Sampling Methods for Shortest Vectors, Closest Vectors and Successive Minima

Johannes Blömer* and Stefanie Naewe**

Department of Computer Science, University of Paderborn
{bloemer,naestef}@uni-paderborn.de

Abstract. In this paper we introduce a new lattice problem, the *subspace avoiding problem* (SAP). We describe a probabilistic single exponential time algorithm for SAP for arbitrary ℓ_p norms. We also describe polynomial time reductions for four classical problems from the geometry of numbers, the *shortest vector problem* (SVP), the *closest vector problem* (CVP), the *successive minima problem* (SMP), and the *shortest independent vectors problem (*SIVP*)* to SAP, establishing probabilistic single exponential time algorithms for them. The result generalize and extend previous results of Ajtai, Kumar and Sivakumar. The results on SMP and SIVP are new for all norms. The results on SVP and CVP generalize previous results of Ajtai et al. for the ℓ_2 norm to arbitrary ℓ_p norms.

1 Introduction

In this paper we study four problems from the geometry of numbers, the *shortest vector problem* (SVP), the *closest vector problem* (CVP), the *successive minima problem* (SMP) and the *shortest linearly independent vectors problem* (SIVP).

In the shortest vector problem, we are given a lattice L and are asked to find a (almost) shortest non-zero vector v in the lattice L. In the closest vector problem, we are given a lattice L and some vector t in the $\mathbb{R}$-vector space $\text{span}(L)$ spanned by the vectors in L. We are asked to find a vector $u \in L$, whose distance to t is as small as possible. The problems SMP and SIVP extend SVP and deal with the successive minima $\lambda_k(L)$ of a lattice. Let k be an integer less than or equal to the dimension of $\text{span}(L)$ (called the rank of L). The *k-th successive minimum $\lambda_k(L)$ of L* is the smallest real number r such that L contains k linearly independent vectors of length at most r. In the successive minima problem SMP we are given a lattice L with rank n. We are asked to find n linearly independent vectors $v_1, \ldots, v_n$ such that the length of $v_k, k = 1, \ldots, n$, is at most $\lambda_k(L)$. In SIVP we are asked to find n linearly independent vectors $v_1, \ldots, v_n$ such that the length of v_k is at most $\lambda_n(L)$. Clearly, SIVP is polynomial time reducible to SMP. Since

* This research was supported by Deutsche Forschungsgemeinschaft, grant BL 314/5.
** This research was partially supported by German Science Foundation (DFG), grant BL 314/5, and Research Training Group GK-693 of the Paderborn Institute for Scientific Computation (PaSCo).

L. Arge et al. (Eds.): ICALP 2007, LNCS 4596, pp. 65–77, 2007.

they can be defined for any norm on $\mathbb{R}^n$, we stated these problems without referring to a specific norm.

Algorithms for lattice problems. In the last 25 years the complexity of the lattice problems Svp, Cvp, Smp, and Sivp has been studied intensively. For the history of this problems we refer to [MG02]. It is known that all problems are NP-hard and even hard to approximate (see for example [Ajt98], [Mic00], [Kho05], [DKRS03], [BS99]). Let us briefly review the best algorithms for lattice problems that predate the results by Ajtai et al. The best algorithm to solve Svp optimally was due to Kannan [Kan87b]. Kannan's algorithm has a running time of $n^{n/2}b^{\mathcal{O}(1)}$, where n is the rank of the lattice L and b is its representation size, i.e., the number of bits used to describe the basis defining L. For Cvp the best algorithm that optimally solves the problem was due to [Blö00]. It has a running time of $n!b^{\mathcal{O}(1)}$. Finally, the best deterministic algorithms for Smp and Sivp were also due to [Blö00]. Their running time is $3^b n!b^{\mathcal{O}(1)}$.

Of course, the best deterministic polynomial time algorithms for approximating all four lattice problems are based on the LLL-algorithm (see [LLL82]) and achieve single exponential approximation factors (see for example [Sch94], [Bab86], [Sch87] and [Kan87a]).

The AKS results for Svp *and* Cvp*.* In a breakthrough paper [AKS01] Ajtai, Kumar, and Sivakumar describe a probabilistic algorithm that solves Svp optimally with probability exponentially close to 1. More precisely, the running time of their algorithm is $\left(2^n b\right)^{\mathcal{O}(1)}$, i.e., single exponential only in the rank of the lattice. The AKS-algorithm is based on a novel sampling technique that generates short vectors from the input lattice L. Later, Ajtai, Kumar, and Sivakumar [AKS02] extended their sampling technique to solve Cvp with approximation factor $(1+\epsilon)$ for any $\epsilon > 0$. The running time of their algorithm is $\left(2^{(1+1/\epsilon)n} b\right)^{\mathcal{O}(1)}$.

