamma_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

i. e., any action $\gamma(z)$ can be composed by the actions $\gamma_1(\tau) = \tau + 1$ and $\gamma_2(\tau) = -1/\tau$.

Instead of $SL_2(\mathbb{Z})$, in our work we consider its congruence subgroup of a given positive integer N:

$$\Gamma(N) \equiv \{\gamma \in SL_2(\mathbb{Z}) : \gamma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} mod \ N\}, \Gamma_1(N) \equiv \{\gamma \in SL_2(\mathbb{Z}) : \gamma = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} mod \ N\} \tag{2.7}$$

Both $\Gamma(N)$ and $\Gamma_1(N)$ are finite-index subgroups in $SL_2(\mathbb{Z})$ [6, 13].

Definition 2.2. *[6, 24] Let k be some positive integer or half-integer, Γ be a subgroup in $SL_2(\mathbb{Z})$ and $\Gamma(\mathrm{N}) \subseteq \Gamma$, $f(\tau) : H \to C$ be a complex function holomorphic on the upper-half plane H. Let $f[\gamma]_k \equiv (c\tau + d)^{-k} f(\gamma(\tau))$ for γ in $SL_2(\mathbb{Z})$, f is defined as* a modular form of weight k with respect to Γ*, if both the following properties hold:*

(1) $f[\gamma]_k$is bounded at the infinity point, i. e., $\lim_{t\to\infty} |f[\gamma]_k(\sigma+it)|$ exists for any γ in Γ and real number σ;

(2) $f[\gamma]_k = f$, i. e., $f(\gamma(\tau)) = (c\tau + d)^k f(\tau)$ for any γ in Γ and τ in H

□

The set of such functions is a linear space and is denoted $M_k(\Gamma)$.

For $\Gamma = \Gamma(N)$ or $\Gamma_1(N)$, there's always an integer w such that

$$\begin{bmatrix} 1 & w \\ 0 & 1 \end{bmatrix} \in \Gamma$$

As a result, $f(\tau + w) = f(\tau)$ for any f in $M_k(\Gamma)$ hence there is always the Fourier expansion

$$f(\tau) = \sum_{m=0}^{\infty} a(m) q^{m/w} \; where \; q = \exp(2\pi i \tau)$$

Example:Consider the case $\Gamma = SL_2(\mathbb{Z})$, then $f(\tau)$ is in $M_k(SL_2(\mathbb{Z}))$ iff it satisfies the above conditions (1) and (2) for all γ's in $SL_2(\mathbb{Z})$. Because $SL_2(\mathbb{Z})$ is generated by two generators $\gamma_1(\tau) = \tau + 1$ and $\gamma_2(\tau) = -1/\tau$, the modularity condition (2) is equivalent to the transformation law $f(\tau + 1) = f(\tau)$ and $f(-1/\tau)) = (1/\tau)^k f(\tau)$.

Finiteness of Modular Form Space's Dimension: The transformation law under the congruence group's action imposed on the modular forms is a very strong restriction, so strong as to imply lots of special properties of the modular forms. One of the most important consequences followed is that the function space $M_k(\Gamma)$ is finite dimensional.

Theorem 2.2. *[6, 14, 24] For any positive integer N, positive integer or half-integer k, $M_k(\Gamma)$ is a finite-dimensional linear space on the complex field with $dim_C M_k(\Gamma) \leq dim_C M_k(\Gamma(N)) = 8kN^3/3$.* □

Remarks: More precisely, for any integer $k \geq 3$ [6, 14]:

$$dim_C M_k(\Gamma(N)) = (\frac{N}{24}(k-1) + \frac{1}{4})N^2 \prod_{p|N,primes} (1 - \frac{1}{p^2})$$

$$dim_C M_k(\Gamma_1(N)) = \frac{N^2}{24}(k-1) \prod_{p|N,primes} (1 - \frac{1}{p^2}) + \frac{1}{4} \prod_{d|N} \varphi(d)\varphi(N/d)$$

where $\varphi(n)$ is the Euler function. When k is half-integer, the dimension formulas are more complicated but the asymptotic relations with k and N are of the same type,i. e., $O(kN^3)$. Details can be seen in, e. g., [24].

2.3 Lattice-Associated Theta Function and Its Modularity

One of the relations between (integral) lattices and modular forms is through the theta function, defined as

$$\vartheta(\tau; \Lambda) \equiv \sum_{\vec{x} \in \Lambda} \exp(2\pi i \tau |\vec{x}|^2) \tag{2.8}$$

where || denotes the ℓ^2 norm and $\tau = \sigma + it$ is a complex variable on the upper-half complex plane. Since Λ is integral, its Fourier expansion is

$$\vartheta(\tau;\Lambda) = \sum_{m\geq 0} a(m)q^m,\ \textit{where}\ q = \exp(2\pi i\tau)$$

where $a(0) = 1$ and $a(m)$ is the number of lattice vectors in Λ which squared ℓ^2-norms equal m. From this viewpoint, solving SVP on Λ reduces to finding its theta function's first non-zero Fourier coefficient $a(m)$ among non-constant items.

It's easy to prove that such theta function absolutely and uniformly converges in any compact subset of the upper half-plane H and is bounded at $+i\infty$, as a result, holomorphic on H. Another obvious property is

$$\vartheta(\tau+1;\Lambda) = \vartheta(\tau;\Lambda)\ \textit{due to}\ \Lambda'\textit{s integrality} \tag{2.9}$$

Let $n = dim\Lambda$ and Λ^* be the dual lattice of Λ. By Poisson formula (proven in the Appendix), we have

$$\vartheta(\tau;\Lambda) = (i/2\tau)^{n/2} det\Lambda^* \vartheta(-1/4\tau;\Lambda^*) \tag{2.10a}$$

or equivalently

$$\vartheta(\tau;\Lambda^*) = (i/2\tau)^{n/2} det\Lambda \vartheta(-1/4\tau;\Lambda) \tag{2.10b}$$

Let h be any positive integer such that $h^{1/2}\Lambda^*$ is also an integral lattice. Since $h|y|^2$ is an integer for any y in Λ^*, for any η in H we have

$$\vartheta(\eta+h;\Lambda^*) = \sum_{\vec{y}\in\Lambda^*} \exp(2\pi i(\eta+h)|\vec{y}|^2) = \sum_{\vec{y}\in\Lambda^*} \exp(2\pi i\eta|\vec{y}|^2) = \vartheta(\eta;\Lambda^*)$$

let $\xi \equiv -(h + 1/4\tau)$,then by (2.10a) and the above h-periodicity

$$\begin{aligned}
\vartheta(\tau/(4h\tau+1);\Lambda) &= \vartheta(-1/4\xi;\Lambda) \\
&= (2\xi/i)^{n/2} det\Lambda^* \vartheta(\xi;\Lambda^*) \\
&= (2\xi/i)^{n/2} det\Lambda^* \vartheta(-1/4\tau;\Lambda^*) \\
&= (2\xi/i)^{n/2} det\Lambda^* (2\tau/i)^{n/2} det\Lambda \vartheta(\tau;\Lambda) \\
&= (-4\xi\tau)^{n/2} \vartheta(\tau;\Lambda) \\
&= (4h\tau+1)^{n/2} \vartheta(\tau;\Lambda)
\end{aligned}$$

A more general relation between (integral) lattices and modular forms is through the following parameterized theta function, defined as

$$\vartheta(\tau;\Lambda^*,\vec{u},\vec{v}) = \sum_{\vec{x}\in\Lambda} \exp(2\pi i\tau|\vec{x}-\vec{u}|^2 + 2\pi i < \vec{x},\vec{v} >) \tag{2.11}$$

where || denotes the ℓ^2 norm, u and v are parameter vectors in $\mathbb{R}^n$ and $\tau = \sigma + it$ is a complex variable on the upper-half complex plane. By Poisson formula (proven in the appendix) we have

$$\vartheta(\tau;\Lambda,\vec{u},\vec{v}) = (i/2\tau)^{n/2} det\Lambda^* \exp(2\pi i < \vec{u},\vec{v} >)\vartheta(-1/4\tau;\Lambda^*,\vec{v},-\vec{u}) \tag{2.12a}$$

and equivalently

$$\vartheta(\tau;\Lambda^*,\vec{u},\vec{v}) = (i/2\tau)^{n/2} det\Lambda \exp(2\pi i < \vec{u},\vec{v} >)\vartheta(-1/4\tau;\Lambda,,\vec{v},-\vec{u}) \tag{2.12b}$$

By calculations similar as before, it can be derived that

$$\vartheta(\tau/(4h\tau+1);\Lambda,\vec{u},\vec{v}) = (4h\tau+1)^{n/2} \exp(-4\pi i < \vec{u},\vec{v} >)\vartheta(\tau;\Lambda,-\vec{u},-\vec{v}) \tag{2.13}$$

Let $\vartheta(\tau;\Lambda,\vec{u}) \equiv \vartheta(\tau;\Lambda,\vec{u},0) = \sum_{\vec{x}\in\Lambda} \exp(2\pi i\tau|\vec{x}-\vec{u}|^2)$,the above identity derives that

$$\vartheta(\tau/(4h\tau+1);\Lambda,\vec{u}) = (4h\tau+1)^{n/2}\vartheta(\tau;\Lambda,-\vec{u}) = (4h\tau+1)^{n/2}\vartheta(\tau;\Lambda,\vec{u}) \tag{2.14}$$

In summary, we have proven:

Lemma 2.3. *For any n-dimensional integral lattice Λ, the integer h such that that $h^{1/2}\Lambda^*$ is integral and an integral vector $\boldsymbol{u}$ in $\mathbb{Z}^n$, $\vartheta(\tau;\Lambda,\vec{u})$ is a modular form of weight $n/2$ with respect to the congruence subgroup generated by*

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \textit{and} \quad \begin{bmatrix} 1 & 0 \\ 4h & 1 \end{bmatrix}$$

□

Remarks: Let such generated congruence subgroup be denoted $J(h)$. The lemma states that

$$\vartheta(\tau;\Lambda,\vec{u}) \in M_{n/2}(J(h))$$

Since $\Gamma(4h) \subset J(h) \subset \Gamma_1(4h)$, it follows that $M_{n/2}(\Gamma_1(4h)) \subset M_{n/2}(J(h)) \subset M_{n/2}(\Gamma(4h))$ and by the dimension formulas (theorem 2.2) when n is even we have

$$Anh^2 \le dim_C M_{n/2}(\Gamma_1(4h)) \le dim_C M_{n/2}(J(h)) \le dim_C M_{n/2}(\Gamma(4h)) \le 8nh^3/3 \tag{2.15}$$

where $A < 1$ is some positive absolute constant. When n is odd we have the same upper-bound by the dimension formulas of the space of modular forms with weight half-integer $n/2$ [24].

In practice, h can be selected as the lattice level $l(\Lambda)$, the minimal positive integer such that $h^{1/2}\Lambda^*$ is integral, a lattice invariant which can be efficiently computed (section 2.1).

3 Our Approach's Framework

In this section we present our approach in a heuristic way, leaving technical details in next sections. To make the idea clear and easy to understand, we present this approach at first to solve two basic problems in section 3.1 and section 3.2 then apply the subroutines to solve the optimization SVP and CVP in section 3.

