Foundations of Software Technology and Theoretical Computer Science (Bangalore) 2008.
Editors: R. Hariharan, M. Mukund, V. Vinay; pp 25-36

Some Sieving Algorithms for Lattice Problems

V. Arvind and Pushkar S. Joglekar
Institute of Mathematical Sciences
C.I.T Campus,Chennai 600 113, India
{arvind,pushkar}@imsc.res.in

ABSTRACT. We study the algorithmic complexity of lattice problems based on the sieving technique due to Ajtai, Kumar, and Sivakumar [AKS01]. Given a k-dimensional subspace $M \subseteq \mathbb{R}^n$ and a full rank integer lattice $\mathcal{L} \subseteq \mathbb{Q}^n$, the *subspace avoiding problem* SAP, defined by Blömer and Naewe [BN07], is to find a shortest vector in $\mathcal{L} \setminus M$. We first give a $2^{O(n+k\log k)}$ time algorithm to solve *the subspace avoiding problem*. Applying this algorithm we obtain the following results.

1. We give a $2^{O(n)}$ time algorithm to compute i^{th} successive minima of a full rank lattice $\mathcal{L} \subset \mathbb{Q}^n$ if i is $O(\frac{n}{\log n})$.
2. We give a $2^{O(n)}$ time algorithm to solve a restricted *closest vector problem CVP* where the inputs fulfil a promise about the distance of the input vector from the lattice.
3. We also show that unrestricted CVP has a $2^{O(n)}$ exact algorithm if there is a $2^{O(n)}$ time exact algorithm for solving CVP with additional input $v_i \in \mathcal{L}, 1 \le i \le n$, where $\|v_i\|_p$ is the i^{th} successive minima of $\mathcal{L}$ for each i.

We also give a new approximation algorithm for SAP and the *Convex Body Avoiding problem* which is a generalization of SAP. Several of our algorithms work for *gauge* functions as metric, where the gauge function has a natural restriction and is accessed by an oracle.

1 Introduction

Fundamental algorithmic problems concerning integer lattices are the shortest vector problem (SVP) and the closest vector problem(CVP). Given a lattice $\mathcal{L} \subset \mathbb{R}^n$ by a basis, the shortest vector problem (SVP) is to find a shortest nonzero vector in $\mathcal{L}$ w.r.t. some metric given by a *gauge* function in general (usually the ℓ_p norm for some p). Likewise, the closest vector problem (CVP) takes as input a lattice $\mathcal{L} \subset \mathbb{R}^n$ and vector $v \in \mathbb{R}^n$ and asks for a $u \in \mathcal{L}$ closest to v w.r.t. a given metric. These problems have polynomial-time approximation algorithms based on the celebrated LLL algorithm for basis reduction [LLL82].

The fastest known exact deterministic algorithms for SVP and CVP have running time $2^{O(n\log n)}$ [Kan87] (also see [Bl00]). More recently, Ajtai, Kumar and Sivakumar in a seminal paper [AKS01] gave a $2^{O(n)}$ time *randomized* exact algorithm for SVP. Subsequently, in [AKS02] they gave a $2^{O(n)}$ time randomized approximation algorithm for CVP. Their algorithms are based on a generic sieving procedure (introduced by them) that exploits the underlying geometry. Recently, Blömer and Naewe [BN07] gave a different $2^{O(n)}$ time randomized approximation algorithm for CVP, also based on the AKS sieving technique.

For $1 \le i \le n$, the i^{th} *successive minima* $\lambda_i(\mathcal{L})$ is defined as the smallest r such that a ball of radius r around origin contains at least i linearly independent lattice vectors. The successive minimas $\lambda_i(\mathcal{L})$ are important lattice parameters. A classical problem is the *successive minima problem* SMP of finding for a given lattice $\mathcal{L}$, n linearly independent vectors

FSTTCS 2008
IARCS Annual Conference on
Foundations of Software Technology and Theoretical Computer Science
http://drops.dagstuhl.de/opus/volltexte/2008/1738

$v_1, v_2, \ldots, v_n \in \mathcal{L}$ such that $\|v_i\|$ is at most $\lambda_i(\mathcal{L})$. This problem clearly subsumes the *shortest independent vectors problem* SIVP where one wants to find linearly independent vectors $v_1, v_2, \ldots, v_n \in \mathcal{L}$ such that $\|v_i\| \leq \lambda_n(\mathcal{L})$. Given a k-dimensional subspace $M \subseteq \mathbb{R}^n$ and a full rank integer lattice $\mathcal{L} \subseteq \mathbb{Q}^n$, the *subspace avoiding problem* SAP, is to find a shortest vector in $\mathcal{L} \setminus M$. The paper [BN07] gives $2^{O(n)}$ time approximation algorithm for these problems.

No exact $2^{O(n)}$ time randomized algorithm is known for CVP or SMP. Recently, Micciancio has shown [Mi08] that CVP is polynomial-time equivalent to several lattice problems, including SIVP and SMP, under deterministic polynomial time rank-preserving reductions. This perhaps explains the apparent difficulty of finding a $2^{O(n)}$ time exact algorithm for CVP or SMP, because SVP reduces to all of these problems but no reduction is known in the other direction. In particular, the reductions in [Mi08] yield $2^{O(n \log n)}$ time exact algorithms for SAP, SMP and SIVP, whereas [BN07] gives $2^{O(n)}$ time randomized approximation algorithm for these problems.

Our results

In this paper we consider some natural restrictions of these problems that can be exactly solved in $2^{O(n)}$ time. We obtain these results giving a $2^{O(n+k \log k)}$ algorithm to solve SAP where n is the rank of the lattice and k is the dimension of the subspace.

As our first result we show that given a full rank lattice $\mathcal{L} \subset \mathbb{Q}^n$ there is $2^{O(n)}$ time randomized algorithm to compute linearly independent vectors $v_1, v_2, \ldots, v_i \in \mathcal{L}$ such that $\|v_i\| = \lambda_i(\mathcal{L})$ if i is $O(\frac{n}{\log n})$. Given a full rank lattice $\mathcal{L} \subset \mathbb{Q}^n$ and $v \in \mathbb{Q}^n$ we also give a $2^{O(n)}$ time algorithm to solve CVP($\mathcal{L}, v$) if the input $(v, \mathcal{L})$ fulfils the promise $d(v, \mathcal{L}) \leq \frac{\sqrt{3}}{2}\lambda_{O(\frac{n}{\log n})}(\mathcal{L})$.

We show that CVP can be solved in $2^{O(n)}$ time if there is a $2^{O(n)}$ time algorithm to compute a closest vector to v in $\mathcal{L}$ where $v \in \mathbb{Q}^n$, $\mathcal{L} \subset \mathbb{Q}^n$ is a full rank lattice and $v_1, v_2, \ldots, v_n \in \mathcal{L}$ such that $\|v_i\|_p$ is equal to i^{th} successive minima of $\mathcal{L}$ for $i = 1$ to n are given as an additional input to the algorithm. As a consequence, we can assume that successive minimas are given for free as an input to the algorithm for CVP. We believe that using basis reduction techniques from [Kan87] one might be able to exploit the information about successive minimas of the lattice to get a better algorithm for CVP.