Our contributions. In this paper, we consider a variant of the AKS-sampling procedure (according to [AKS01] proposed by M. Sudan, described in lecture notes by O. Regev [Reg04]).

- We describe a general sampling procedure to compute short lattice vectors outside some given subspace. We call this the *subspace avoiding problem* (Sap).
- We show polynomial time reductions from exact and approximate versions of Sap to exact and approximate versions of Svp, Cvp, Smp and Sivp.
- In consequence, we obtain single exponential time $(1 + \epsilon)$ approximation algorithms for Svp, Cvp, Smp and Sivp for all ℓ_p norms. The running time is $((2 + 1/\epsilon)^n b)^{\mathcal{O}(1)}$.
- By slightly modifying the sampling procedure and its analysis we are able to solve Sap provided there do not exist too many short lattice vectors outside the given subspace. As a consequence, we obtain single exponential time algorithms for Svp and for restricted versions of Cvp and Sivp.

Organization. The paper is organized as follows. In Section 2 we state the most important facts used in this paper. In Section 3 we formally define the lattice problem SAP and prove polynomial time reductions from SVP, CVP, SMP, and SIVP to SAP. In Section 4, we show that the problem SAP can be approximated with factor $1+\epsilon, \epsilon > 0$ arbitrary, by a sampling procedure. Finally, the modified sampling procedure solving restricted versions of SAP optimally is presented in Section 5.

2 Basic Definitions and Facts

For $m > 0$ is $\mathbb{R}^m$ a m-dimensional vector space over $\mathbb{R}$. The ℓ_p norm of a vector $x \in \mathbb{R}^m$ is defined by $||x||_p = (\sum_{i=1}^m x_i^p)^{1/p}$ for $1 \leq p < \infty$ and $||x||_\infty = \max\{|x_i|, i = 1, \ldots, m\}$ for $p = \infty$. In the sequel we consider the ℓ_p norm for an arbitrary p with $1 \leq p \leq \infty$. We set $B^{(p)}(x, r) := \{y \in \mathbb{R}^m | \ ||y - x||_p < r\}$. The volume $\mathrm{vol}(B^{(p)}(x, r))$ satisfies:

$$\text{For all } c > 0 \quad \mathrm{vol}(B^{(p)}(x, c \cdot r)) = c^m \cdot \mathrm{vol}(B^{(p)}(x, r)). \tag{1}$$

A lattice L is a discrete additive subgroup of $\mathbb{R}^m$. Each lattice L has a basis, i. e. a sequence $b_1, \ldots, b_n$ of n elements of L that generate L as an abelian group. We denote this by $L = \mathcal{L}(b_1, \ldots, b_n)$. We call n the rank of L. If $m = n$, the lattice is full dimensional. In the rest of the paper we only consider full dimensional lattices. However, our results can easily be generalized to arbitrary lattices. For a basis $B = \{b_1, \ldots, b_n\}$ we define the half open parallelepiped $\mathcal{P}(B) := \{\sum_{j=1}^n \alpha_j b_j | 0 \leq \alpha_j < 1, j = 1, \ldots, n\}$. For every vector $v \in \mathbb{R}^n$ there is a unique representation $v = u + w$ with $u \in L$ and $w \in \mathcal{P}(B)$. We write $v \equiv w \mod L$.

We always assume that $L \subseteq \mathbb{Q}^n$. The representation size b of a lattice $L \subseteq \mathbb{Q}^n$ with respect to the basis $\{b_1, \ldots, b_n\}$ is the maximum of n and the binary lengths of the numerators and denominators of the coordinates of the basis vectors b_j. The representation size of a subspace M and the representation size of a vector $u = \sum_{i=1}^n u_i b_i$ with $u_i \in \mathbb{Q}$ with respect to $\{b_1, \ldots, b_n\}$ are defined in the same way. In the sequel, if we speak of the representation size of a lattice L, a subspace M or of a vector $u \in \mathrm{span}(L)$ without referring to some specific basis, we implicitly assume that some basis is given.