3.1 Basic Problems

Definition 3.1. *Given an integral lattice* $\Lambda(\boldsymbol{B}) = \mathrm{Z}\boldsymbol{b}_1 + \ldots + \mathrm{Z}\boldsymbol{b}_n$ *in* $\mathbb{Q}^n$ *and a positive integer* m*, the* ℓ^2- vector counting problem*,* $VCP(\Lambda, m)$ *in brief, is to find the number of lattice vectors in* $\Lambda(\boldsymbol{B})$ *which squared* ℓ^2*-norms equal* m*, i. e., to find* $a(m) = |\{\boldsymbol{x} \text{ in } \Lambda : |\boldsymbol{x}|^2 = m\}|$*.* □

Definition 3.2. *Given an integral lattice* $\Lambda(\boldsymbol{B}) = \mathrm{Z}\boldsymbol{b}_1 + \ldots + \mathrm{Z}\boldsymbol{b}_n in\ \mathbb{Q}^n$*, a vector* $\mathbf{u}$ *in* $\mathbb{Z}^n$ *such that* $2\boldsymbol{B}\mathbf{u}$ *also in* $\mathbb{Z}^n$*, a positive integer* m*, the* ℓ^2- non-homogenous vector counting problem*,* $n\text{-}VCP(\Lambda, m, \mathbf{u})$ *in brief, is to find the number of lattice vectors in* $\Lambda(\boldsymbol{B})$ *which squared* ℓ^2*-distances to* $\mathbf{u}$ *equal* m*, i. e., to find* $b(m) = |\{\mathbf{x} \text{ in } \Lambda : |\mathbf{x} - \mathbf{u}|^2 = m\}|$*.* □

Remark: As long as both the lattice matrix $\mathbf{B}$ and the target vector $\boldsymbol{u}$ have only rational entries, it's easy to satisfy all the above requirements by scaling the original lattice $\Lambda(\mathbf{B})$ and $\boldsymbol{u}$ simultaneously with some appropriately large integer. In this case, i. e., for an integral lattice $\Lambda(\mathbf{B}) = Z\boldsymbol{b}_1 + \ldots + Z\boldsymbol{b}_n$ in $\mathbb{Q}^n$ and a vector $\boldsymbol{u}$ in $\mathbb{Z}^n$ such that $2\mathbf{B}\boldsymbol{u}$ also in $\mathbb{Z}^n$, the squared distance $|\boldsymbol{x} - \boldsymbol{u}|^2 = |\mathbf{B}\boldsymbol{z} - \boldsymbol{u}|^2 = \boldsymbol{z}^T\mathbf{B}^T\mathbf{B}\boldsymbol{z} - 2\boldsymbol{z}^T\mathbf{B}\boldsymbol{u} + \boldsymbol{u}^T\boldsymbol{u}$ ($\boldsymbol{z}$ in $\mathbb{Z}^n$) is always an integer.

3.2 Solving the Basic Problems

Given an integral lattice $\Lambda(\mathbf{B}) = Z\boldsymbol{b}_1 + \ldots + Z\boldsymbol{b}_n$ of dimension n and a positive integer m, consider how to solve $VCP(\Lambda, m)$ at first. Assume the level of Λ is h. By lemma 2.3 the lattice-associated theta function $\vartheta(\tau; \Lambda)$ (2.8) is in $M_{n/2}(J(h))$, it follows that

$$\vartheta(\tau; \Lambda) = \sum_{\alpha=1}^{M} h_\alpha(\Lambda)\varphi_\alpha(\tau) \tag{3.1}$$

where $M = dimM_{n/2}(J(h))$ and $\varphi_\alpha(\tau)$'s are basis of the space $M_{n/2}(J(h))$. Let

$$\varphi_\alpha(\tau) = \sum_{m=0}^{\infty} a_\alpha(m)q^m, \ where\ q = \exp(2\pi i\tau) \tag{3.2}$$

The basis $\{\varphi_\alpha(\tau) : \alpha = 1, \ldots, M\}$ and therefore their Fourier coefficients $a_\alpha(m)$ only depend on the congruence subgroup $J(h)$, which can be determined even in preprocessing when h is fixed. Then the

m-th Fourier coefficient $a(m)$, i. e., the solution to the problem $VCP(\Lambda, m)$, can be computed by the formula

$$a(m) = \sum_{\alpha=1}^{M} h_\alpha(\Lambda) a_\alpha(m) \tag{3.3}$$

In this viewpoint, as long as the linear combination coefficients $h_\alpha(\Lambda)$ are known, the solution $a(m)$ is obtained.

Then arises the second question: how to compute $\{h_\alpha(\Lambda)\}_{\alpha=1,\ldots,M}$? Suppose we know $M(= dim M_{n/2}(J(h)))$ points $\tau_1, \ldots, \tau_M$ on the upper-half plane H and the values of the theta function at these points, $\theta(\tau_1; \Lambda), \ldots, \theta(\tau_M; \Lambda)$. As long as $det(\varphi_\alpha(\tau_\beta)) \neq 0$, by solving the linear system of equations

$$\vartheta(\tau_\alpha; \Lambda) = \sum_{\beta=1}^{M} h_\beta(\Lambda) \varphi_\beta(\tau_\alpha) \qquad \alpha = 1, \ldots, M \tag{3.4}$$

all of $h_\alpha(\Lambda)$, $\alpha = 1, \ldots, M$ can be efficiently obtained.

Now the third question: for a given lattice Λ and any given point τ on the upper-half plane H , how to determine the value of $\theta(\tau; \Lambda)$? By definition $\theta(\tau; \Lambda)$ depends on the norms of all lattice vectors in Λ including those to be found in question, how to determine such an object prior to determining some of its unknown constituents? It is to solve this (and only this) sub-problem that the randomness in our algorithm is introduced.

The idea is to estimate $\theta(\tau; \Lambda)$ by appropriate random sampling over the lattice Λ. Note that when $t > 0$:

$$1/\vartheta(it; \Lambda) = 1/\sum_{x \in \Lambda} \exp(-2\pi |x|^2 t) = \underset{x \leftarrow D_{\Lambda,1/t}}{E}[\delta(x)]$$

where $\delta(\boldsymbol{x})$ is the delta-function on Λ, vanishing at all non-zero lattice vectors and having the value 1 at $\boldsymbol{x} = 0$:

$$\delta(\boldsymbol{x}) = 1 \ if \ \boldsymbol{x} = 0; \ \delta(\boldsymbol{x}) = 0 \ if \ \boldsymbol{x} \neq 0 \tag{3.5}$$

and $D_{\Lambda,1/t}(\boldsymbol{x})$ is the discrete Gaussian probabilistic distribution over lattice Λ:

$$D_{\Lambda,1/t}(\boldsymbol{x}) \equiv \exp(-2\pi |x|^2 t) / \sum_{x' \in \Lambda} \exp(-2\pi |x'|^2 t) \quad for \ \boldsymbol{x} \ in \ \Lambda$$

As a result, $1/\vartheta(it; \Lambda)$ might be estimated by statistical averaging over a set of $\delta(\boldsymbol{x}_j)$'s where each $\boldsymbol{x}_j$ is a lattice vector independently sampled from Λ with distribution $D_{\Lambda,1/t}$. However, the existed Gaussian samplers [8, 21] requires that t in this case be sufficiently small, potentially incompatible with some other requirements and practical considerations in our algorithm construction. Instead, we consider another way to estimate $\theta(it;\Lambda)$. The starting point is Poisson formula (2.10a):

$$\vartheta(it;\Lambda) = (1/2t)^{n/2} det\Lambda^* \vartheta(i/4t;\Lambda^*)$$

$$1/\vartheta(i/4t;\Lambda^*) = 1/\sum_{y\in\Lambda^*} \exp(-2\pi|y|^2/4t) = \underset{y\leftarrow D_{\Lambda^*,4t}}{E}[\delta(y)] \tag{3.6}$$

where $\delta(\boldsymbol{y})$ is the delta-function on the dual lattice Λ^* and $D_{\Lambda^*,4t}(\boldsymbol{y})$ is the discrete Gaussian distribution over the dual lattice Λ^*:

$$D_{\Lambda^*,4t}(\boldsymbol{x}) \equiv \exp(-2\pi|y|^2/4t)/\sum_{y'\in\Lambda^*} \exp(-2\pi|y'|^2/4t) \quad for\ \boldsymbol{y}\ in\ \Lambda^* \tag{3.7}$$

Borrowing the techniques developed in [8,21], $1/\theta(i/4t;\Lambda^*)$ hence $1/\theta(it;\Lambda)$ can be estimated by efficient random sampling algorithms as long as t is appropriately large. In this case $\theta(it;\Lambda) = O(1)$, i. e. $1/\theta(it;\Lambda)$ is not too small hence $\theta(i/4t;\Lambda^*)$ can be estimated by $1/\theta(i/4t;\Lambda^*)^{-1}$ with sufficiently small errors. Once $\theta(i/4t;\Lambda^*)$ can be estimated with given $t>0$, by the following equation (derived in section 4)

$$\theta(\sigma+it;\Lambda) = (i/2(\sigma+it))^{n/2} det(\Lambda^*)\theta(it/4(\sigma^2+t^2);\Lambda^*) \underset{y\leftarrow D_{\Lambda^*,4(\sigma^2+t^2)/t}}{\boldsymbol{E}}[\exp(-2\pi i\sigma|y|^2/4(\sigma^2+t^2))] \tag{3.8}$$

$\theta(\tau;\Lambda)$ can be estimated at $\tau = \sigma + it(t>0)$ where in the expectation (replaced by statistical average when doing estimation) lattice vectors are distributed with the probability $D_{\Lambda^*,4(\sigma^2+t^2)/t}(y)$ over Λ^*(with appropriately large t). Up to this point, the basic problem $VCP(\Lambda,m)$ is completely solved.

Similar steps are taken to solve the non-homogenous vector counting problem n-$VCP(\Lambda;m,\boldsymbol{u})$ by using Fourier expansions in space $M_{n/2}(J(h))$ and estimating the theta function $\theta(\tau;\Lambda,\boldsymbol{u}) = \sum_{\vec{x}\in\Lambda} \exp(2\pi i\tau|\vec{x}-\vec{\boldsymbol{u}}|^2)$ in a similar randomized method.

Remark: As long as $\theta(\tau;\Lambda)$ can be estimated at sufficiently many points $\tau_j = \sigma_j + it_j$, it seems that its Fourier coefficient $a(m)$ can be computed directly by approximating the integral.

$$a(m) = \exp(2\pi mt)\int_0^1 d\sigma\hat{\vartheta}(\sigma+it;\Lambda)\exp(-2\pi im\sigma)$$

other than by (3.3), where $\hat{\vartheta}(\sigma+it;\Lambda)$ is the estimation for $\vartheta(\sigma+it;\Lambda)$ and $t>0$. However, a complete analysis (details see section 5) concludes that this direct method has the time complexity at least $exp(2\Pi n^2)$, inferior to the approach we take in (3.5)-(3.8) which is at most $n^{O(n)} = exp(nlogn)$ in time complexity.

3.3 Solving SVP and CVP

For a given (integral) lattice $\Lambda(\boldsymbol{B})$ in $\mathbb{Q}^n$, let $m^* = \lambda_1(\Lambda)^2$ and by $\theta(\tau;\Lambda)$'s Fourier expansion

$$\vartheta(\tau;\Lambda) = \sum_{m=0}^{\infty} a(m)exp(2\pi im\tau) = 1 + a(m^*)exp(2\pi im^*\tau) + \ldots$$

solving $SVP(\Lambda)$ reduces to computing the first non-zero $a(m)$ which can be achieved by repeatedly calling the subroutine $VCP(\Lambda, m)$ described in last section from m=1,2,... up to some appropriate upper-bound, e. g., the upper-bound $cn|det(\Lambda)|^{2/n} = O(nl(\Lambda))$ derived from Mincowski's theorem (section 2.1).

Similarly, let $d^* = dist(\Lambda; \boldsymbol{u})^2$, $\boldsymbol{u}$ be an integral vector such that $2\mathbf{B}\boldsymbol{u}$ in $\mathbb{Z}^n$ and $\boldsymbol{u} \notin \Lambda$, by $\theta(\tau; \Lambda, \boldsymbol{u})$'s Fourier expansion

$$\vartheta(\tau; \Lambda, \vec{u}) = \sum_{m=1}^{\infty} b(m)exp(2\pi im\tau) = b(d^*)exp(2\pi id^*\tau) + \dots$$

solving $CVP(\Lambda; \boldsymbol{u})$ reduces to computing the first non-zero $b(m)$ which can be achieved by repeatedly calling the subroutine n-$VCP(\Lambda, m, \boldsymbol{u})$ described in last section from m=1,2,... up to some appropriate upper-bound, e. g., the upper-bound $O(n^2l(\Lambda))$ derived from the transference theorem ((2.5)-(2.6)).