We give a new $2^{O(n+k \log 1/\epsilon)}$ time randomized algorithm to solve $1 + \epsilon$ approximation of SAP, where n is rank of the lattice and k is the dimension of subspace. We get better approximation guarantee than the one in [BN07] parametrised on k. We also consider a generalization of SAP (the *convex body avoiding* problem) and give a singly exponential approximation algorithm for the problem.

2 Preliminaries

A lattice $\mathcal{L}$ is a discrete additive subgroup of R^n, n is called dimension of the lattice. For algorithmic purposes we can assume that $\mathcal{L} \subseteq \mathbb{Q}^n$, and even in some cases $\mathcal{L} \subseteq \mathbb{Z}^n$. A lattice is usually specified by a basis $B = \{b_1, \cdots, b_m\}$, where $b_i \in \mathbb{Q}^n$ and b_i's are linearly independent. m is called the rank of the lattice. If the rank is n the lattice is said to be a *full rank* lattice. Although most results in the paper hold for general lattices, for convenience we mainly consider only full-rank lattices. For $x \in \mathbb{Q}^n$ let size(x) denote the number of bits for the standard binary representation as an n-tuple of rationals. Let size($\mathcal{L}$) denote $\sum_i$ size($\mathrm{b_i}$). Next we recall the definition of gauge functions.

DEFINITION 1.[Si45] *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called a* gauge function *if it satisfies following properties:*

1. $f(x) > 0$ *for all* $x \in \mathbb{R}^n \setminus \{0\}$ *and* $f(x) = 0$ *if* $x = 0$.
2. $f(\lambda x) = \lambda f(x)$ *for all* $x \in \mathbb{R}^n$ *and* $\lambda \in \mathbb{R}$.
3. $f(x + y) \leq f(x) + f(y)$ *for all* $x, y \in \mathbb{R}^n$.

For $v \in \mathbb{R}^n$ we denote $f(v)$ by $\|v\|_f$ and call it norm of v with respect to the gauge function f. It is easy to see that any l_p norm satisfies all the above properties. Thus gauge functions generalize the usual l_p norms. A gauge function f defines a natural metric d_f on $\mathbb{R}^n$ by setting $d_f(x, y) = f(x - y)$ for $x, y \in \mathbb{R}^n$. For $x \in \mathbb{R}^n$ and $r > 0$, let $B_f(x, r)$ denote the f-ball of radius r with center x with respect to the gauge function f, defined as $B_f(x, r) = \{y \in \mathbb{R}^n | f(x - y) \leq r\}$. We denote the metric balls with respect to usual l_p norm by $B_p(x, r)$. Unless specified otherwise we always consider balls in $\mathbb{R}^n$. The next well-known proposition characterizes the class of all gauge functions.

PROPOSITION 2.[Si45] *Let $f : \mathbb{R}^n \to \mathbb{R}$ be any gauge function then a unit radius ball around origin with respect to f is a n dimensional bounded O-symmetric convex body. Conversely, for any n dimensional bounded O-symmetric convex body C, there is a gauge function $f : \mathbb{R}^n \to \mathbb{R}$ such that $B_f(0, 1) = C$.*

Given an f-ball of radius r around origin with respect to a gauge function f, from the Proposition 2 it follows that $B_f(0, r)$ is an O-symmetric convex body. It is easy to check that for any $r > 0$ and any constant c we have $\mathrm{vol}(B_f(0, cr)) = c^n \mathrm{vol}(B_f(0, r))$, where $\mathrm{vol}(C)$ denotes the volume of the corresponding convex body C (see e.g. [Si45]).

We now place a natural restriction on gauge functions. A gauge function f, given by oracle access, is a *nice gauge function* if it satisfies the following property: For some polynomial $p(n)$, $B_2(0, 2^{-p(n)}) \subseteq B_f(0, 1) \subseteq B_2(0, 2^{p(n)})$, i.e. there exists a Euclidean sphere of radius $2^{-p(n)}$ inside the convex body $B_f(0, 1)$, and $B_f(0, 1)$ is contained inside a Euclidean sphere of radius $2^{p(n)}$. Note that if f is a nice gauge function and $v \in \mathbb{Q}^n$ we have size($f(v)$)=poly(n,size(v)). For a nice gauge function f we can sample points from convex body $B_f(0, r)$ almost uniformly at random in poly(size(r),n) time using the Dyer-Frieze-Kannan algorithm [DFK91]. It is easy to check that all l_p norms $p \geq 1$ define nice gauge functions. The i^{th} successive minima of a lattice $\mathcal{L}$ with respect to ℓ_p norm is smallest $r > 0$ such that $B_p(0, r)$ contains at least i linearly independent lattice vectors. It is denoted by $\lambda_i^p(\mathcal{L})$.

Remarks: In this paper we consider lattice problems with respect to nice gauge functions. Let $\mathcal{L}$ be a lattice with basis $\{b_1, b_2, \ldots, b_n\}$ and f be a nice gauge function. Suppose B is a full rank $n \times n$ matrix with columns $b_1, b_2, \ldots, b_n$. Note that the linear transformation B^{-1} maps lattice $\mathcal{L}$ isomorphically to the standard lattice $\mathbb{Z}^n$. Furthermore, it is easy to see that the set $C = B^{-1}(B_f(0, 1))$ is an O-symmetric convex body. Hence, by Proposition 2 it follows that $C = B_g(0, 1)$ for some gauge function g. As f is a nice gauge function, it easily follows that g is also a nice gauge function.

Thus, our algorithms that work for nice gauge functions can be stated for the the standard lattice $\mathbb{Z}^n$ and a nice gauge function g. However, some of our results hold only for ℓ_p norms. Thus, to keep uniformity we allow our algorithms to take arbitrary lattices as input even when the metric is give by a nice gauge function.

3 A Sieving Algorithm for SAP

In this section we present a different analysis of the AKS sieving [AKS01, Re04] applied to the Subspace Avoiding Problem (SAP). Our analysis is quite different from that due to Blömer and Naewe [BN07] and gives us improved running time for computing a $1+\epsilon$ approximate solution.

Recall that an input instance of the subspace avoiding problem (SAP) consists of $(\mathcal{L}, M)$ where $\mathcal{L} \subset \mathbb{Q}^n$ is a full rank lattice and $M \subset \mathbb{R}^n$ is a subspace of dimension k. The SAP problem is to find a vector $v \in \mathcal{L} \setminus M$ with least norm with respect to a nice gauge function f.