3 The Subspace Avoiding Problem SAP, Main Result, and Reductions for SVP, SMP, SIVP and CVP

Definition 1. *Given a lattice L and some subspace $M \subset \mathrm{span}(L)$, we call the problem to compute a vector $v \in L \backslash M$, that is as short as possible, the subspace avoiding problem (SAP). We set*

$$\lambda_M^{(p)}(L) := \min\{r \in \mathbb{R} | \exists v \in L \backslash M, \ ||v||_p \leq r\}.$$

It is not hard to show that for an LLL-reduced basis $\{b_1, \ldots, b_n\}$ we have $||b_k||_2 \leq 2^{n-1} \lambda_M^{(2)}(L)$, where $k = \min\{1 \leq j \leq n | b_j \in L \backslash M\}$. Therefore, we get

Theorem 1. *The LLL-algorithm can be used to approximate in polynomial time* SAP *for the ℓ_2 norm with factor 2^{n-1}.*

In Section 4 we will show that in single exponential time we can approximate SAP with any factor $1+\epsilon$, $0 < \epsilon \leq 2$. More precisely

Theorem 2. *For all ℓ_p norms, $1 \leq p \leq \infty$, there exists a randomized algorithm, that approximates* SAP *with probability exponentially close to 1. The approximation factor is $1+\epsilon$ for any $0 < \epsilon \leq 2$ and the running time of the algorithm is $((2+1/\epsilon)^n \cdot b)^{\mathcal{O}(1)}$, where b is the size of the lattice and the subspace.*

In the remainder of this section we show that there are polynomial time reductions from SVP, CVP, SMP and SIVP to SAP. Together with Theorem 2 this implies single exponential time approximation algorithms for SVP, CVP, SMP and SIVP. The core of the reductions is a suitable definition of the subspace.

For the reduction of SVP to SAP we choose $M := \{0\} \subseteq \text{span}(L)$. If we compute a (almost) shortest non-zero lattice vector $u \in L \backslash M$, we compute a (almost) shortest non-zero lattice vector $u \in L$.

Theorem 3. *For all ℓ_p norms, $1 \leq p \leq \infty$,* SVP *with approximation factor $1+\epsilon, \epsilon \geq 0$, is polynomial time reducible to* SAP *with approximation factor $1+\epsilon$.*

Theorem 4. *For all ℓ_p norms, $1 \leq p \leq \infty$,* SMP *and* SIVP *with approximation factor $1+\epsilon, \epsilon \geq 0$, are polynomial time reducible to* SAP *with approximation factor $1+\epsilon$.*

Proof. We are given access to an oracle $\mathcal{A}$, that solves SAP with an approximation factor $1+\epsilon$ for some arbitrary $\epsilon \geq 0$. Using this oracle we get a $(1+\epsilon)$-approximation of the first successive minimum as in Theorem 3. For $i > 1$ define $M := \text{span}(v_1, \ldots, v_{i-1})$ with $v_1, \ldots, v_{i-1} \in L$ linearly independent. Since $\dim(M) < i$, there exists a vector $w \in L$ with $||w||_p \leq \lambda_i^{(p)}(L)$ and $w \notin M$. Therefore, $\lambda_M^{(p)}(L) \leq \lambda_i^{(p)}(L)$ and using the oracle $\mathcal{A}$ with input L and M we get a $(1+\epsilon)$-approximation for the i-th successive minimum.

The reduction of CVP to SAP relies on a lifting technique introduced by Kannan [Kan87b] and refined by Goldwasser and Micciancio [MG02] and Ajtai, Kumar and Sivakumar [AKS02]. A proof for it will be contained in the full version of this paper.

Theorem 5. *For all ℓ_p norms, $1 \leq p \leq \infty$, the exact version of* CVP *is polynomial time reducible to the exact version of* SAP. *Also, for all ℓ_p norms, $1 \leq p \leq \infty$,* CVP *with approximation factor $(1+\epsilon)(1+\alpha)$ for $0 < \epsilon \leq 1/2$ and $\alpha \geq 0$ is reducible to* SAP *with approximation factor $1+\epsilon/6$. The reduction is polynomial time in the input size of the* CVP *instance and in $1/\alpha$.*

Combining this result with the inapproximability results for CVP due to Dinur et al. [DKRS03] we get the following inapproximability result for SAP.