In summary, our algorithms to solve $SVP(\Lambda)$ and $CVP(\Lambda, \boldsymbol{u})$ in n dimension can be sketched in the following steps.

(1) Call the Gaussian sampler N times independently to get dual lattice vectors $y_1, \dots, y_N$ in Λ^*. N needs to be large enough to make the error sufficiently small.

(2) Estimate the lattice-associated theta function $\theta(\tau; \Lambda)$ (in case of solving SVP) or $\theta(\tau; \Lambda, \boldsymbol{u})$ (CVP and CVPP) by $y_1, \dots, y_N$ and $\boldsymbol{u}$ at sufficiently many points $\tau_j = \sigma_j + it_j$ with all $t_j > 0$. This step is only $poly(n)$ in time and space complexity.

(3) Compute the linear combination coefficients of the theta function on appropriately selected basis in the modular form space. This step is also $poly(n)$ in time and space complexity.

(4) Search the first non-zero Fourier coefficient in the theta function's Fourier expansions.

We note that step (1) can be completely parallelized, i. e., all N Gaussian samplers can work completely in concurrency (each sampler only performs in polynomial time and space complexity [8, 21]). For solving $CVPP(\Lambda, \boldsymbol{u})$, this step can be even performed totally in preprocessing. Since each Fourier coefficient can be computed independently, step (4) can also operate in concurrency of $O(nl(\Lambda))$(for SVP) and $O(n^2l(\Lambda))$(for CVP and CVPP) where $l(\Lambda)$ is the level of lattice Λ. As a result, the whole algorithm can be easily parallelized to be polynomial in time complexity.

So far the framework to solve (integral) lattice optimization problems SVP and CVP has been established. All technical details and computational complexity analysis are elaborated in next sections.

4 Algorithms

In this section all notations are as before, e.g., $\tau = \sigma + it$ denotes the complex variable on the upper-half plane($t > 0$). However, the lattice Λ needs not to be integral in section 4.1.

4.1 Estimating the Values of Lattice-Associated Theta Functions $\vartheta(\tau;\Lambda)$ and $\vartheta(\tau;\Lambda,\vec{u})$

Recall the algorithm framework developed in section 3.2, the goal of estimating theta function's values is to compute the linear coefficients $h_\alpha(\Lambda), \alpha = 1,\ldots,M$ via solving the linear system of equations (3.4). Therefore, it's adequate to do the estimation at a finite number of points $\tau_\alpha, \alpha = 1,\ldots,M$ where $M = \dim M_{n/2}(J(h))$. In particular, these points in H can be selected according to computational efficiency considerations. Our approach to do the estimation is based upon the techniques developed in [8, 21] which basic result is presented in theorem 4.1. For all technical details, see section 4 in [8] and [21].

Theorem 4.1. *[8]:There is a probabilistic polynomial-time algorithm that, given the basis* $\mathbf{B} = (\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n)$ *of* n *-dimensional lattice* Λ*, a parameter* $s > \omega(\log(n)) \max_j |\tilde{b}_j|^2$ *where* $\tilde{b}_1,\ldots,\tilde{b}_n$ *is the Gram-Schmidt orthogonalization of* $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n$*, and a vector* $\boldsymbol{u}$ *in* R^n*, outputs a sample from the distribution which is statistically close to the discrete Gaussian distribution*

$$D_{\Lambda,s,u}(\mathbf{x}) \equiv \exp(-2\pi|x-u|^2/s)/\sum_{x'\in\Lambda}\exp(-2\pi|x'-u|^2/s) \quad for\ \mathbf{x}\ in\ \Lambda$$

□

When $\vec{u} = 0$, $D_{\Lambda,s,u}(x)$ is simply denoted $D_{\Lambda,s}(x)$. Hereafter the sampler in theorem 4.1 is denoted $SampD(\Lambda(\mathbf{B}), s, \boldsymbol{u})$. The original sampler [8]'s efficiency is significantly improved in [21] at a mild price of a larger t. In this paper we neglect such efficiency differences and call $SampD$ as a black-box. In the following we almost always apply $SampD$ to sample on the dual lattice $\Lambda^*(\boldsymbol{b}_1^*,\ldots,\boldsymbol{b}_n^*)$ of the input lattice $\Lambda(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n)$, in this case the original condition $s > \omega(\log n)\max_j |\tilde{b}_j|^2$ becomes $s > \omega(\log n)\max_j |\tilde{b}_j|^{-2}$ because of the relationship between the Gram-Schmidt orthogonalization $\tilde{b}_1,\ldots,\tilde{b}_n$ of the basis $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n$ and $\boldsymbol{d}_n,\ldots,\boldsymbol{d}_1$, that of the dual basis $(\boldsymbol{b}_1^*,\ldots,\boldsymbol{b}_n^*)$:$d_j = \tilde{b}_j/|\tilde{b}_j|^2$. In particular $|\boldsymbol{d}_j| = |\tilde{b}_j|^{-1}$ for $j = 1,\ldots,n$.

Our algorithms to do the estimation are presented in four subroutines, each works with appropriately large $t > 0$.

Estimating $\vartheta(it;\Lambda)$:The randomized algorithm to estimate $\theta(it;\Lambda)$ is presented as follows.

EstimTheta$(it;\Lambda)$:Version ♯1

Input: A pure complex number it with $t > 0$ and a lattice Λ with basis $\mathbf{B} = (\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n)$;

Parameter: A positive integer N;

Output: An estimation of $\theta(it;\Lambda)$;

Operations:

(1) Call $SampD(\Lambda^*(\mathbf{B}^*), 4t, \boldsymbol{0})$ [8] independently N times to compute the delta-function's average, i. e., to obtain

$$\hat{\eta}_N(it; \Lambda^*) = N^{-1} \sum_{j=1}^{N} \delta(y_j)$$

where $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ are dual lattice vectors independently sampled (by *SampD*) under the distribution.

$$D_{\Lambda^*,4t}(\boldsymbol{y}) \equiv \exp(-2\pi|y|^2/4t) / \sum_{y' \in \Lambda^*} \exp(-2\pi|y'|^2/4t) \quad \textit{for } \boldsymbol{y} \textit{ in } \Lambda^*$$

(2) output $\hat{\vartheta}(it; \Lambda) = (2t)^{-n/2} \det(\Lambda^*)/\hat{\eta}_N(it; \Lambda^*)$.

It's direct to see that this subroutine *EstimTheta*can be implemented totally in parallel and if it is implemented on N processors, i.e., N concurrent and independent *SampD*'s, its total time complexity is just a polynomial in its input size. All other estimators in the following have the same characteristic.

As explained in section 3.2, $\hat{\eta}_N(it; \Lambda^*)$ is an estimation for $1/\theta(i/4t; \Lambda^*)$ and according to the Poisson formula $\hat{\vartheta}_N(it; \Lambda)$ is an appropriate estimation for $\theta(it; \Lambda)$. The random estimation subroutine's performance is based-on the following theorem.

Theorem 4.2. *Let* $n = \dim \Lambda(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$*,* c *be any absolute constant satisfying* $c > (2\pi)^{-1/2}$*,* $t > \max(c^2 n/\lambda_1(\Lambda)^2, \omega(\log n) \max_j |\tilde{b}_j|^{-2})$ *where* $\tilde{b}_1, \ldots, \tilde{b}_n$ *are the Gram-Schmidt orthogonalization of* Λ'*s basis* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$*,* $\hat{\eta}_N(it; \Lambda^*)$ *and* $\hat{\vartheta}_N(it; \Lambda)$ *as specified in the subroutine EstimTheta*$(it, \Lambda(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n))$ *with parameter* N*. Then*

$$P[|\hat{\vartheta}_N(it; \Lambda) - \vartheta(it; \Lambda)| < \varepsilon_1] > 1 - \varepsilon_2 \tag{4.1}$$

where A*,*β *are absolute positive constants and* $\varepsilon_1 < 2\varepsilon/((1/2)(2t)^{-n/2} \det(\Lambda^*) - \varepsilon), \varepsilon_2 < A \exp(-\beta N \varepsilon^2) + \exp(-(N/2)(2t)^{-n/2} \det(\Lambda^*))$ *for sufficiently small* $\varepsilon > 0$*.* □

Remark: Intuitively, this theorem guarantees that $\hat{\vartheta}_N(it; \Lambda^*)$ is a "good" estimation for $\vartheta(it; \Lambda)$ with appropriately large t. In practice, one of the "appropriate largeness" condition that $t > c^2 n/\lambda_1(\Lambda)^2$ can be satisfied by replacing the (unkown) $\lambda_1(\Lambda)$ with some of its easy-to-estimate lower-bound, e.g., $\lambda_1(\Lambda) > \min_j |\tilde{b}_j|$ [22] so it is sufficient that $t > c^2 n/\min_j |\tilde{b}_j|^2$. For the integral lattice Λ we can even simply select $t > \max(c^2 n, \omega(\log n) \max_j |\tilde{b}_j|^{-2})$ because $\lambda_1(\Lambda) \geq 1$ in this case.

Theorem 4.2 's proof bases on the following two facts.

Lemma 4.3 (Banaszczk inequality [4]). *Let* $n = \dim \Lambda, c > (2\pi)^{-1/2}$ *an absolute constant and* $\delta_n \equiv (c(2\pi e)^{1/2} \exp(-\pi c^2))^n$*. It is true that*

$$\sum_{\vec{x} \in \Lambda : |\vec{x}| > c\sqrt{n}} \exp(-\pi|\vec{x}|^2) / \sum_{\vec{x} \in \Lambda} \exp(-\pi|\vec{x}|^2) < \delta_n \tag{4.2}$$

□

Lemma 4.4 (Hoeffding-type inequality [23]). *Let $X_1, \ldots, X_N$ be independent random variables and $|x_j| \leq 1$ for any j, then*

$$P[|N^{-1}\sum_{j=1}^{N} X_j - E[X]| > \varepsilon] < A\exp(-\beta N\varepsilon^2) \tag{4.3}$$

□

where both A and β are some absolute positive constants.

Proof. [Proof of Theorem 4.1] We prove theorem in several steps. (1)Under the condition that $t > c^2 n/\lambda_1(\Lambda)^2$,$\Theta(it;\Lambda)$ is both-sides bounded. That's because

$$\begin{aligned}
1 < \vartheta(it;\Lambda) &= \sum_{\vec{x}\in\Lambda} \exp(-2\pi t|\vec{x}|^2) = \sum_{\vec{x}\in\sqrt{2t}\Lambda} \exp(-\pi|\vec{x}|^2) \\
&= 1 + \sum_{\vec{x}\in\sqrt{2t}\Lambda:|\vec{x}|>0} \exp(-\pi|\vec{x}|^2) = 1 + \sum_{\vec{x}\in\Lambda:|\vec{x}|\geq\sqrt{2t}\lambda_1(\Lambda)} \exp(-\pi|\vec{x}|^2) \\
&< 1 + \delta_n \sum_{\vec{x}\in\sqrt{2t}\Lambda} \exp(-\pi|\vec{x}|^2) \quad (\mathit{by}\ t^{1/2}\lambda_1(\Lambda) > cn^{1/2} \mathit{and\ lemma}\ 4.3) \\
&= 1 + \delta_n \sum_{\vec{x}\in\Lambda} \exp(-2\pi t|\vec{x}|^2) = 1 + \delta_n\vartheta(it;\Lambda)
\end{aligned}$$

It follows that

$$1 < \vartheta(it;\Lambda) < 1/(1-\delta_n) \tag{4.4}$$

(2)Under the dual lattice vectors independent sampling with distribution statistically close to $D_{\Lambda^*,4t}(\boldsymbol{y})$ on Λ^* and $t > c^2 n/\lambda_1(\Lambda)^2$, $\hat{\eta}_N(it;\Lambda^*)$ is almost always non-zero when N is sufficiently large:

$$P[\hat{\eta}_N(it;\Lambda^*) \neq 0] > 1 - \exp(-N(1-\delta_n)(2t)^{-n/2}\det(\Lambda^*)) \tag{4.5}$$