We give an intuitive outline of our approximation algorithm: Our analysis of AKS sieving will use the fact that the sublattice $\mathcal{L} \cap M$ of $\mathcal{L}$ is of rank k. We will use the AKS sieving procedure to argue that we can sample $2^{O(n+k\log(1/\epsilon))}$ points from *some* coset of $\mathcal{L} \cap M$ in $2^{O(n+k\log(1/\epsilon))}$ time. We can then apply a packing argument in the coset (which is only k-dimensional) to obtain points in the coset that are close to each other. Then, with a standard argument following the original AKS result [AKS01] we can conclude that their differences will contain a good approximation.

Suppose, without loss of generality, that the input lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is n-dimensional given by a basis $\{b_1, \cdots, b_n\}$, so that $\mathcal{L} = \sum_{i=1}^n \mathbb{Z} \cdot b_i$. Let us fix a nice gauge function f and let $v \in \mathcal{L}$ denote a shortest vector in $\mathcal{L} \setminus M$ with respect to gauge function f, i.e. $f(x)$ for $x \in \mathcal{L} \setminus M$ attains minimum value at $x = v$. Let $s = \text{size}(\mathcal{L}, \text{M})$ denote the input size (which is the number of bits for representing the vectors b_i and the basis for M). As v is a shortest vector in $\mathcal{L} \setminus M$ and f is a nice gauge function it is quite easy to see that $size(f(v))$ is bounded by a polynomial in s. Thus, we can scale the lattice $\mathcal{L}$ to ensure that $2 \leq f(v) \leq 3$. More precisely, we can compute polynomially many scaled lattices from $\mathcal{L}$, so that $2 \leq f(v) \leq 3$ holds for at least one scaled lattice. Thus, we can assume that $2 \leq f(v) \leq 3$ holds for the lattice $\mathcal{L}$.

We first describe the AKS sieving procedure [AKS01] for any gauge function, analyze its running time and explain its key properties. The following lemma is crucially used in the algorithm.

LEMMA 3.*[Sieving Procedure] Let $f : \mathbb{R}^n \to \mathbb{R}$ be any gauge function. Then there is a sieving procedure that takes as input a finite set of points $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_N\} \subseteq B_f(0, r)$, and in $N^{O(1)}$ time it outputs a subset of indices $S \subset [N]$ such that $|S| \leq 5^n$ and for each $i \in [N]$ there is a $j \in S$ with $f(\mathbf{v}_i - \mathbf{v}_j) \leq r/2$.*

Proof. The sieving procedure is exactly as described in Regev's lecture notes [Re04]. The sieving procedure is based on a simple greedy strategy. We start with $S = \emptyset$ and run the following step for all elements $v_i, 1 \leq i \leq N$. At the i^{th} step we consider v_i. If $f(v_i - v_j) > r/2$ for all $j \in S$ include i in the set S and increment i. After completion, for all $i \in [N]$

there is a $j \in S$ such that $f(v_i - v_j) \leq r/2$. The bound on $|S|$ follows from a packing argument combined with the fact that $\mathrm{vol}(B_f(0, cr)) = c^n \mathrm{vol}(B_f(0, r))$ for any $r > 0$ and a constant $c > 0$. More precisely, for any two points $v_i, v_j \in S$ we have $f(v_i - v_j) > r/2$. Thus, all the convex bodies $B_f(v_i, r/4)$ for $v_i \in S$ are mutually disjoint and are contained in $B_f(0, r + r/4)$. Also note that $\mathrm{vol}(B_f(0, dr)) = d^n \mathrm{vol}(B_f(0, r))$ for any constant $d > 0$. It follows that $5^n \mathrm{vol}(B_f(v_i, r/4)) \geq \mathrm{vol}(B_f(0, r + r/4))$. Hence, $|S| \leq 5^n$. The second property of S is guaranteed by the sieving procedure. ■

Next, our algorithm follows the usual AKS random sampling procedure. Let $R = n \cdot max_i \|b_i\|_f$. It is clear that size($R$) is polynomial in s since f is a nice gauge function. Let $B_f(0, 2)$ denote the f-ball of radius 2 around the origin. Since we have an oracle for membership in $B_f(0, 2)$ and f is a nice gauge function we can almost uniformly sample from $B_f(0, 2)$ using the Dyer-Frieze-Kannan algorithm [DFK91]. Let $x_1, x_2, \cdots, x_N$ denote such a random sample, for $N = 2^{c \cdot (n + k \log(1/\epsilon))} \cdot \log R$ where the constant $c > 0$ will be suitably chosen. Now, using the lattice $\mathcal{L}$ we can round off the points x_i. More precisely, we express $x_i = \Sigma_j \alpha_{ij} b_j$ for rationals α_{ij}. Then, from each vector x_i we compute the vector $y_i = \Sigma_j \beta_{ij} b_j$, where $0 \leq \beta_{ij} < 1$, by adding appropriate integral multiples of the b_j's to the expression for x_i. Thus, the points $y_1, \cdots, y_N$ are in the interior of the fundamental parallelepiped of $\mathcal{L}$, and each $x_i - y_i \in \mathcal{L}$. We denote this by $y_i = x_i (\text{mod } \mathcal{L})$. We now have the set of N pairs $P = \{(x_i, y_i) \mid i \in [N]\}$, where $x_i - y_i$ are lattice points. Since y_i lie inside the fundamental parallelepiped we have $\|y_i\|_f \leq n \cdot max_i \|b_i\|_f = R$ for $i = 1$ to N.

Now, we apply the AKS sieving procedure in Lemma 3 to the set $\{y_1, y_2, \cdots, y_N\}$. The result is a subset $S \subset [N]$ of at most 5^n indices such that for each $i \in [N]$ there is some $j \in S$ such that $f(y_i - y_j) \leq R/2$. We remove from P all (x_j, y_j) for $j \in S$ and replace each remaining $(x_i, y_i) \in P$ by a corresponding $(x_i, y_i - (y_j - x_j))$, where $j \in S$ is the first index such that $f(y_i - y_j) \leq R/2$. After the sieving round, the set P has the property that for each $(x_i, z_i) \in P$ we have $x_i - z_i \in \mathcal{L}$ and $f(x_i - z_i) \leq 4 + R/2$, and P has shrunk in size by at most 5^n. We continue with $O(\log\ R)$ sieving rounds so that we are left with a set P with $N - O(\log\ R)5^n$ pairs (x_i, z_i) such that $x_i - z_i \in \mathcal{L}$ and $f(x_i - z_i) \leq 8$. We can ensure that $|P| \geq 2^{c'(n + k \log(1/\epsilon))}$ for an arbitrary constant c' by appropriately choosing constant c. The vectors, $x_i - z_i$ for $(x_i, z_i) \in P$ follows some distribution among lattice points inside $B_f(0, 8)$. Next, we need following simple proposition.