Theorem 6. *For all ℓ_p norms, $1 \leq p < \infty$, there is some constant $c > 0$, such that* SAP *is* NP*-hard to approximate to within factor $n^{c/\log\log n}$, where n is the dimension of the input lattice.*

4 The Sieving Procedure and the Sampling Procedure

In this section we present a sampling procedure, that solves the subspace avoiding problem with approximation factor $1+\epsilon, 0 < \epsilon \leq 2$. We closely follow Regev's lecture notes on the AKS single exponential algorithm for SVP [Reg04]. First, we show that we can restrict ourselves to instances of SAP with $2 \leq \lambda_M^{(p)}(L) < 3$. A proof for it will be contained in the full version of this paper.

Lemma 1. *For all ℓ_p norms, if there is an algorithm $\mathcal{A}$ that for all lattices L for which $2 \leq \lambda_M^{(p)}(L) < 3$ and all subspaces M solves* SAP *with approximation factor $1+\epsilon$ and in time $T = T(n, b, \epsilon)$, then there is an algorithm $\mathcal{A}'$ that solves* SAP *for all lattices and subspaces M with approximation factor $1+\epsilon$ and in time $\mathcal{O}(nT + n^4 b)$. Here n is the rank of L and b is the representation size of L, M.*

4.1 The Sieving Procedure

The main part of the sampling procedure is a sieving procedure (see Algorithm 2). Its main properties are described in the following lemma. The parameter a is rational and $a > 1$.

Algorithm 2 The sieving procedure
Input: $x_1, \ldots, x_N \in B^{(p)}(0, R)$
  $J \leftarrow \emptyset$
  For $j = 1, \ldots, N$ *do*
    If there exists $i \in J$ *with* $||x_i - x_j||_p \leq R/a$, *then* $\eta(i) \leftarrow j$.
    Else $J \leftarrow J \cup \{i\}$ *and* $\eta(i) \leftarrow i$.

Lemma 2. *Let $R \in \mathbb{R}$, $R > 0$, $a \in \mathbb{Q}$ with $a > 1$. For any set of points $x_1, \ldots, x_N \in B^{(p)}(0, R)$ the sieving procedure 2 finds a subset $J \subseteq \{1, 2, \ldots, N\}$ of size at most $(2a+1)^n$ and a mapping $\eta : \{1, 2, \ldots, N\} \to J$ such that for any $i \in \{1, \ldots, N\}$, $||x_i - x_{\eta(i)}||_p \leq R/a$. The running time of the procedure is $\mathcal{O}(N^2 \cdot poly(m))$, if $x_1, \ldots, x_N$ are rationals of representation size m.*

Proof. Obviously, for all $i \in \{1, \ldots, N\}$, $||x_i - x_{\eta(i)}||_p \leq R/a$. The distance between any two points in J is larger than R/a. If we take balls of radius $R/(2a)$ around each point x_i, $i \in J$, then these balls are disjoint and their union is contained in $B^{(p)}(0, (1 + 1/(2a))R)$. Therefore, the number of balls, and hence $|J|$, is bounded by $\mathrm{vol}\left(B^{(p)}\left(0, (1+\frac{1}{2a})R\right)\right) / \mathrm{vol}\left(B^{(p)}\left(0, \frac{1}{2a}R\right)\right) = (2a+1)^n$ (Equation (1)).

4.2 The Sampling Procedure

Now we present a sampling procedure (see Algorithm 3) that for all ℓ_p norms approximates SAP with the factor $1+\epsilon$, $0 \leq \epsilon \leq 2$ arbitrary. The algorithm chooses N points uniformly at random in a ball $B^{(p)}(0, r)$ with radius r. Using the general algorithm of Dyer, Frieze and Kannan (see [DFK91]) we are able to

sample the N points in $B^{(p)}(0,r)$ with the required accuracy. For the sake of simplicity, we will neglect this aspect in the following. The parameter N will be defined later. For each point x_i with $i \in \{1,\ldots,N\}$ we compute the point $y_i \in \mathcal{P}(B)$ such that $y_i - x_i$ is a lattice point. Using the mapping $\eta : \{1,\ldots,N\} \longrightarrow J$, for each vector y_i we get a representative $y_{\eta(i)}$ with $||y_i - y_{\eta(i)}||_p < R/a$. We replace y_i with $y_i - (y_{\eta(i)} - x_{\eta(i)})$. This procedure is repeated until the distance between the lattice vectors and their representatives is small enough. We use parameters δ, r and a satisfying $0 < \delta \leq 1/2$, $r \geq 1/2$ and $a = 1 + 2/\delta$.

Algorithm 3 The sampling procedure
Input: A lattice $L = \mathcal{L}(B), B = \{b_1,\ldots,b_n\}$, and a subspace $M \subseteq \mathrm{span}(B)$.