This is because $P[\hat{\eta}_N(it;\Lambda^*) = 0] = P[\delta(\boldsymbol{y}_1) = \ldots = \delta(\boldsymbol{y}_N) = 0] = P[\boldsymbol{y}_1 \neq 0 \bigwedge \ldots \bigwedge \boldsymbol{y}_N \neq 0] = P[\boldsymbol{y} \neq 0]^N$so

$$\begin{aligned}
P[\hat{\eta}_N(it;\Lambda^*) = 0] &= (1 - P[\boldsymbol{y} = 0])^N = (1 - 1/\sum_{y\in\Lambda^*} \exp(-2\pi|y|^2/4t))^N \\
&= (1 - (2t)^{-n/2}\det(\Lambda^*)/\sum_{x\in\Lambda} \exp(-2\pi|x|^2 t))^N \quad (\mathit{by\ Poisson\ formula}(\ 2.10a\ )) \\
&\leq (1 - (1-\delta_n)(2t)^{-n/2}\det(\Lambda^*))^N \ (\mathit{by}\ (\ 4.4\ )) \\
&\leq \exp(-N(1-\delta_n)(2t)^{-n/2}\det(\Lambda^*)) \ (\mathit{by}\ 1-x < \exp(-x)\ \mathit{for}\ x > 0)
\end{aligned}$$

(3)Let $\eta(it;\Lambda^*) \equiv 1/\theta(i/4t;\Lambda^*)$ and $|\eta(it;\Lambda^*) - \hat{\eta}_N(it;\Lambda^*)| < \varepsilon$. By (4.4) and the Poisson formula we have $1/(1-\delta_n) > \theta(it;\Lambda) = (2t)^{-n/2}\det(\Lambda^*)\theta(i/4t;\Lambda^*)$, hence $\eta(it;\Lambda^*) \equiv 1/\theta(i/4t;\Lambda^*) > (1-\delta_n)(2t)^{-n/2}\det(\Lambda^*)$, therefore

$$\hat{\eta}_N(it;\Lambda^*) > \eta(it;\Lambda^*) - \varepsilon > (1-\delta_n)(2t)^{-n/2}\det(\Lambda^*) - \varepsilon \tag{4.6}$$

(4)Let the random variable $X \equiv \delta(\boldsymbol{y})$. Notice that $|X| \leq 1$ and the expectation $E[X] = \eta(it;\Lambda^*)$ under the distribution $D_{\Lambda^*,4t}(\boldsymbol{y})$ on the dual lattice Λ^*, it follows from lemma 4.4 that

$$P[|\hat{\eta}_N(it;\Lambda^*) - \eta(it;\Lambda^*)| < \varepsilon] = P[|N^{-1}\sum_{j=1}^{N} X_j - E[X]| < \varepsilon] > 1 - A\exp(-\beta N\varepsilon^2)$$

when $|\hat{\eta}_N(it;\Lambda^*) - \eta(it;\Lambda^*)| < \varepsilon$ we have

$$\begin{aligned}
|\hat{\vartheta}_N(it;\Lambda) - \vartheta(it;\Lambda)| &= (2t)^{-n/2}\det(\Lambda^*)|1/\hat{\eta}_N(it;\Lambda^*) - 1/\eta(it;\Lambda^*)| \\
&= (2t)^{-n/2}\det(\Lambda^*)|\hat{\eta}_N(it;\Lambda^*) - \eta(it;\Lambda^*)|/\hat{\eta}_N(it;\Lambda^*)\eta(it;\Lambda^*) \\
&= \theta(it;\Lambda)|\hat{\eta}_N(it;\Lambda^*) - \eta(it;\Lambda^*)|/\hat{\eta}_N(it;\Lambda^*) \ (by\ Poisson\ formula) \\
&< \varepsilon/\hat{\eta}_N(it;\Lambda^*)(1-\delta_n) \ (by\ (\ 4.4\ )) \\
&< 2\varepsilon/((1-\delta_n)(2t)^{-n/2}\det(\Lambda^*) - \varepsilon)(by(\ 4.6\ ))
\end{aligned}$$

Combined with (4.5) we obtain

$$\begin{aligned}
P[|\hat{\vartheta}_N(it;\Lambda) - \vartheta(it;\Lambda)| < \varepsilon_1] &> P[|\hat{\eta}_N(it;\Lambda^*) - \eta(it;\Lambda^*)| < \varepsilon]P[\hat{\eta}_N(it;\Lambda^*) \neq 0] \\
&> (1 - A\exp(-\beta N\varepsilon^2))(1 - \exp(-N(1-\delta_n)(2t)^{-n/2}\det(\Lambda^*))) \\
&\equiv 1 - \varepsilon_2
\end{aligned}$$

where $\varepsilon_1 = 2\varepsilon/((1-\delta_n)(2t)^{-n/2}\det(\Lambda^*) - \varepsilon) < 2\varepsilon/((1/2)(2t)^{-n/2}\det(\Lambda^*) - \varepsilon)$ and

$$\begin{aligned}
\varepsilon_2 &= A\exp(-\beta N\varepsilon^2) + (1 - A\exp(-\beta N\varepsilon^2))\exp(-N(1-\delta_n)(2t)^{-n/2}\det(\Lambda^*)) \\
&< A\exp(-\beta N\varepsilon^2) + \exp(-(N/2)(2t)^{-n/2}\det(\Lambda^*))
\end{aligned}$$

since $\delta_n < 1/2$ for sufficiently large n. □

Estimating $\vartheta(\sigma + it;\Lambda)$: By Poisson formula

$$\vartheta(\tau;\Lambda) = (i/2\tau)^{n/2}\det\Lambda^*\vartheta(-1/4\tau;\Lambda^*)$$

estimating $\theta(\tau;\Lambda)$ at $\tau = \sigma + it$ reduces to estimating $\theta(-1/4\tau;\Lambda^*)$. Note that

$$\vartheta(-1/4(\sigma+it);\Lambda^*) = \sum_{y\in\Lambda^*}\exp(-2\pi i\sigma|y|^2/4(\sigma^2+t^2))\exp(-2\pi t|y|^2/4(\sigma^2+t^2))$$

$$= \vartheta(it/4(\sigma^2 + t^2); \Lambda^*) \sum_{y \in \Lambda^*} \exp(-2\pi i\sigma |y|^2/4(\sigma^2 + t^2)) D_{\Lambda^*, 4(\sigma^2+t^2)/t}(y)$$

$$= \vartheta(it/4(\sigma^2 + t^2); \Lambda^*) \underset{y \leftarrow D_{\Lambda^*, 4(\sigma^2+t^2)/t}}{\boldsymbol{E}} [\exp(-2\pi i\sigma |y|^2/4(\sigma^2 + t^2))]$$

This implies that $\vartheta(-1/4(\sigma + it); \Lambda^*)$(hence $\vartheta(\sigma + it; \Lambda)$) can be estimated on basis of the estimation of $\theta(it/4(\sigma^2 + t^2), \Lambda^*)$ and the expectation of $\exp(2\pi i\sigma |\boldsymbol{y}|^2/4(\sigma^2 + t^2))$ by sampling dual lattice vectors under the distribution

$$D_{\Lambda^*, 4(\sigma^2+t^2)/t}(y) = \exp(-2\pi t|y|^2/4(\sigma^2 + t^2)) / \sum_{y' \in \Lambda^*} \exp(-2\pi t|y'|^2/4(\sigma^2 + t^2)) \tag{4.7}$$

Note that when t satisfies the condition in theorem 4.2, so does $(\sigma^2 + t^2)/t$. As a result, similar methods as those in *EstimTheta* version ♯1 can be used.

EstimTheta$(\sigma + it, \Lambda)$:Version♯2

Input: A complex number $\sigma + it$ with $t > 0$ and a lattice Λ with basis $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$;

Parameter: A positive integer N;

Output: An estimation of $\theta(\sigma + it; \Lambda)$;

Operations:

(1)Call *SampD*$(\Lambda^*(\mathbf{B}^*), 4(\sigma^2 + t^2)/t, \boldsymbol{0})$ N times independently to compute

$$\hat{\eta}_N^* = N^{-1} \sum_{j=1}^{N} \delta(y_j)$$

$$\hat{E}_N = N^{-1} \sum_{j=1}^{N} \exp(-2\pi i\sigma |y_j|^2/4(\sigma^2 + t^2))$$

where $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ are dual lattice vectors independently sampled under the distribution (4.7)

(2)output $\hat{\vartheta}_N(\sigma + it; \Lambda) = (i/2(\sigma + it))^{n/2} \det(\Lambda^*) \hat{E}_N / \hat{\eta}_N^*$.

The complete estimation for $\theta(\sigma + it; \Lambda)$ is in form of the product of multiple estimated quantities. In this case the error and correctness probability are related by the following general theorem.

Theorem 4.5. *Let* $\hat{\rho_1}$ *and* $\hat{\rho_2}$ *be estimations for* ρ_1 *and* ρ_2 *respectively and*

$$P[|\hat{\rho_1} - \rho_1| < \varepsilon_1] > 1 - \delta_1, P[|\hat{\rho_2} - \rho_2| < \varepsilon_2] > 1 - \delta_2$$

Suppose both ρ_1 *and* ρ_2 *are bounded, i. e.,* $|\rho_1| \le A_1$ *and* $|\rho_2| \le A_2$ *. Then*

$$P[|\hat{\rho_1}\hat{\rho_2} - \rho_1\rho_2| < A_1\varepsilon_2 + A_2\varepsilon_1 + \varepsilon_1\varepsilon_2] > 1 - \delta_1 - \delta_2$$

□

Proof. For any random events X and Y it's true that $P[X\ and\ Y]+P[X\ or\ Y]=P[X]+P[Y]$, hence $P[|\hat{\rho_1}-\rho_1|<\varepsilon_1\ or\ |\hat{\rho_2}-\rho_2|<\varepsilon_2]+P[|\hat{\rho_1}-\rho_1|<\varepsilon_1\ and\ |\hat{\rho_2}-\rho_2|<\varepsilon_2]=P[|\hat{\rho_1}-\rho_1|<\varepsilon_1]+P[|\hat{\rho_2}-\rho_2|<\varepsilon_2]>1-\delta_1+1-\delta_2=2-\delta_1-\delta_2$. It follows that $P[|\hat{\rho_1}-\rho_1|<\varepsilon_1\ and\ |\hat{\rho_2}-\rho_2|<\varepsilon_2]>1-\delta_1-\delta_2$. Furthermore, when $|\hat{\rho_1}-\rho_1|<\varepsilon_1$ and $|\hat{\rho_2}-\rho_2|<\varepsilon_2$ we have $|\hat{\rho_1}\hat{\rho_2}-\rho_1\rho_2|<|\hat{\rho_1}||\hat{\rho_2}-\rho_2|+|\rho_2||\hat{\rho_1}-\rho_1|$ and $|\hat{\rho_1}|<|\rho_1|+\varepsilon_1<A_1+\varepsilon_1$, then $|\hat{\rho_1}\hat{\rho_2}-\rho_1\rho_2|<A_1\varepsilon_2+A_2\varepsilon_1+\varepsilon_1\varepsilon_2$, hence

$$1-\delta_1-\delta_2<P[|\hat{\rho_1}-\rho_1|<\varepsilon_1\ and\ |\hat{\rho_2}-\rho_2|<\varepsilon_2]\leq P[|\hat{\rho_1}\hat{\rho_2}-\rho_1\rho_2|<A_1\varepsilon_2+A_2\varepsilon_1+\varepsilon_1\varepsilon_2]$$

□

Combining theorem 4.2 and 4.5, a theorem about the relationship between the estimation error and correctness probability similar as that established in theorem 4.2 can be easily derived for *EstimTheta* Version♯2, having exactly the same condition for t and the same upper-bounds for ε_1 and ε_2 with only differences in some absolute constants. We simply omit all these redundant details, leaving theorem 4.2 as an universal conclusion for all these subroutines' performances.