PROPOSITION 4. *Let $\mathcal{L} \subset \mathbb{R}^n$ be a rank n lattice, $v \in \mathcal{L}$ such that $2 \leq f(v) \leq 3$ for a nice gauge function f. Consider the convex regions $C = B_f(-v, 2) \cap B_f(0, 2)$ and $C' = B_f(v, 2) \cap B_f(0, 2)$. Then $C' = C + v$ and vol$(C) =$ vol$(C') = \Omega(\frac{vol(B_f(0,2))}{2^{O(n)}})$.*

Proposition 4 is easy to prove since $B_f(-v/2, 1/2) \subseteq C, B_f(v/2, 1/2) \subseteq C'$. Note that we have picked $x_1, \ldots, x_N$ uniformly at random from $B_f(0, 2)$,where $N = 2^{c \cdot (n + k \log(1/\epsilon))} \cdot \log R$. By Proposition 4, the point x_i is in C with probability at least $2^{-O(n)}$. Hence by choosing the constant c large enough we can ensure that with high probability there is a subset $Z \subseteq P$ such that $|Z| \geq 2^{c_1 (n + k \log(1/\epsilon))}$ for a constant c_1 and for all $(x_i, z_i) \in Z$, $x_i \in C$. We now prove the main theorem of this section.

THEOREM 5. *Let $\mathcal{L} \subset \mathbb{Q}^n$ be a full rank lattice and let $v \in \mathcal{L} \setminus M$ such that $2 \leq f(v) \leq 3$ for a given gauge function f and $f(v) \leq f(x)$ for all $x \in \mathcal{L} \setminus M$. Let $\epsilon > 0$ be an arbitrary constant. Then there is a randomized algorithm that in time $2^{O(n+klog(1/\epsilon))}$.poly(size($\mathcal{L}$)) computes a set P of pairs (x_i, z_i) such that $|P| \geq 2^{c' \cdot (n+k\log(1/\epsilon))}$ for a constant c' and $f(x_i - z_i) \leq 8$ for all $(x_i, z_i) \in P$. Moreover, $z_i - x_i \in \mathcal{L}$ are such that with probability $1 - 2^{-O(n)}$ there is a pair of points $(x_i, z_i), (x_j, z_j) \in P$ such that $v + u = (x_i - z_i) - (x_j - z_j)$ for a vector $u \in \mathcal{L} \cap M$ with $f(u) \leq \epsilon$.*

Proof.

Consider the set P of pairs (x_i, z_i), obtained after the AKS sieving as described above, such that $|P| \geq 2^{c'(n+k\log(1/\epsilon))}$, and $f(x_i - z_i) \leq 8$ for all $(x_i, z_i) \in P$. We know that by choosing c large enough we can ensure that with high probability there is $Z \subseteq P$ such that $|Z| \geq 2^{c_1(n+k\log(1/\epsilon))}$ for any constant c_1 and for all $(x_i, z_i) \in Z$, $x_i \in C$.

Note that $\mathcal{L} \cap M$ is a rank k sublattice of $\mathcal{L}$. We will now analyze Z using the cosets of the sublattice $\mathcal{L} \cap M$.

Write Z as a partition $Z = \bigcup_{j=1}^{m} Z_j$, where for each Z_j there is a distinct coset $(\mathcal{L} \cap M) + v_j$ of $\mathcal{L} \cap M$ in $\mathcal{L}$ such that $z_i - x_i \in (\mathcal{L} \cap M) + v_j$ for all $(x_i, z_i) \in Z_j$. Let $Z'_j = \{z_i - x_i \mid (x_i, z_i) \in Z_j\}$. Suppose $u_j \in Z'_j \subseteq (\mathcal{L} \cap M) + v_j$ for $j = 1$ to m.

CLAIM 6.*[Coset sampling] By choosing constant c_1 large enough we can ensure that there is an index t, $1 \leq t \leq m$ such that $|Z_t| \geq 2^{c_2(n+k\log(1/\epsilon))}$ for any constant c_2.*

Proof of Claim Note that u_i and u_j for $i \neq j$ lie in different cosets of $\mathcal{L} \cap M$. So $u_i - u_j \notin M$. Since v is a shortest f-vector in $\mathcal{L} \setminus M$ with $2 \leq f(v) \leq 3$, we have $f(u_i - u_j) \geq 2$. Hence unit radius f-balls around u_i's are disjoint. Note that $B_f(u_i, 1) \subset B_f(0, 9)$ for $i = 1$ to m. Since $\text{vol}(B_f(0,9))/\text{vol}(B_f(0,1)) \leq 2^{dn}$ for some constant d, we have $m \leq 2^{dn}$. We have $|Z| \geq 2^{c_1(n+k\log(1/\epsilon))}$ and Z is partitioned as $Z = \bigcup_{j=1}^{m} Z_j$. So it is clear that by choosing c_1 large enough we can ensure that there is an index t, $1 \leq t \leq m$ such that $|Z_t| \geq 2^{c_2(n+k\log(1/\epsilon))}$ for any constant c_2. ∎

By renumbering the indices assume that $Z_t = \{(x_1, z_1), \ldots, (x_q, z_q)\}, q \geq 2^{c_2(n+k\log(1/\epsilon))}$. Let $\beta_i = z_i - x_i$ for $(x_i, z_i) \in Z_t$. Thus, each such β_i lies in the same coset $(\mathcal{L} \cap M) + v_\ell$.

CLAIM 7.*[Packing argument] By choosing the constant c_2 large enough we can ensure that there exists $(x_i, z_i), (x_j, z_j) \in Z_t, i \neq j$ such that $f(\beta_i - \beta_j) \leq \epsilon$.*

Proof of Claim Suppose for all $(x_i, z_i), (x_j, z_j) \in Z_t, i \neq j$ $f(\beta_i - \beta_j) \geq \epsilon$. We also have $f(\beta_i - \beta_j) \leq 16$ for $i, j \in [q]$. Let $\gamma_i = \beta_i - v_\ell \in \mathcal{L} \cap M \subset M$ for $i = 1$ to q. It is clear that $f(\gamma_i - \gamma_j) = f(\beta_i - \beta_j)$ for $i, j \in [q]$. Let $\{b_1, \ldots, b_k\}$ be an orthonormal basis of M. Consider the linear transformation $T : \mathrm{M} \rightarrow \mathbb{R}^k$ such that $T(b_i) = e_i$ for $i = 1$ to k, where $\{e_1, e_2, \ldots, e_k\}$ is a standard basis of $\mathbb{R}^k$. Let $\delta_i = T(\gamma_i)$ for $i = 1$ to q. By standard linear algebra it follows that T preserves distances between points with respect to any norm. In particular, we have $f(\gamma_i - \gamma_j) = f(\delta_i - \delta_j)$ for $i, j \in [q]$. So we have $\epsilon/2 \leq f(\delta_i - \delta_j) \leq 16$. As $\delta_i \in \mathbb{R}^k$ for $i \in [q]$, it follows that *k-dimensional* balls of radius $\epsilon/2$ around δ_i's are mutually disjoint. By a packing argument it follows that $|Z_t| \leq \frac{(16+\epsilon/2)^k}{(\epsilon/2)^k} = 2^{f(k\log(1/\epsilon))}$ for a constant f. This is a contradiction since choosing c_2 large enough we can ensure that $|Z_t| \geq 2^{c_2(n+k\log(1/\epsilon))} > 2^{f(k\log(1/\epsilon))}$.