1. (a) $R_0 \leftarrow n \cdot \max_i ||b_i||_p$
   (b) *Choose N points $x_1,\ldots,x_N$ uniformly in $B^{(p)}(0,r)$.*
   (c) *Compute $y_i \in \mathcal{P}(B)$ with $y_i \equiv x_i \mod \mathcal{L}(B)$ for $i = 1,\ldots,N$.*
   (d) *Set $\mathcal{Z} \leftarrow \{(x_1,y_1),\ldots,(x_N,y_N)\}$ and $R \leftarrow R_0$.*
2. *While $R > (1+\delta)r$ do*
   (a) *Apply the sieving procedure to $\{y_i | (x_i,y_i) \in \mathcal{Z}\}$ with the parameters a and R. The result is a set J and a mapping η.*
   (b) *Remove from $\mathcal{Z}$ all pairs (x_i,y_i) with $i \in J$.*
   (c) *Replace each remaining pair $(x_i,y_i) \in \mathcal{Z}$ with $(x_i, y_i - (y_{\eta(i)} - x_{\eta(i)}))$.*
   (d) $R \leftarrow R/a + r$

Output: A shortest vector $v \in \{y_i - x_i | (x_i,y_i) \in \mathcal{Z}\}$ with $v \notin M$, if such a vector exists.

Lemma 3. *Let δ and r be chosen as above. Given a lattice $L = \mathcal{L}(B)$, if the sampling procedure 3 returns the vector v, then $v \in L \cap B^{(p)}(0,(2+\delta)r)$.*

The proof follows from the fact that during the sampling procedure the following two properties are satisfied: 1) For all $(x_i,y_i) \in \mathcal{Z}$ we have $y_i - x_i \in \mathcal{L}(B)$ and 2) for all $i \in \{1,\ldots,N\}$ the length of y_i is bounded by the parameter R. For details see [Reg04].

The number of iterations of the while-loop dominates the running time of the sampling procedure.

Lemma 4. *If the sampling procedure 3 is executed with the parameters δ, a and r chosen as above, then the number of iterations of the while-loop is at most $2\log_2(1+2/\delta) \cdot (\log_2 R_0 + \log_2(1+2/\delta))$.*

Proof. After i iterations the parameter R is $R_0/a^i + r\sum_{j=0}^{i-1} a^{-j}$. The loop terminates if $R < (1+\delta)r$. Using the geometric series the loop terminates if $R_0/a^i + r \cdot a/(a-1) \leq (1+\delta)r$. Since $a = 1 + 2/\delta$ and $r \geq 1/2$, the lemma follows.

Using this bound for the number of iterations we can analyze the running time of the sampling procedure.

Lemma 5. *Given a lattice basis B and a subspace $M \subset \mathrm{span}(\mathcal{L}(B))$, with the parameters r, a and δ chosen as above, the running time of the sampling procedure 3 is bounded by $((1+2/\delta)\cdot b\cdot N)^{\mathcal{O}(1)}$. Here b is the size of $\mathcal{L}(B)$ and M. Furthermore, N is the number of points chosen in the sampling procedure.*

Proof. The number of iterations in the while-loop is at most $2\log_2(1+2/\delta)\cdot(\log_2(1+2/\delta)+\log_2 R_0) \leq (1+2/\delta)b^{\mathcal{O}(1)}$. In each iteration we apply the sieving procedure. Since the input size is at most b the running time of the sampling procedure is at most $(1+2/\delta)N^2 b^{\mathcal{O}(1)} = ((1+2/\delta)\cdot b\cdot N)^{\mathcal{O}(1)}$.

Summarizing the previous results about the sampling procedure 3, we get

Theorem 7. *For every $0 < \epsilon \leq 2$ there exists a $\delta > 0$ such that the following holds: Given a lattice $L = \mathcal{L}(B)$, a subspace M, and r satisfying $1/2 \leq r \leq (1/2)\cdot(1+\delta)^2\lambda_M^{(p)}(L)$, the sampling procedure 3 computes a set of vectors from $L \cap B^{(p)}(0,(1+\epsilon)\lambda_M^{(p)}(L))$. The running time of the sampling procedure is $((2+1/\epsilon)^n\cdot b)^{\mathcal{O}(1)}$, where b is the size of L and M.*

The proof is obviously if we choose $\delta = (1/4)\epsilon$ and combine this with the results of Lemma 3 and Lemma 5.