Estimating $\vartheta(it;\Lambda,\vec{u})$:By definition in section 2.3 and Poisson formula, we have

$$\begin{aligned}\vartheta(it;\Lambda,\vec{u})&=\sum_{\vec{x}\in\Lambda}\exp(-2\pi t|\vec{x}-\vec{u}|^2)\\&=\sum_{\vec{x}'\in\Lambda}\exp(-2\pi t|\vec{x}'|^2)\cdot\sum_{\vec{x}\in\Lambda}\exp(-2\pi t|\vec{x}-\vec{u}|^2)/\sum_{\vec{x}\in\Lambda}\exp(-2\pi t|\vec{x}|^2)\\&=\vartheta(it;\Lambda)\cdot\sum_{\vec{y}\in\Lambda^*}(\exp(-2\pi|\vec{y}|^2/4t)\exp(2\pi i<\vec{u},\vec{y}>)/\sum_{\vec{y}\in\Lambda^*}\exp(-2\pi|\vec{y}|^2/4t))\\&=\vartheta(it;\Lambda)\cdot\underset{y\leftarrow D_{\Lambda^*,4t}}{\boldsymbol{E}}[\exp(2\pi i<\vec{u},\vec{y}>)]\end{aligned}\tag{4.8}$$

where the expectation is over the discrete Gaussian distribution $D_{\Lambda^*,4t}(\boldsymbol{y})$ on the dual lattice Λ^*. Notice that this derives a product type of estimation as for $\theta(\sigma+it;\Lambda)$ and a theorem like theorem 4.2 can be established in a similar way. In proving such theorem the only difference is to use a variant of lemma 4.3 (Banaszczk inequality) which is presented in the following:

Lemma 4.6. *[4] : Let $n=\dim\Lambda$, $c>(2\pi)^{-1/2}$ and δ_n as in lemma 4.3. For any n-dimensional vector u it is true that*

$$\sum_{\vec{x}\in\Lambda:|\vec{x}-\vec{u}|>c\sqrt{n}}\exp(-\pi|\vec{x}-\vec{u}|^2)/\sum_{\vec{x}\in\Lambda}\exp(-\pi|\vec{x}|^2)<2\delta_n.$$

□

.We now present the subroutine to do the random estimation.

EstimTheta$(it,\Lambda,\boldsymbol{u})$:Version♯3

Input: A pure complex number it with $t>0$, a lattice Λ with basis $\mathbf{B}$ and a vector $\boldsymbol{u}$ in $\mathbb{R}^n$;

Parameter: A positive integer N;

Output: An estimation of $\theta(it;\Lambda,\boldsymbol{u})$;

Operations:

(1)Call *SampD*$(\Lambda^*(\mathbf{B}^*),4t,\boldsymbol{0})$ N times independently to compute

$$\hat{\eta}_N^* = N^{-1}\sum_{j=1}^{N}\delta(y_j)$$

$$\hat{E}_N = N^{-1}\sum_{j=1}^{N}\exp(2\pi i < u, y_j >)$$

where $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N$ are dual lattice vectors independently sampled under the distribution $D_{\Lambda^*,4t}$.

(2)output $\hat{\vartheta}_N(it;\Lambda,u) = (2t)^{-n/2}\det(\Lambda^*)\hat{E}_N/\hat{\eta}_N^*$

Estimating $\vartheta(\sigma+it;\Lambda,\vec{u})$: By definition in section 2.3 and Poisson formula, we have

$$\vartheta(\sigma+it;\Lambda,\vec{u}) = (i/2(\sigma+it))^{n/2}\det(\Lambda^*)\sum_{y\in\Lambda^*}\exp(-2\pi i|y|^2/4(\sigma+it))\exp(2\pi i<\vec{u},\vec{y}>)$$

$$=(i/2(\sigma+it))^{n/2}\det(\Lambda^*)\sum_{y\in\Lambda^*}\exp(-2\pi i|y|^2(\sigma-it)/4(\sigma^2+t^2))\exp(2\pi i<\vec{u},\vec{y}>)$$

$$=(i/2(\sigma+it))^{n/2}\det(\Lambda^*)\vartheta(it/4(\sigma^2+t^2);\Lambda^*)\cdot \mathop{\mathbf{E}}_{y\leftarrow D_{\Lambda^*,4(\sigma^2+t^2)/t}}[\exp(\frac{-2\pi i\sigma|y|^2}{4(\sigma^2+t^2)}+2\pi i<\vec{u},\vec{y}>)]$$

so $\theta(\sigma+it;\Lambda,\boldsymbol{u})$ can be estimated by the following subroutine.

EstimTheta$(\sigma+it,\Lambda,\boldsymbol{u})$: Version♯4

Input: A complex number $\sigma+it$ with $t>0$, a lattice Λ with basis $\mathbf{B}=(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n)$ and a vector $\boldsymbol{u}$ in R^n.

Parameter: A positive integer N;

Output: An estimation of $\theta(\sigma+it;\Lambda,\vec{u})$;

Operations:

(1)Call *SampD*$(\Lambda^*(\mathbf{B}^*),4(\sigma^2+t^2)/t,\boldsymbol{0})$ N times independently to compute

$$\hat{\eta}_N^* = N^{-1}\sum_{j=1}^{N}\delta(y_j)$$

$$\hat{E}_N = N^{-1}\sum_{j=1}^{N}\exp(-2\pi i\sigma|y_j|^2/4(\sigma^2+t^2)+2\pi i<y_j,u>)$$

where $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N$ are dual lattice vectors independently sampled under the distribution (4.7)

(2)output $\hat{\vartheta}_N(\sigma+it;\Lambda) = (i/2(\sigma+it))^{n/2}\det(\Lambda^*)\hat{E}_N/\hat{\eta}_N^*$.

Remarks: Obviously the Gaussian sampler *SampD* can be called completely in concurrency in all the four *EstimTheta* subroutines. As a result, all the four subroutines can work in just polynomial time complexity with N-concurrency. In addition, by calling *SampD* offline, we obtain an algorithm to solve CVPP in polynomial time and N space complexity. In section 5 we'll see that $N = n^{O(n)}$.

4.2 Computing Linear Combination Coefficients $h_\alpha(\Lambda)$ and $h_\alpha(\Lambda; \boldsymbol{u})$

In solving SVP, let $\{\varphi_\alpha(\tau) : \alpha = 1, \ldots, M\}$ be the basis of space $M_{n/2}(J(h))$, $M = dimM_{n/2}(J(h))$, $\hat{\vartheta}(\tau_i; \Lambda)$be theta-function value estimations at points $\tau_1, \ldots \tau_M$ on the upper-half plane H. As long as $det(\varphi_\alpha(\tau_\beta))_{1\le\alpha,\beta\le M} \neq 0$, by solving the system of linear equations

$$\hat{\vartheta}(\tau_\alpha; \Lambda) = \sum_{\beta=1}^{M} \hat{h}_\beta(\Lambda)\varphi_\beta(\tau_\alpha) \qquad \alpha = 1, \ldots, M \tag{4.9}$$

all $h_\alpha(\Lambda)$'s estimations $\hat{h}_\alpha(\Lambda)$ can be obtained.

In solving CVP the situation is similar with the only difference that we need to solve the linear system of equations

$$\hat{\vartheta}(\tau_\alpha; \Lambda, u) = \sum_{\beta=1}^{M} \hat{h}_\beta(\Lambda, u)\varphi_\beta(\tau_\alpha) \qquad \alpha = 1, \ldots, M \tag{4.10}$$

. For simplicity hereafter we only use the notation h_α instead of $h_\alpha(\Lambda)$ and $h_\alpha(\Lambda; u)$.

In essence, what is really needed in our algorithms is not any specific basis of space $M_{n/2}(J(h))$, but just a set of points $\tau_1 \ldots \tau_M$ such that $det(\varphi_\alpha(\tau_\beta))_{1\le\alpha,\beta\le M} \neq 0$, a set of function values $\{\varphi_\alpha(\tau_\beta))\}_{1\le\alpha,\beta\le M}$ and a set of Fourier coefficients of these basis (see section 4.3). Moreover, notice the fact that $M_k(J(h))$ is a subspace in $M_k(\Gamma(4h))$, in practice we can even use the basis of the much better understood space $M_k(\Gamma(4h))$ with only moderate prices in time and space complexity (see (2.15) and complexity analysis in next section). Given any positive integer N, the space $M_k(\Gamma(N))$ has an orthogonal decomposition (with respect to the Petersson inner product) [6, 13, 24]where $S_k(\Gamma(N))$ is the so called cusp form subspace.

$$M_k(\Gamma(N)) = S_k(\Gamma(N)) \oplus E_k(\Gamma(N))$$

For instance, the basis of subspace $E_k(\Gamma(N))$ can be selected to be the Eisenstein functions

$$G_k^{(u,v)}(\tau) = \sum_{(c,d):(c,d)=(u,v)mod\ N} 1/(c\tau + d)^k \quad = \sum_{m\ge0} g_k^{(u,v)}(m)exp(2\pi im\tau/N)$$

for all integer-pairs (u, v) of order N in $Z_N \times Z_N$. It's well known that these basis have Fourier coefficients [6, 24]

$$g_k^{(u,v)}(m) = ((-2\pi i)^k/(k-1)!) \sum_{j=-m,\ldots,+m, j|m, m/j=u\ mod\ N, j\neq0} sgn(j)j^{k-1}exp(2\pi ivj/N) \qquad m \ge 1$$

where $sgn(j)$=1 when $j > 0$, -1 when $j < 0$(we neglect $g_k^{(u,v)}(0)$ which is not needed in our algorithm). It's clear from the formulas that the m-th Fourier coefficient can be computed in at most $poly(m, logk)$ time complexity. In the proceeding applications to solve lattice problems, both m and k are O(n) where n is the lattice's dimension. Similar situation holds for $S_k(\Gamma(N))$.

Since this paper is only concentrated on the algorithm's logic and complexity analysis, we defer to discuss all numerical computation related details in a separate paper, only pointing out that for integer or half-integer k there exit efficient algorithms (polynomial in log$k$ and m) to output the m-th Fourier coefficient of the basis in space $M_k(\Gamma(N))$.

To complete the computation, we need to confirm that the condition $det(\varphi_\alpha(\tau_\beta))_{1\leq\alpha,\beta\leq M} \neq 0$ can be really satisfied. The following lemma guarantees the existence of such points $\tau_1, \dots, \tau_M$ on the upper-half plane H.