We now complete the proof with a standard argument from [AKS01, Re04] using a modified distribution.

We have $(x_i, z_i), (x_j, z_j) \in Z_t \subset Z, i \neq j, x_i, x_j \in C$ such that $f(\beta_i - \beta_j) \leq \epsilon$ and $\beta_i - \beta_j \in \mathcal{L} \cap M$. Now, we apply the argument as explained in Regev's notes [Re04] to reason with a modified distribution of the x_i. Note that in the sieving procedure described before Theorem 5, each x_i is picked independently and uniformly at random from $B_f(0,2)$. Now, notice that we can replace the original distribution of x_i with a modified distribution in which we output x_i if it lies in $B_f(0,2) \setminus (C \cup C')$ and if $x_i \in C$ it outputs either x_i or $x_i + v$ with probability $1/2$ each. Similarly, if $x_i \in C' = C + v$ it outputs either x_i or $x_i - v$ with probability $1/2$ each. By Proposition 4 it follows that this modified distribution is also uniform on $B_f(0,2)$ (indeed, this distribution is required only for the purpose of analysis). Furthermore, we can replace each x_i by the modified distribution just before it is used in the algorithm for the first time. The reason we can do this is because the distribution of y_i's remains same even if we replace x_i by the modified distribution because $y_i = x_i(\mathrm{mod}\mathcal{L})$ and $v \in \mathcal{L}$. This is explained further in Regev's notes [Re04]. Now recall that we have $(x_i, z_i), (x_j, z_j) \in Z$ with $x_i, x_j \in C$ and $f(\beta_i - \beta_j) \leq \epsilon$. Putting it together with the above argument, it follows that with good probability the points (x_i, z_i) and $(x_j + v, z_j)$ are in the set P, where P is the set of pairs left after the sieving. This is easily seen to imply that with high probability we are likely to see the vector $v + (\beta_i - \beta_j)$ as the difference of $z_i - x_i$ and $z_j - x_j$ for some two pairs $(x_i, z_i), (x_j, z_j) \in P$. The theorem now follows since $f(\beta_i - \beta_j) \leq \epsilon$. ■

By choosing M as the 0-dimensional subspace we get a $2^{O(n)}$ algorithm for SVP with respect to any nice gauge function. As an immediate consequence of Theorem 5 we get a $1 + \epsilon$ approximation algorithm for SAP problem that runs in time $2^{O(n+k\log\frac{1}{\epsilon})} \cdot poly(size(\mathcal{L}, M))$.

Remarks: The $1 + \epsilon$ approximation algorithm in [BN07] for SAP has running time $2^{O(n\log\frac{1}{\epsilon})} \cdot poly(size(\mathcal{L}, M)))$. Our algorithm has running time $2^{O(n+k\log\frac{1}{\epsilon})}$ for computing $1 + \epsilon$ approximate solution. Put another way, for $k = o(n)$ we get a $2^{O(n)}$ time algorithm for obtaining $1 + 2^{-n/k}$ approximate solutions to SAP.

There is a crucial difference in our analysis of the AKS sieving and that given in [BN07]. In [BN07] it is shown that with probability $1 - 2^{-O(n)}$ the sieving procedure outputs a $1 + \epsilon$ approximate solution $u \in \mathcal{L} \setminus M$.

On the other hand, we show in Claim 6 that with probability $1 - 2^{-O(n)}$ the sieving procedure samples $2^{O(n+k\log(1/\epsilon)}$ lattice points in *some* coset of the sublattice $\mathcal{L} \cap M$ in $\mathcal{L}$. Then we argue that with probability $1 - 2^{-O(n)}$ the sample contains a lattice point u in $\mathcal{L} \cap M + v$ such that such that $d(u, v)$ is small, for some shortest vector v in $\mathcal{L} \setminus M$. We argue this in Claim 7 by a packing argument in the coset of $\mathcal{L} \cap M$. As $\mathcal{L} \cap M$ has rank k, the packing argument in k dimensions gives the improved running time for our approximation algorithm for the problem.

The fact that the AKS sampling contains many points from the same coset of $\mathcal{L} \cap M$ also plays crucial role in our exact algorithm for SAP shown in Theorem 12.

Corollary 8. *Given a rank n lattice $\mathcal{L}$ and a k-dimensional subspace $M \subset \mathbb{R}^n$, there is $1+\epsilon$ randomized approximation algorithm for SAP (for any nice gauge function) with running time $2^{O(n+k\log\frac{1}{\epsilon})} \cdot poly(size(\mathcal{L}, M))$.*

Proof. The algorithm will examine all $(z_i - x_i) - (z_j - x_j)$ for $(x_i, z_i), (x_j, z_j) \in P$ obtained after sieving and output that element in $\mathcal{L} \setminus M$ of minimum f-value. The proof of correctness and running time guarantee follows immediately from Theorem 5. ∎

4 Convex Body Avoiding Problem

In this section we consider a generalization of SAP: given a lattice $\mathcal{L}$ and a convex body C the problem is to find a shortest vector (w.r.t. ℓ_p norm) in $\mathcal{L} \setminus C$. We consider convex bodies C that are bounded and O-symmetric. We refer to this problem as the *Convex body Avoiding Problem* (CAP).

A set $S \subseteq \mathbb{R}^n$ is *O-symmetric* if $x \in S$ if and only if $-x \in S$. Notice that a subspace $M \subseteq \mathbb{R}^n$ is convex and O-symmetric (but not bounded).

The input to CAP is the lattice $\mathcal{L}$ and the convex body C, where C is given by a membership oracle. An algorithm can query the oracle for any $x \in \mathbb{R}^n$ to test if $x \in C$.

We give an approximation algorithm to solve CAP.

Theorem 9. *Given an integer lattice $\mathcal{L}$ of rank n and an O-symmetric convex body C in $\mathbb{R}^n$ given by a membership oracle, there is $1+\epsilon$ factor approximation algorithm to solve CAP (w.r.t. any ℓ_p norm) with running time $2^{O(n)\cdot\log(1/\epsilon)} \cdot poly(size(\mathcal{L}))$.*

Proof. It suffices to solve the problem for the case when C is n-dimensional. To see this, suppose C is contained in some k-dimensional subspace M of $\mathbb{R}^n$. We can find a basis for M with high probability by sampling vectors from C using the polynomial-time almost uniform sampling algorithm described in [DFK91]. Next, we compute the sublattice $\mathcal{L} \cap M$ and find a $(1+\epsilon)$ approximate solution u for the k-dimensional convex body avoidance for the lattice $\mathcal{L} \cap M$ and C. We also solve the SAP instance $(\mathcal{L}, M)$ and find a $(1+\epsilon)$ approximate solution $v \in \mathcal{L} \setminus M$ using Theorem 5. The shorter of vectors u and v is clearly a $(1+\epsilon)$ approximate solution for the input CAP instance.