Lemma 4.7. *Let m be a positive integer, D be a domain in the upper-half plane H, $\varphi_1(\tau), \dots, \varphi_m(\tau)$ be complex-valued functions holomorphic in D. If $\varphi_1(\tau), \dots, \varphi_m(\tau)$ are linearly independent over the complex field, then there exist m points $\tau_1, \dots, \tau_m$ in D such that $det(\varphi_\alpha(\tau_\beta))_{1\leq\alpha,\beta\leq m} \neq 0$.* □

Proof. (by induction on m) For $m = 1$ the result is trivial. Now suppose the lemma is true for m. For $m+1$ complex linearly independent functions $\varphi_0(\tau), \varphi_1(\tau), \dots, \varphi_m(\tau)$ holomorphic in D, by induction there exist points $\tau_1, \dots, \tau_m$ in D such that $det(\varphi_\alpha(\tau_\beta))_{1\leq\alpha,\beta\leq m} \neq 0$. Because $\varphi_0(\tau)$ is holomorphic and not identically zero in D, we can always assume (by slightly changing some τ_β's if needed) that at least one of the $\varphi_0(\tau_\beta)$'s is non-zero. As a result, there exist (obtained by solving the following linear system of equations) complex values $a_1, \dots, a_m$ such that

$$\varphi_0(\tau_\beta) = a_1\varphi_1(\tau_\beta) + \ldots + a_m\varphi_m(\tau_\beta) \qquad for\ all\ \beta = 1, \dots, m \tag{4.11}$$

and at least one of the a_β's is non-zero. By complex linear independency among the functions φ_0, $\varphi_1, \dots, \varphi_m$, $\varphi_0 \neq a_1\varphi_1 + \ldots + a_m\varphi_m$ so there exists a point τ_0 in D such that

$$\varphi_0(\tau_0) \neq a_1\varphi_1(\tau_0) + \ldots + a_m\varphi_m(\tau_0) \tag{4.12}$$

in consequence, $det(\varphi_\alpha(\tau_\beta))_{0\leq\alpha,\beta\leq m} \neq 0$ (otherwise the following matrix

$$\begin{bmatrix} \varphi_0(\tau_0) & \varphi_1(\tau_0) & \dots & \varphi_m(\tau_0) \\ \varphi_0(\tau_1) & \varphi_1(\tau_1) & \dots & \varphi_m(\tau_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0(\tau_m) & \varphi_1(\tau_m) & \dots & \varphi_m(\tau_m) \end{bmatrix}$$

is singular so there exist $a_1, \dots, a_m$ such that

$$\varphi_0(\tau_\beta) = a_1\varphi_1(\tau_\beta) + \ldots + a_m\varphi_m(\tau_\beta) \qquad for\ all\ \beta = 0, 1, \dots, m$$

But due to $det(\varphi_\alpha(\tau_\beta))_{1\leq\alpha,\beta\leq m} \neq 0$, these $a_1,\dots,a_m$'s are exactly those in (4.11), a contradiction to (4.12) □

Remark: It's easy to derive an efficient algorithm from the lemma's proof to output a sequence of points $\tau_1,\dots,\tau_m$ in D such that $det(\varphi_\alpha(\tau_\beta))_{1\leq\alpha,\beta\leq m} \neq 0$, given the functions $\varphi_1,\dots,\varphi_m$ and domain D satisfying the conditions specified in lemma 4.7.

4.3 Complete Algorithms

Now we integrate all the components to construct the complete algorithms to solve the optimization lattice problems. As indicated before, these algorithms find not only the classical solutions to the optimization SVP and CVP but also the number of lattice vectors which reach the minimums.

To make the algorithm's structure clear, we introduce an oracle to help collect necessary information.

Oracle-M(h, k, m^*, t_0)

Input: A positive integer h, a positive integer or half-integer k and two positive real numbers m^*, t_0.

Output:

(1)A collection of Fourier coefficients $\{a_\alpha(m) : \alpha = 1,\dots,M, m = 1,\dots,m^*\}$ where $M = dim_C M_k(J(h))$ with respect to some basis $\{\varphi_\alpha(\tau) : \alpha = 1,\dots,M\}$ of the space $M_k(J(h))$. $a_\alpha(m)$ denotes the m-th Fourier coefficient of $\varphi_\alpha(\tau)$.

(2)A collection of points $\tau_1,\dots,\tau_M$ on the upper-half complex plane H such that

$$Im\tau_\alpha > t_0 \ for\ each\ 1 \leq \alpha \leq M$$

and

$$det(\varphi_\alpha(\tau_\beta))_{1\leq\alpha,\beta\leq M} \neq 0$$

(3)A collection of values $\{\Phi_{\alpha\beta} : 1 \leq \alpha,\beta \leq M,\ the\ matrix\ (\Phi_{\alpha\beta}) = (\varphi_\alpha(\tau_\beta))^{-1}\}$

Remark: The oracle-M can be implemented based on and only on the knowledge about the congruence subgroup $J(h)$ or, as explained in section 4.2, the group $\Gamma(4h)$. As explained in section 4.2, any basis of space $M_k(J(h))$ or even $M_k(\Gamma(4h))$ is sufficient for our algorithmic goals so we can always select the most appropriate and efficient basis in practice. In summary, each Fourier coefficient $a_\alpha(m)$ can be computed with time complexity polynomial in k and m, and the points $\tau_1,\dots,\tau_M$ can be also determined efficiently.

Now we present our algorithms to solve the optimization lattice problem SVP and CVP.

Algorithm to Solve Optimization SVP:

Input: an integral lattice $\Lambda(\mathbf{B}) = Z\mathbf{b}_1 + \dots + Z\mathbf{b}_n$ in $\mathbb{Q}^n$.

Parameters: Positive absolute constants $c \leq 1$ *and* $c_0 > (2\pi)^{-1/2}$.

Output: $\lambda_1(\Lambda) \equiv min\{|\boldsymbol{x}| : \boldsymbol{x} \ in \ \Lambda \ and \ non\text{-}zero\}$ and $a^*(\Lambda) = |\{\boldsymbol{x} \ in \ \Lambda : \ |\boldsymbol{x}| = \lambda_1(\Lambda)\}|$.

Operations:

(1)Compute $h = l(\Lambda)$, the level of lattice Λ, as stated in the paragraph following definition 2.1.

(2)Set $m^* = cn|det(\Lambda)|^{2/n}$ and $t_0 > max(c_0^2 n, \omega(logn)max_j|\tilde{b_j}|^{-2})$. Call *Oracle-M*$(h, \ n/2, \ m^*, \ t_0)$ to obtain:

A collection of Fourier coefficients $\{a_\alpha(m) : \ \alpha = 1, \ldots, M, \ \ m = 1, \ldots, m^*\}$ where $M = dim_C M_{n/2}(J(h))$ with respect to some basis $\{\varphi_\alpha(\tau) : \ \ \alpha = 1, \ldots, M\}$ of space $M_{n/2}(J(h))$ and $a_\alpha(m)$ denotes the m-th Fourier coefficient of $\varphi_\alpha(\tau)$;

A collection of points $\{\tau_\beta = \sigma_\beta + it_\beta : \ \ \beta = 1, \ldots, M\}$ such that $t_\beta > t_0$ for each $1 \leq \beta \leq M$ and $det(\varphi_\alpha(\tau_\beta))_{1 \leq \alpha, \beta \leq M} \neq 0$;

A collection of values $\{\Phi_{\alpha\beta} : 1 \leq \alpha, \beta \leq M, \ the \ matrix \ (\Phi_{\alpha\beta}) = (\varphi_\alpha(\tau_\beta))^{-1}\}$.

(3)For each $\beta = 1, \ldots, M$ call *EstimTheta* version#2 with input $(\tau_\beta, \Lambda(\mathbf{B}))$ and parameter N to obtain $\hat{\vartheta}(\tau_\beta; \Lambda)$($N$ depends on $n = dim\Lambda(\mathbf{B})$ and its value will be determined according to complexity analysis in next section).

(4)compute $\hat{h}_\beta = \sum_{\alpha=1}^{M} \Phi_{\alpha\beta}\hat{\vartheta}(\tau_\alpha; \Lambda)$ for each $\beta = 1, \ldots, M$.

(5)For each $m = 1, 2, \ldots, m^*$ do:

$$Compute \ \hat{a}(m) = \sum_{\beta=1}^{M} \hat{h}_\beta a_\beta(m); \ \ if \ \ \hat{a}(m) > 1/2 \ then \ break;$$

(6)Output $(m^{1/2}, \ [\hat{a}(m)])$ where $[x]$ denotes the integer nearest to x.

Algorithm to Solve Optimization CVP:

Input: an integral lattice $\Lambda(\mathbf{B}) = Z\boldsymbol{b}_1 + \ldots + Z\boldsymbol{b}_n$ in $\mathbb{Q}^n$, a vector $\boldsymbol{u}$ in $\mathbb{Z}^n \backslash \Lambda(\mathbf{B})$ such that $2\mathbf{B}\boldsymbol{u}$ in $\mathbb{Z}^n$.

Parameters: Positive absolute constants $d \leq 1/2 \ and \ c_0 > (2\pi)^{-1/2}$.

Output: $dist(\Lambda; u) \equiv \min\{|x - u| : x \in \Lambda\}$ and $b^*(\Lambda) = |\{x \in \Lambda : |x - u| = dist(\Lambda; u)\}|$

Operations:

(1) and (2): The same as steps (1) and (2) in the algorithm to solve the optimization SVP, except that $m^* = dn^2 h$. All notations are inherited.

(3)For each $\beta = 1, \ldots, M$ call *EstimTheta* version#4 with input $(\tau_\beta, \Lambda(\mathbf{B}), \boldsymbol{u})$ and parameter N to obtain $\hat{\vartheta}(\tau_\beta; \Lambda, u)$ (N depends on $n = dim\Lambda(\mathbf{B})$ and its value will be determined according to complexity analysis in next section).

(4)Compute $\hat{h}_\beta = \sum_{\alpha=1}^{M} \Phi_{\alpha\beta}\hat{\vartheta}(\tau_\alpha; \Lambda, u) \ for \ each \ \beta = 1, \ldots, M.$

(5)For each $m = 1, 2, \ldots, m^*$ *do* :

$$Compute\ \hat{b}(m) = \sum_{\beta=1}^{M} \hat{h}_\beta a_\beta(m);\ if\ \hat{b}(m) > 1/2\ then\ break;$$

(6)Output $(m^{1/2},\ [\hat{b}(m)])$ where $[x]$ denotes the integer nearest to x.

5 Complexity Analysis

Before delve into the algorithm's complexity, we need a fact about the modular form's Fourier coefficient's asymptotic increasing degree.

Lemma 5.1. *[6, 13] Let* Γ *be a congruence subgroup in* $SL_2(\mathbb{Z})$ *and* $\varphi(\tau)$ *be a modular form in* $M_k(\Gamma)$ *with Fourier coefficients* $a(m)$*,* $m \geq 1$*, then*

$$|a(m)| \leq Am^k\ for\ any\ m \geq 1.$$

where A *is a constant irrelevant with* k*. For cusp form* $\varphi(\tau)$ *in* $S_k(\Gamma)$ *with Fourier coefficients* $a(m)$*,* $m \geq 1$*, the inequality is*

$$|a(m)| \leq Am^{k/2}\ for\ any\ m \geq 1.$$

□

Let $h = l(\Lambda)$, the level of lattice Λ. Now consider the algorithm for SVP. According to step (5), for any $1 \leq m \leq m^*$ we have

$$\begin{aligned}
&|the\ error\ of\ \hat{a}(m)|\\
\leq& M \max_{1\leq\beta\leq M} |\hat{h}_\beta| \max_{1\leq\beta\leq M} |a_\beta(m)|\\
\leq& M^2 \max_{1\leq\alpha\leq M} |the\ estimation\ error\ of\ \hat{\vartheta}(\tau_\alpha;\Lambda)| \max_{1\leq\alpha,\beta\leq M} |\Phi_{\alpha_\beta}| Am^{n/2}\\
\leq& constant \cdot M^2 m^{n/2} \max_{1\leq\alpha\leq M} |the\ estimation\ error\ of\ \hat{\vartheta}(\tau_\alpha;\Lambda)|\\
\leq& constant \cdot (nh^3)^2 (nh)^{n/2} \max_{1\leq\alpha\leq M} |the\ estimation\ error\ of\ \hat{\vartheta}(\tau_\alpha;\Lambda)|\\
\leq& constant \cdot n^{2+n/2} h^{n/2+6} \max_{1\leq\alpha\leq M} |the\ estimation\ error\ of\ \hat{\vartheta}(\tau_\alpha;\Lambda)|
\end{aligned}$$

The second inequality is derived by step(4) and the upper-bound for $|a_\beta(m)|$. The fourth inequality is from (2.11), i. e., $m \leq m^* = O(n|det(\Lambda)|^{2/n})$ and $det(\Lambda)^2|h^n)$.