Thus, we can assume C is n-dimensional. Let v be a shortest vector in $\mathcal{L} \setminus C$ which, as before, we can assume satisfies $2 \leq \|v\|_p \leq 3$ by considering polynomially many scalings of the lattice and the convex body. As in Theorem 5, we pick random points $x_1, \cdots, x_N$ from $B_p(0, 2)$ for $N = 2^{cn\log(1/\epsilon)} \cdot poly(s)$. The constant $c > 0$ will be suitably chosen later. Let $y_i = x_i (\text{mod } \mathcal{L})$ for $i = 1$ to N. We apply several rounds of the AKS sieving on the set $\{(x_1, y_1), \cdots, (x_N, y_N)\}$ until we are left with a set S of $2^{c_1 n\log(1/\epsilon)}$ pairs (x_i, z_i) such that $\|x_i - z_i\|_p \leq 8$. From proposition 4 it follows easily that with good probability we have $Z \subseteq S$ such that $|Z| \geq 2^{c_2 n\log(1/\epsilon)}$ and for all $(x_i, z_i) \in Z$ we have $x_i \in D \cup D'$ where $D = B_p(0, 2) \cap B_p(-v, 2)$ and $D' = B_p(0, 2) \cap B_p(v, 2)$. Note that the the constant c_2 can be chosen as large as we like by appropriate choice of c. Let $Z' = \{z_i - x_i \mid (x_i, z_i) \in Z\}$. Now consider ℓ_p ball of radius $\epsilon/2$ centered at each lattice point $\beta \in Z'$. It is clear that for all $\beta \in Z'$, $B_p(\beta, \epsilon/2) \subseteq B_p(0, 8 + \epsilon/2)$. If for all $\beta \in Z'$ ℓ_p balls $B_p(\beta, \epsilon/2)$ are mutually disjoint, by packing argument we get $|Z'| \leq \frac{(8+\epsilon/2)^n}{(\epsilon/2)^n} = 2^{c'n\log(1/\epsilon)}$ for a constant c'. We choose constant

c appropriately to ensure that $c_2 > c'$. This implies that there exists tuples $(x_i, z_i), (x_j, z_j) \in Z$ such that $\|\beta_i - \beta_j\| \leq \epsilon$, where $\beta_i = z_i - x_i$ and $\beta_j = z_j - x_j$. Let $\beta = \beta_i - \beta_j$. We claim that it is not possible that both $\beta + v, \beta - v$ lie inside the convex body C. Because this implies $v - \beta \in C$ since C is O-symmetric. Therefore $v = \frac{(\beta+v)+(v-\beta)}{2} \in C$, which contradicts with assumption $v \notin C$. So without loss of generality assume that $\beta + v \notin C$. Note that without loss of generality we can also assume that $x_i \in D'$ with good probability. Now, we apply the argument as explained in [Re04] to reason with a modified distribution of the x_i. As $x_i \in D'$ we can replace x_i by $x_i - v$. It is easy to see that after sieving with good probability there exists tuples $(x_i, z_i), (x_j, z_j) \in S$ such that $r_{i,j} = (z_i - x_i) - (z_j - x_j) = v + \beta_i - \beta_j$. Hence, $r_{i,j} = v + \beta \notin C$ and, clearly, $\|r_{i,j}\|_p \leq (1+\epsilon)\|v\|_p$ since $\|\beta_i - \beta_j\|_p \leq \epsilon$. It is easy to see that the algorithm runs in time $2^{O(n \log(1/\epsilon))} poly(size(\mathcal{L}))$. This completes the proof of the theorem. ■

5 Applications

The results of this section are essentially applications of ideas from Theorem 5 and Section 3.

First we describe an exact algorithm for SAP for ℓ_p norms. We prove our result for full rank lattices, but it is easy to see that the result holds for general lattices as well. Let $\mathcal{L} \subset \mathbb{Q}^n$ be a full rank integer lattice given by a basis $\{b_1, \cdots, b_n\}$ and let $M \subseteq \mathbb{R}^n$ is a subspace of dimension $k < n$. For any ℓ_p norm we give a randomized $2^{O(n+k\log k)} poly(s)$ time algorithm to find a shortest vector in $\mathcal{L} \setminus M$, where $s = size(\mathcal{L}, M)$. Our exact algorithm uses the same sieving procedure and analysis described in the proof of Theorem 5 in Section 3. As before, by considering polynomially many scalings of the lattice, we can assume that a shortest vector $v \in \mathcal{L} \setminus M$ satisfies $2 \leq \|v\|_p \leq 3$. We now describe the algorithm.

1. Let $N = 2^{cn} \log(n.max_i \|b_i\|_p)$. Pick $x_1, x_2, \cdots, x_N$ uniformly at random from $B_p(0, 2)$.
2. Let $y_i = x_i (\text{mod } \mathcal{L})$. Apply AKS sieving to the set $\{(x_1, y_1), \cdots, (x_N, y_N)\}$ as described in Section 3 until $\|x_i - z_i\|_p \leq 8$ for each pair (x_i, z_i) left after the sieving.
3. Let $P = \{(x_i, z_i) | i \in T\}, T \subset [N]$ be the set of tuples left after the sieving procedure. For all $i, j \in T$ compute lattice points $v_{i,j} = (z_i - x_i) - (z_j - x_j)$.
4. Let $w_{i,j}$ be a closest lattice vector to $v_{i,j}$ in the rank k lattice $\mathcal{L} \cap M$ (found using Kannan's exact CVP algorithm [Kan87]), and let $r_{i,j} = v_{i,j} - w_{i,j}$. Output a vector of least nonzero ℓ_p norm among all the vectors $r_{i,j}$ for $i, j \in T$.

First we prove the correctness of the algorithm.

LEMMA 10. *For an appropriate choice of the constant c in the algorithm, it outputs a shortest nonzero vector in $\mathcal{L} \setminus M$ with respect to ℓ_p norm.*

Proof. Let v be a shortest vector in $\mathcal{L} \setminus M$. Consider the set of pairs $P = \{(x_i, z_i) | i \in T\}, T \subset [N]$, that remains after the sieving procedure in Step 3 of the algorithm. If we choose ϵ as a constant in Theorem 5, it follows that there is a constant c such that with probability $1 - 2^{-O(n)}$ there exists $(x_i, z_i), (x_j, z_j) \in P$ such that $v + u = \beta_i - \beta_j$ for some $u \in \mathcal{L} \cap M$ where $\beta_i = z_i - x_i$ and $\beta_j = z_j - x_j$. Hence, in Step 3 of the algorithm we have some $v_{i,j} = v + u$ for some vector $u \in \mathcal{L} \cap M$, i.e. $v_{i,j}$ and v lie in same coset of $\mathcal{L} \cap M$.