Notice that the exact value of each $a(m)$, the Fourier coefficient of the theta function $\vartheta(\tau;\Lambda)$, is a non-negative integer so it is sufficient to get the correct solution as long as $|the\ error\ of\ \hat{a}(m)| < 1/2$. As a result, we need |the estimation error of $\hat{\vartheta}(\tau_\alpha;\Lambda)|$=$O(n^{-2-n/2}h^{-n/2-6})$ for all τ_α's in step(3). By theorem 4.2 and let $\varepsilon_1 = n^{-2-n/2}h^{-n/2-6}$, a direct calculation shows that this requires the number of (dual) lattice vector samples N, i. e., the times for the Gaussian sampler to be independently called, should be $N = O(n^{4+2n}h^{n+12}|det(\Lambda)|^2 \log(1/\varepsilon_2)) = O(n^{4+2n}h^{2n+12} \log(1/\varepsilon_2))$ to make $P[|\hat{\vartheta}_N(\sigma+it;\Lambda) - \vartheta(\sigma+it;\Lambda)| < \varepsilon_1] > 1-\varepsilon_2$, equivalently, to make $P[|\hat{a}(m) - a(m)| < 1/2] > 1-\varepsilon_2$. In summary, we have proven:

Theorem 5.2. *For the algorithm in section 4.3 to solve the optimization SVP for integral lattice Λ, $n = dim(\Lambda)$, $h = l(\Lambda)$ and $1 > \varepsilon > 0$, it holds that the probability of the algorithm terminating with the correct solution $(\lambda_1(\Lambda), a^*(\Lambda))$ is at least $1-\varepsilon$, if the number of lattice vector samples $N = O(n^{4+2n}h^{2n+12} \log(1/\varepsilon))$.*

For the algorithm to solve the optimization CVP, the result is the same. □

Now we can estimate the algorithm's time and space complexity. Let $T(i)$ and $S(i)$ denote the time and space complexity in step i respectively, $n = dim(\Lambda)$, $h = l(\Lambda)$, $S = \max_{1\leq i\leq n}$ the bit size of each entry in $\boldsymbol{b}_i$, poly denote some (multivariate) polynomial. It's easy to verify that:

both $T(1)$ and $S(1)$ are $poly(n, S)$ according to the analysis after definition 2.1.

$T(2) = \sum_{1\leq m\leq m^*} poly(m, S) = poly(n, S, h)$ according to $m^* = O(n|det(\Lambda)|^{2/n})$ and $det(\Lambda)^2|h^n$, the analysis in section 4.2 and remarks on oracle-M. S(2)=the space to store the outputs from the *oracle-*$M = O(m^*M + M^2)poly(S) = O(n^2h^6)poly(S) = poly(n, S, h)$ according to remarks on lemma 2.3, i. e., $M = O(nh^3)$.

$T(3) = Npoly(n, S)$ where $N = O(n^{4+2n}h^{2n+12}log(1/\varepsilon))$ as stated in theorem 5.2 and $S(3) = poly(n, S)$.

$T(4) = Mpoly(S) = poly(n, S, h)$ and $S(4) = poly(n, S)$.

$T(5) = Mm^*poly(S) = poly(n, S, h)$ and $S(5) = poly(nS)$.

In summary we obtain the central result in this paper:

Theorem 5.3. *There exists a randomized algorithm to solve the optimization SVP with correctness probability at least $1-\varepsilon$ for integral lattice instance $\Lambda(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ of dimension n and level $l(\Lambda)$, in time complexity of $n^{4+2n}l(\Lambda)^{2n+12}poly(n, S)log(1/\varepsilon) + poly(n, S, l(\Lambda))$ and space complexity of $poly(n, S, l(\Lambda))$ where $S = \max_{1\leq i\leq n}$bit-size of each entry in $\boldsymbol{b}_i$.*

For solving the optimization CVP, the same result holds. □

Remarks: It is not really necessary for the algorithms in section 4.3 to store all the outputs from the oracle in a batch. Instead they can get these outputs in sequence when needed. As a result, the space complexity for both algorithms can be actually reduced to only $poly(n, S)$, independent of the level $l(\Lambda)$, while the time complexity's asymptotic bounds are unchanged.

For high dimensional but low level lattices, these algorithms have relatively good performance because, due to the algorithm's logic, in this case the only exponential factor in time complexity is contributed by step (3), the times N to independently sample lattice vectors. All the rest operations in the algorithms only contribute an additive $poly(n, S)$ to time complexity. For example, for the lattice family of constant-bounded levels but arbitrarily high dimensions, the time complexity is $n^{4+2n+\delta}\log(1/\varepsilon)$ with $\delta > 0$ for the correctness probability to be higher than $1-\varepsilon$ (such family of integral lattices always exist and frequently occur in practice, for example, the self-dual lattices of arbitrary dimension have their levels only 1. Lots of such important examples with interesting applications can be found,e. g., in [5]). We conclude the complexity results in this useful case in:

Corollary 5.4. *There exists a randomized algorithm to solve the optimization SVP such that on input the integral lattice family* $\{\Lambda_n\}_{n\geq 1}$ *of* $n = dim\Lambda_n$ *and level* $l(\Lambda) = O(n^{\alpha})$, $\alpha > 0$, *the algorithm outputs the correct solution with probability at least* $1-\varepsilon$ *in time complexity of* $n^{(\alpha+1)2n}poly(n, S)\log(1/\varepsilon)$ *and space complexity of* $poly(n, S)$ *where* $S = \max_{1\leq i\leq n}$*bit-size of each entry in base* $\boldsymbol{b}_i$.

For solving the optimization CVP, the same result holds. □

Another characteristic of our algorithm is its ability to be parallelized to be polynomial in time complexity. As noticed in the end of section 4.1 every *EstimTheta* subroutine can be easily implemented in parallel, i. e., to sample the dual lattice by calling N independent *SampD*'s in concurrency, and in this concurrent version not only each *EstimTheta* but also the whole algorithm to solve SVP or CVP becomes polynomial in time complexity. Such characteristic will be valuable in practice. Notice that step (5) in the algorithms in section 4.3 can be also parallelized on m^* processors. In summary, we have:

Corollary 5.5. *(1)There exists a randomized algorithm to solve the optimization SVP such that on input the integral lattice family* $\{\Lambda_n\}_{n\geq 1}$ *of* $n = dim\Lambda_n$ *and level* $l(\Lambda) = O(n^{\alpha})$, $\alpha > 0$, *it outputs the correct solution with probability at least* $1-\varepsilon$ *in time complexity of* $poly(n, S)$ *on at most* $n^{(\alpha+1)2n}poly(n, S)log(1/\varepsilon)$ *processors, where* $S = \max_{1\leq i\leq n}$*bit-size of each entry in base* $\boldsymbol{b}_i$.

(2)For solving the optimization CVP, the same result holds.

(3)For solving CVPP, there exists a randomized algorithm such that on input $\{(\Lambda_n, \boldsymbol{u})\}_{n\geq 1}$ *where the integral lattice* Λ_n*'s are as in (1), it outputs the correct solution with probability at least* $1-\varepsilon$ *in time complexity of* $poly(n, S)$ *and space complexity of* $n^{(\alpha+1)2n}poly(n, S)log(1/\varepsilon)$, *where* $S = \max_{1\leq i\leq n}$*bit-size of each entry in base* $\boldsymbol{b}_i$. □

Before ending the section we make a brief analysis on why the Fourier coefficient $a(m)$ is not computed directly by approximating the following integral

$$a(m) = \exp(2\pi mt)\int_0^1 d\sigma\hat{\vartheta}(\sigma + it; \Lambda)exp(-2\pi im\sigma) \quad m = 1, 2, \ldots, m^*$$

where $\hat{\vartheta}(\sigma + it; \Lambda)$ is the estimation for $\theta(\sigma + it; \Lambda)$ and $t > 0$. The reason is that, for the error of all such computed $a(m)$'s to be within 1/2, the estimation error of $\hat{\vartheta}(\sigma + it; \Lambda)$ needs to be within $O(\exp(-2\pi m^* t) = O(\exp(-2\pi l(\Lambda)n^2))$ implying that the number of lattice vector samples in step (3), N, needs to be $N = O(\exp(4\pi l(\Lambda)n^2)log(1/\varepsilon))$ to make the correctness probability at least $1-\varepsilon$, significantly inferior to the performance concluded in theorem 5.3.

6 Extensions and Future Works

In this paper we constructed lattice problem algorithms by exploiting the algebraic properties of (integral) lattice associated theta functions. To solve SVP, e. g., such function is:

$$\vartheta(\tau; \Lambda) \equiv \sum_{\vec{x} \in \Lambda} \exp(2\pi i \tau |x|^2)$$

where $|\boldsymbol{x}|$ denotes the vector $\boldsymbol{x}$'s ℓ^2-norm and $\tau = \sigma + it$ is a complex variable on the upper-half complex plane. On one hand this approach is specific to ℓ^2-norm, on the other hand, it can be extended to solve more generalized types of lattice problems or problems of lattices with more special algebraic structures. In this section we give a brief description on some of these extensions.

6.1 Algorithms for Generalized SVP

SVP for (integral) lattices is a special case of the following quadratic minimization problem

$$\min\{\boldsymbol{z}^T \mathbf{A} \boldsymbol{z} : \boldsymbol{z} \textit{ in } \mathbb{Z}^n \textit{ and nonzero}\} \tag{6.1}$$

where $\mathbf{A}$ is a given positive-definite and symmetric integral matrix. When $\mathbf{A}$ is the Grahm matrix of some lattice $\Lambda(\mathbf{B})$, e. g., $\mathbf{A} = \mathbf{B}^T\mathbf{B}$, (6.1) becomes SVP for lattice Λ. But in general cases none of the existed algorithms to solve lattice SVP can be extended to solve (6.1) because all these solvers depend on lattice specific geometric properties the general integral quadratic form doesn't have. However, the techniques in our approach still work in case of (6.1). The theta function associated with the integral matrix $\mathbf{A}$ is

$$\vartheta(\tau; A) \equiv \sum_{\vec{z} \in \mathbb{Z}^n} \exp(2\pi i \tau z^T A z) \tag{6.2}$$

where τ is on the upper-half complex plane. Its first non-zero Fourier coefficient $a(m)$ among non-constant items, i. e., the m^* such that $a(m) = 0$ for all $1 \le m \le m^* - 1$ and $a(m^*) \ne 0$, implies that

$$m^* = \min\{\boldsymbol{z}^T \mathbf{A} \boldsymbol{z} : \boldsymbol{z} \textit{ in } \mathbb{Z}^n \textit{ and nonzero}\}$$

and $a(m^*)$ is exactly the number of integral solutions of the quadratic equation $m^* = \boldsymbol{z}^T\mathbf{A}\boldsymbol{z}$. Therefore the goal is still to compute such m^* in order to solve (6.1) and this is feasible because of $\vartheta(\tau; A)$'s modularity, which is the consequence of Poisson formula (details see the (appendix A.3)

$$\vartheta(\tau; A) = (i/2\tau)^{n/2}(det\Lambda)^{1/2}\vartheta(-1/4\tau, A^{-1}) \tag{6.3}$$

Let h be any positive integer such that $h\mathbf{A}^{-1}$ is also an integral matrix (in practice h can be selected to be the minimal one, denoted $l(\mathbf{A})$), by the same calculations as in section 2.3 we can get

$$\vartheta(\tau/(4h\tau + 1); A) = (4h\tau + 1)^{n/2}\vartheta(\tau; A) \tag{6.4}$$

$$and\ \ \vartheta(\tau + 1; A) = \vartheta(\tau; A) due\ to\ A's\ integrality \tag{6.5}$$

hence:

Lemma 6.1. *For any n-by-n integral and positive-definite symmetric matrix $\boldsymbol{A}$ and the integer h such that $h\boldsymbol{A}^{-1}$ is also integral, $\vartheta(\tau; \boldsymbol{A})$ is a modular form of weight $n/2$ with respect to the congruence subgroup generated by*

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \ and\ \begin{bmatrix} 1 & 0 \\ 4h & 1 \end{bmatrix}$$

i. e., $\vartheta(\tau; \boldsymbol{A}) \in M_{n/2}(J(h))$ □

The algorithm to solve the generalized SVP (6.1) on input matrix $\mathbf{A}$ can be constructed in almost the same way as in section 4 with the only (technical) differences that the sampling subroutines now work in lattice $\mathbb{Z}^n$. For example, D_t is now a distribution for lattice vectors in $\mathbb{Z}^n$

$$D_t(z) \equiv \exp(-2\pi z^T\mathbf{A}zt)/\sum_{z'\in\mathbb{Z}^n}\exp(2\pi i\tau z'^T\mathbf{A}z't)\ for\ z\ in\mathbb{Z}^n \tag{6.6}$$

and all other sampling distributions are modified in this way. In summary we have

Theorem 6.2. *There exists a randomized algorithm to solve the generalized optimization SVP (6.1) with correctness probability at least 1-ε on input the n-by-n matrix instance A with level $l(\boldsymbol{A})$, in time complexity of $n^{4+2n}l(A)^{2n+12}poly(n, S)log(1/\varepsilon)+poly(n, S, l(\boldsymbol{A}))$ and space complexity of $poly(n, S, l(\boldsymbol{A}))$ where $S = \max_{1\leq i\leq n}$bit-size of $\boldsymbol{A}$'s entries.*

In addition, this algorithm can be parallelized to be in time complexity of $poly(n, S)$ on $n^{(\alpha+1)2n}poly(n, S)log(1/\varepsilon)$ processors where $S = \max_{1\leq i\leq n}$bit-size of each entry in matrix $\boldsymbol{A}$. □

As remarked on theorem 5.3, it is not really necessary for the algorithm to store all the outputs from the oracle in a batch. In stead it can get these outputs "on-line" and in sequence when needed. As a result, the space complexity can be actually reduced to only $poly(n, S)$, independent of the level $l(\mathbf{A})$, while the time complexity's asymptotic bounds are unchanged

6.2 Dealing with SVP and CVP in Ideal Lattices

For the number field K of degree n, its integer ring O_K and any (fractional) ideal J are ideal lattices [5,14] of dimension n which have indispensable effect, e. g., in constructing innovative cryptography schemes for trusted cloud computing [7] based upon variants of SVP, CVP or other related computationally hard problems.

Ideal lattices have special algebraic properties the general lattices don't have. The associated theta functions have more special properties on which basis we might develop more efficient (and more technically involved) algorithms to estimate $\theta(\tau; \Lambda)$ and $\theta(\tau; \Lambda, \vec{u})$ in our framework. Applying our approach in this paper to such case is helpful not only to developing more efficient algorithms but also to disclosing how the problem's computational hardness is impacted by the number field's intrinsic algebraic properties, an interesting and valuable open problem for us to proceed in the future.

References

[1] D. Aharonov and Regev. Lattice problems in NP∩coNP. *J. ACM*, 52(5):749–765, 2005.

[2] M. Ajati, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC'01*, pages 601–610, 2001.

[3] M. Ajati, R. Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *IEEE Conference on Computational Complexity*, pages 53–57, 2002.

[4] W. Banaszczk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635m, 1993.

[5] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups.* Springer-Verlag, 3rd ed edition, 1998.

[6] F. Diamond and J.Shurman. *A first course in modular forms.* Springer-Verlag, Berlin, 2005.

[7] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. 41st ACM STOC*, pages 169–178, 2009.

[8] C. Gentry, C. Peikert, and V. Vaikuntananthan. How to use a short basis: trapdoors for hard lattices and new cryptographic constructions. In *Proc. 40st ACM STOC*, pages 197–206, 2008.

[9] G. H. Hanrot, X. Pujol, and D. Stehle. Algorithms for the shortest and closest lattice problems. In *Proc. IWCC*, 2011.

[10] I. Haviv and O. Regev. Hardness of the covering radius problem on lattices. In *IEEE CCC'06*, pages 145–158, 2006.

[11] R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proc. STOC'83*, pages 193–206, 1983.

[12] R. Kannan. Mincowski' s convex body theorem and integer programming. *Math. Oper.. Res.*, 12(3):415–440, 1987.

[13] N. Koblitz. *Introduction to elliptic curves and modular forms.* Springer-Verlag, 1993.

[14] W-Q Li. *Number theory with applications.* World Science Publication, 1996.

[15] D. Macciancio. Efficient reductions amomg lattice problems. In *Proc. SODA'08*, pages 84–93, 2008.

[16] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoii cell computations. *SIAM J. Comput*, 2012. Special Issue on STOC'2010.

[17] D. Micianccio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective.* Kluwer Academic Publishers, Boston, Massachusetts, 2002.

[18] D. Micianccio and O. Regev. Worst-case to average-case reductions based-on gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007.

[19] P. Q. Nguyen and B. Vallee(eds). *The LLL Algorithm: Survey and Applications.* Springer-Verlag, 2009.

[20] P. Q. Nguyen and T. Vidick. Sieve algorithms for the shortest vector problem are practical. In *Journal of Mathematical Cryptology*, volume 2, pages 181–207, 2008.

[21] C. Peikert. An efficient and parallel gaussian sampler for lattices. In *Crypto.*, pages 80–97, 2010.

[22] O. Regev. *Lecture notes of lattices in computer science.* Dept. of Computer Science., Tel Aviv Univ, 2004. `http://www.cs.tau.il/~odedr`.

[23] R. Vershynin. *Introduction to the non-asymptotic analysis of random matrices*, chapter 5 of: Compressed Sensing, Theory and Applications, pages 210–268. Cambridge University Press, 2012.

[24] X. Wang and D. Pei. *Modular forms with integral and half-integral weights (in English).* Science Press, Beijing, 2011.

Appendix A Poisson Formulas

For reading convenience, in this appendix the general Poisson formula is deduced with applications to the lattice-associated theta functions.

A.1 The General Formula

Let n-dimensional Fourier transformation

$$\tilde{f}(\vec{\xi}) \equiv \int_{R^n} d^n x f(\vec{x}) \exp(-2\pi i < \vec{x}, \vec{\xi} >)$$

on smooth function $f(x)$ quickly decreasing when $|x| \to \infty$. Let u be arbitrary vector in $\mathbb{R}^n$, the most general Poisson formula is

$$\sum_{x\in\Lambda} f(\vec{x}+\vec{u}) = det\Lambda^* \sum_{y\in\Lambda^*} \tilde{f}(y) exp(2\pi i < \vec{u}, \vec{y} >) \tag{A.1}$$

In particular (setting $\boldsymbol{u}$ to be zero-vector)$\sum_{x\in\Lambda} f(\vec{x}) = det\Lambda^* \sum_{y\in\Lambda^*} \tilde{f}(y)$

Proof. Calculate the Fourier expansion of the following Λ-periodic function

$$F(u;\Lambda) \equiv \sum_{x\in\Lambda} f(\vec{x}+\vec{u}) = \sum_{y\in\Lambda^*} a(\vec{y}) exp(2\pi i < \vec{u}, \vec{y} >)$$

we have

$$\begin{aligned} a(y) &= (det\Lambda)^{-1} \int_{Parallelotope(\Lambda)} d^n u F(\vec{u};\Lambda) exp(-2\pi i < \vec{u}, \vec{y} >) \\ &= det\Lambda^* \sum_{x\in\Lambda} \int_{Parallelotope(\Lambda)} d^n u f(\vec{x}+\vec{u}) exp(-2\pi i < \vec{u}, \vec{y} >) \\ &= det\Lambda^* \sum_{x\in\Lambda} \int_{x+Parallelotope(\Lambda)} d^n u f(\vec{u}) exp(-2\pi i < \vec{u}, \vec{y} >) \\ &= det\Lambda^* \int_{\mathbb{R}^n} d^n u f(\vec{u}) exp(-2\pi i < \vec{u}, \vec{y} >) \\ &= det\Lambda^* \tilde{f}(\vec{y}) \end{aligned}$$

Then (A.1) follows □

Equivalently, (A.1)'s generalized functional version is

$$\sum_{x\in\Lambda} \exp(-2\pi i < \vec{x}+\vec{u}, \vec{\xi} >) = det\Lambda^* \sum_{y\in\Lambda^*} \delta^n(\vec{\xi}-\vec{y}) exp(-2\pi i < \vec{u}, \vec{y} >)$$

A.2 Poisson Formulas for Lattice-Associated $\theta(\tau;\Lambda,u,v)$:

$$\theta(\tau;\Lambda,\vec{u},\vec{v}) \equiv \sum_{\vec{x}\in\Lambda} exp(-2\pi i|\vec{x}-\vec{u}|^2 + 2\pi i < \vec{x},\vec{v} >) \tag{A.2}$$

where Λ is a lattice (unnecessary to be rational) in $\mathbb{R}^n$, $||$ denotes the ℓ^2 norm in R^n, $\boldsymbol{u}$ and $\boldsymbol{v}$ are arbitrary vectors in R^n, $\tau = \sigma + it$ is a complex variable on the upper-half complex plane. $\theta(\tau;\Lambda,\vec{u},\vec{v})$ is a template of a few lattice-associated theta functions. Its Poisson formula is:

$$\vartheta(\tau;\Lambda,\vec{u},\vec{v}) = (i/2\tau)^{n/2} det\Lambda^* exp(2\pi i < \vec{u},\vec{v} >)\theta(-1/4\tau;\Lambda^*,\vec{v},-\vec{u}) \tag{A.3}$$

By $\Lambda^{**} = \Lambda$, the equivalent version is

$$\vartheta(\tau;\Lambda^*,\vec{u},\vec{v}) = (i/2\tau)^{n/2} det\Lambda exp(2\pi i < \vec{u},\vec{v} >)\theta(-1/4\tau;\Lambda^*,\vec{v},-\vec{u}) \tag{A.4}$$

Proof. Let $f(x) \equiv exp(-|x|^2/2\sigma^2)$, direct calculation shows that

$$\tilde{f}(\vec{\xi}) = (2\pi\sigma^2)^{n/2}\exp(-2\pi^2\sigma^2|\xi|^2)$$

It follows from (A.1) that for any real σ

$$\begin{aligned}
&\sum_{x\in\Lambda} \exp(-|\vec{x}-\vec{u}|^2)/2\sigma^2 + 2\pi i < \vec{x},\vec{y} >)\\
=&(2\pi\sigma^2)^{n/2} det\Lambda^* \sum_{y\in\Lambda^*} \exp(-2\pi^2\sigma^2|\vec{y}-\vec{v}|^2 - 2\pi i < \vec{u},\vec{y}-\vec{v} >)\\
=&(2\pi\sigma^2)^{n/2}\exp(2\pi i < \vec{u},\vec{v} >)det\Lambda^* \sum_{y\in\Lambda^*} \exp(-2\pi^2\sigma^2|\vec{y}-\vec{v}|^2 - 2\pi i < \vec{u},\vec{y} >)
\end{aligned}$$

Let $1/2\sigma^2 = -2\pi i\tau$ and because the above functions on both sides are holomorphic wherever $Im\tau > 0$, (A.3) follows due to the principle of analytic continuation. □

A.3 Poisson Formula for Matrix-Associated $\theta(\tau;\mathbf{A})$:

$$\vartheta(\tau;A) \equiv \sum_{z\in\mathbb{Z}^n} \exp(2\pi i\tau z^T Az) \tag{A.5}$$

where $\mathbf{A}$ is a positive-definite symmetric matrix. Let $f(\boldsymbol{x}) \equiv \exp(-|\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}|/2\sigma^2)$, by calculation its Fourier transformation

$$\tilde{f}(\xi) = (det\mathbf{A})^{-1/2}(2\pi\sigma^2)^{n/2}\exp(-2\pi^2\sigma^2\xi^T\mathbf{A}^{-1}\xi)$$

It follows from (A.1) that for any real number Υ

$$\sum_{z\in\mathbb{Z}^n}\exp(-x^T\mathbf{A}x/2\sigma^2)=(det\mathbf{A})^{-1/2}(2\pi\sigma^2)^{n/2}\sum_{y\in\mathbb{Z}^n}exp(-2\pi^2\sigma^2y^TA^{-1}y)$$

By setting $1/2\sigma^2=-2\pi i\tau$ we derive the Poisson formula

$$\vartheta(\tau:\mathbf{A})=(i/2\tau)^{n/2}(det\Lambda)^{1/2}\vartheta(-1/4\tau,\mathbf{A}^{-1}) \tag{A.6}$$