Let $w_{i,j} \in \mathcal{L} \cap M$ be a closest vector to $v_{i,j}$. So we have $d(v_{i,j}, w_{i,j}) \leq d(v_{i,j}, u) = \|v\|_p$, i.e. $\|v_{i,j} - w_{i,j}\|_p \leq \|v\|_p$. But since we have $v_{i,j} \notin \mathcal{L} \cap M$ and $w_{i,j} \in \mathcal{L} \cap M$ clearly $v_{i,j} - w_{i,j} \notin$

$\mathcal{L} \cap M$ and since v is a shortest vector in $\mathcal{L} \setminus M$, this implies $\|v_{i,j} - w_{i,j}\|_p = \|v\|_p$. So with probability $1 - 2^{-O(n)}$ the algorithm will output (in Step 4) a vector $r_{i,j}$ with $\|r_{i,j}\|_p = \|v\|_p$. This proves the correctness of the algorithm. ∎

Next we argue that the running time of the algorithm is $2^{O(n+k\log k)} \cdot poly(s)$ where s is the input size. In Step 1 of the algorithm we are sampling $N = 2^{O(n)}$ points from $B_p(0,2)$, a ball of radius 2 with respect to l_p norm. Since $B_p(0,2)$ is a convex body, the task can be accomplished using Dyer-Frieze-Kannan algorithm [DFK91] in time $2^{O(n)} \cdot poly(s)$. It easily follows that the sieving procedure in Step 2 can be performed in $2^{O(n)}$ time. Note that $\mathcal{L} \cap M$ is a rank k lattice and a basis for it can be computed efficiently. We need the following easy lemma from [Mi08].

Lemma 11.[Mi08, Lemma 1] *There is a polynomial-time algorithm that takes as input a lattice $\mathcal{L} \subset \mathbb{Q}^n$ and a subspace $M \subset \mathbb{R}^n$ of dimension $k < n$ outputs a basis for rank k lattice $\mathcal{L} \cap M$.*

From the above lemma it is clear that a basis for $\mathcal{L} \cap M$ can be efficiently computed in polynomial time. In Step 4 of the algorithm we are solving $2^{O(n)}$ many instances of CVP for the rank k lattice $\mathcal{L} \cap M$. For $i, j \in S$ a closest vector to $v_{i,j}$ in the rank k lattice $\mathcal{L} \cap M$ can be computed in $2^{O(k\log k)}$ time using Kannan's algorithm for CVP [Kan87]. Hence the Step 4 takes $2^{O(n+k\log k)}$ time. Therefore the overall running time of the algorithm is $2^{O(n+k\log k)} \cdot poly(s)$. Note that by repeating above algorithm $2^{O(n)}$ times we can make the success probability of the algorithm exponentially close to 1.

Theorem 12. *Given a full rank lattice $\mathcal{L} \subset \mathbb{Q}^n$ and a subspace $M \subseteq \mathbb{R}^n$ of dimension $k < n$, There is a randomized algorithm to finds $v \in \mathcal{L} \setminus M$ with least possible l_p norm. The running time of the algorithm is $2^{O(n+k\log k)}$ times a polynomial in the input size and it succeeds with probability $1 - 2^{-cn}$ for an arbitrary constant c.*

Blömer and Naewe [BN07] gave $2^{O(n)}$ time $1 + \epsilon$ factor approximation algorithms to solve the SMP and SIVP problems. As a simple consequence of Theorem 12 we get a $2^{O(n)}$ time randomized algorithm to "partially" solve SMP: we can compute the first $O(\frac{n}{\log n})$ successive minima in $2^{O(n)}$ time. More precisely, we can compute a set of i linearly independent vectors $\{v_1, v_2, \ldots, v_i\} \subset \mathcal{L}$ such that $\|v_j\|_p = \lambda_j^p(\mathcal{L})$ for $j = 1$ to i if i is $O(\frac{n}{\log n})$.

Given a lattice $\mathcal{L}$, let $M = 0 \subset \mathbb{R}^n$ be the zero-dimensional subspace in $\mathbb{R}^n$ and consider the SAP instance $(\mathcal{L}, M)$. Clearly, v_1 is a shortest vector in $\mathcal{L} \setminus M$. Hence, by Theorem 12 we can compute v_1 in $2^{O(n)}$ time. Now, inductively assume that we have computed linearly independent vectors $v_1, v_2, \ldots, v_k \in \mathcal{L}$ such that $\|v_j\|_p = \lambda_j^p(\mathcal{L})$. Consider the instance $(\mathcal{L}, M)$ of SAP where M is the space generated by $v_1, \ldots, v_k$ and compute $v \in \mathcal{L} \setminus M$ using Theorem 12 in time $2^{O(n+k\log k)}$. It is clear that $\|v\|_p = \lambda_{k+1}^p(\mathcal{L})$ and as $v \notin M$ the vectors $v_1, v_2, \ldots, v_k, v$ are linearly independent. If k is $O(\frac{n}{\log n})$ it is clear that algorithm takes $2^{O(n)}$ time. This proves Corollary 13.

Corollary 13. *Given a full rank lattice $\mathcal{L} \subset \mathbb{Q}^n$ and a positive integer $i \leq \frac{cn}{\log n}$ for a constant c, there is a randomized algorithm with running time $2^{O(n)} \cdot poly(size(\mathcal{L}))$ to* *compute linearly independent vectors $v_1, v_2, \ldots, v_i \in \mathcal{L}$ such that $\|v_j\|_p = \lambda_j^p(\mathcal{L})$ for $j = 1$ to i.*

The CVP problem is polynomial-time reducible to SAP, as noted in [BN07]. Micciancio [Mi08] has shown that CVP, SAP and SMP are all polynomial-time equivalent. Our algorithm computes $v \in \mathcal{L} \setminus M$ with least norm by solving $2^{O(n)}$ instances of CVP. We have basically given a randomized $2^{O(n)}$ time Turing reduction from SAP to CVP. An interesting property of our reduction is that we are solving instance $(\mathcal{L}, M)$ of SAP by solving $2^{O(n)}$ many CVP instances $(\mathcal{L} \cap M, v)$ where $\mathcal{L} \cap M$ is a rank k lattice, where k is dimension of M. In contrast, for the CVP instance (N, v) produced by the SAP to CVP reduction in [BN07] the lattice N has rank $O(n)$.

As a consequence of this property of our reduction we obtain Corollary 14 which states that it suffices to look for a $2^{O(n)}$ randomized exact algorithm for CVP that can access all successive minimas of the input lattice.

COROLLARY 14. *Suppose for all m there is a $2^{O(m)}$ randomized exact algorithm for* CVP *that takes as input a* CVP *instance (M, v) where M is full rank lattice of rank m and $v \in \mathbb{R}^m$ (along with the* extra *input $v_i \in M$ such that $|v_i|_p = \lambda_i^p(M)$ for $i = 1$ to m where $\lambda_i^p(M)$ is i^{th} successive minima in M). Then, in fact, there is a $2^{O(n)}$ randomized exact algorithm for solving* CVP *on any rank n lattice.*

Proof. By [Mi08], CVP is polynomial-time equivalent to SMP (the successive minima problem). Consider the full rank lattice $\mathcal{L} \subset \mathbb{Q}^n$ as input to SMP. It suffices to compute linearly independent vectors $v_1, \ldots, v_n \in \mathcal{L}$ with $\|v_i\|_p = \lambda_i^p(\mathcal{L})$ for $i = 1$ to n in $2^{O(n)}$ time. We proceed as in the proof of Corollary 13. Inductively assume that we have computed linearly independent vectors $v_1, \ldots, v_k \in \mathcal{L}$ with $\|v_i\|_p = \lambda_i^p(\mathcal{L})$. Let M be the space generated by $v_1, \ldots, v_k$. As in proof of Theorem 12 we can solve the SAP instance $(\mathcal{L}, M)$ by solving $2^{O(n)}$ many instances of CVP $(\mathcal{L} \cap M, v')$. Note that $\mathcal{L} \cap M$ is rank k lattice and it is clear that $\|v_i\|_p \lambda_i^p(\mathcal{L} \cap M)$ for $i = 1$ to k. Hence we can solve these instances in $2^{O(n)}$ time (although $\mathcal{L} \cap M$ is not full rank lattice, but it is not difficult to convert all these instances of CVP to full rank by applying a suitable linear transformation). This takes time $2^{O(n+k)}$ which is at most $2^{O(n)}$. Hence, it is clear that we can compute linearly independent vectors $v_1, \ldots, v_n \in \mathcal{L}$ such that $\|v_i\|_p = \lambda_i^p(\mathcal{L})$ in time $n \cdot 2^{O(n)}$. ■

In the next corollary we give a $2^{O(n)}$ time algorithm to solve certain CVP instances $(\mathcal{L}, v)$ for any ℓ_p norm. We prove the result only for ℓ_2 norm and it is easy to generalize it for general ℓ_p norms. Let $\lambda_i(\mathcal{L})$ denote i th successive minima of the lattice $\mathcal{L}$ with respect to ℓ_2 norm.

COROLLARY 15. *Let $(\mathcal{L}, v)$ be a* CVP *instance such that $\mathcal{L}$ is full rank with the promise that $d(v, \mathcal{L}) < \sqrt{3}/2\lambda_t(\mathcal{L})$, $t \leq \frac{cn}{\log n}$. Then there is a $2^{O(n)} \cdot poly(size(\mathcal{L}))$ time randomized algorithm that solves such a* CVP *instance exactly.*

Proof. By Corollary 13 we first compute $\lambda_t(\mathcal{L})$. We now use ideas from Kannan's CVP to SVP reduction [Kan87]. Let $b_1, b_2, \cdots, b_n$ be a basis for $\mathcal{L}$. We obtain new vectors $c_i \in \mathbb{Q}^{n+1}$ for $i = 1$ to n by letting $c_i^T = (b_i^T, 0)$. Likewise, define $u \in \mathbb{Q}^{n+1}$ as $u^T = (v^T, \lambda_t/2)$. Let $\mathcal{M}$ be the lattice generated by the $n+1$ vectors $u, c_1, c_2, \cdots c_n$. Compute the vectors $v_j \in \mathcal{M}$

such that $\|v_j\|_2 = \lambda_j(\mathcal{M})$ for $j = 1$ to t using Corollary 13 in time $2^{O(n)} \cdot poly(size(\mathcal{L}))$. Write vectors v_j as $v_j = u_j + \alpha_j u$, $u_j \in \mathcal{L}(c_1, \cdots, c_n)$ and $\alpha_j \in \mathbb{Z}$. Clearly, $|\alpha_j| \leq 1$ since u has $\lambda_t/2$ as its $(n+1)^{th}$ entry. As $d(v, \mathcal{L}) < \sqrt{3}/2\lambda_t(\mathcal{L})$ we have $d(u, \mathcal{M}) < \lambda_t(\mathcal{L})$. Hence, there is at least one index i, $1 \leq i \leq t$ such that $|\alpha_i| = 1$. Consider the set $S = \{u_i \mid 1 \leq i \leq t, |\alpha_i| = 1\}$and let u_j be the shortest vector in S. Writing $u_j = (w_j^T, 0)$, it is clear that the vector $-w_j \in \mathcal{L}$ is closest vector to v if $\alpha_j = 1$ and w_j is a closest vector to v if $\alpha_j = -1$. ■

References

[AKS01] M. AJTAI, R. KUMAR, D. SIVAKUMAR, A sieve algorithm for the shortest lattice vector. *In Proceedings of the 30th Annual ACM Symposium on Theory of Computing,* 266-275, 2001.

[AKS02] M. AJTAI, R. KUMAR, D. SIVAKUMAR, Sampling short lattice vectors and the closest lattice vector problem. *In Proceedings of the 17th IEEE Annual Conference on Computational Complexity-CCC,* 53-57, 2002.

[Bl00] J. BLÖMER, Closest vectors, successive minima, and dual HKZ-bases of lattices. *In Proceedings of th 17th ICALP,* Lecture Notes in Computer Science 1853, 248-259, Springer, 2000.

[BN07] J. BLÖMER, S. NAEWE Sampling Methods for Shortest Vectors, Closest Vectors and Successive Minima of lattices. *In Proceedings of ICALP,* 65-77, 2007.

[DFK91] M. DYER, A. FRIEZE, R. KANNAN A random polynomial time algorithm for approximating the volume of convex bodies. *Journal of the ACM ,* 38(1):1-17, 1991.

[Kan87] R. KANNAN Minkowski's convex body theorem and integer programing. *Mathematics of Operational Rearch ,*12(3):415-440, 1987.

[LLL82] A. K. LENSTRA, H. W. LENSTRA, JR. AND L. LOVASZ, Factoring Polynomials with Rational Coefficients, *Mathematische Annalen,* 261:515-534, 1982.

[MG02] D. MICCIANCIO, S. GOLDWASSER, *Complexity of Lattice Problems. A Cryptographic Perspective,* Kluwer Academic Publishers, 2002.

[Mi08] D. MICCIANCIO, Efficient reductions among lattice problems,*SODA*,2008,84-93

[Re04] O. REGEV, Lecture Notes — Lattices in Computer Science, lecture 8. Available at the website: http://www.cs.tau.ac.il/ odedr/teaching/lattices_fall_2004/index.html.

[Si45] C. L. SIEGEL Lectures on Geometry of Numbers. *Springer-Verlag publishing company*, 1988.