Also, using Lemma 2 and Lemma 4 we get

Lemma 6. *If we apply the sampling procedure 3 with the parameters δ, a and r chosen as above, we remove at most*

$$z(R_0,\delta) := (\log_2 R_0 + \log_2(1+2/\delta))(2(1+2/\delta)+1)^{n+1} \tag{2}$$

pairs from the set $\mathcal{Z}$.

4.3 Modification of the Sampling Procedure

We need to show that the sampling procedure 3 computes vectors in $L\backslash M$. For this we use the randomization in the algorithm. We change our point of view and consider a modified sampling procedure that behaves exactly like the sampling procedure 3. We are able to show that the modified sampling procedure computes with probability exponentially close to 1 a vector $v \in L\backslash M$. Hence, the same is true for the sampling procedure 3.

Let $u \in L\backslash M$ a lattice vector with $||u||_p = \lambda_M^{(p)}(L)$. Define

$$C_1 := B^{(p)}(0,r) \cap B^{(p)}(u,r) \text{ and } C_2 := B^{(p)}(0,r) \cap B^{(p)}(-u,r).$$

If the parameter r satisfies

$$\frac{1}{2}(1+\delta)\lambda_M^{(p)}(L) \leq r \leq \frac{1}{2}(1+\delta)^2\lambda_M^{(p)}(L) \tag{3}$$

for a $\delta > 0$, the sets C_1 and C_2 are non-empty and disjoint. We define a bijective mapping $\tau_u : B^{(p)}(0,r) \to B^{(p)}(0,r)$ depending on the lattice vector u.

$$\tau_u(x) = \begin{cases} x+u\ , & x \in C_2 \\ x-u\ , & x \in C_1 \\ x\quad , & \text{otherwise} \end{cases}$$

Using the mapping τ_u we define the modified sampling procedure (see Algorithm 4). Since the modified sampling procedure is only used for the analysis, we do not worry about its running time and the fact that it uses the unknown u. The sampling procedure 3 and the modified sampling procedure 4 return vectors in $L \cap B^{(p)}(0, (1+\epsilon)\lambda_M^{(p)}(L))$ distributed according to certain distributions. We call these the *output distributions* generated by the sampling procedure and the modified sampling procedure, respectively. Next, we show

Algorithm 4 The modified sampling procedure
Input: A lattice $L = \mathcal{L}(B), B = \{b_1, \ldots, b_n\}$, *and a subspace* $M \subseteq \mathrm{span}(B)$

1. (a) $R_0 \leftarrow n \cdot \max_i ||b_i||_p$.
   (b) *Choose* N *points* $x_1, \ldots, x_N$ *uniformly in* $B^{(p)}(0, r)$.
   (c) *Compute* $y_i \in \mathcal{P}(B)$ *with* $y_i \equiv x_i \mod \mathcal{L}(B)$ *for* $i = 1, \ldots, N$.
   (d) *Set* $\mathcal{Z} \leftarrow \{(x_1, y_1), \ldots, (x_N, y_N)\}$ *and* $R \leftarrow R_0$.
2. *While* $R > (1+\delta)r$ *do*
   (a) *Apply the sieving procedure 2 to* $\{y_i | (x_i, y_i) \in \mathcal{Z}\}$ *with the parameters a and R. The result is a set* J *and a mapping* η.
   (b) *Remove from* $\mathcal{Z}$ *all pairs* (x_i, y_i) *with* $i \in J$.
   (c) *For each pair* $(x_i, y_i), i \in J$, *replace* x_i *with* $\tau_u(x_i)$ *with probability* $1/2$.
   (d) *Replace each remaining pair* $(x_i, y_i) \in \mathcal{Z}$ *with* $(x_i, y_i - (y_{\eta(i)} - x_{\eta(i)}))$.
   (e) $R \leftarrow \frac{R}{a} + r$
3. *For each pair* $(x_i, y_i) \in \mathcal{Z}$ *replace* x_i *with* $\tau_u(x_i)$ *with probability* $1/2$.

Output: A shortest vector $v \in \{y_i - x_i | (x_i, y_i) \in \mathcal{Z}\}$ *with* $v \notin M$, *if such a vector exists.*

Theorem 8. *The sampling procedure 3 and the modified sampling procedure 4 generate the same output distribution.*

Proof. We consider a series of modifications to the sampling procedure 3 leading to the modified sampling procedure 4. In the first modification, after choosing in step 1b the points x_i we decide for each x_i uniformly at random whether to keep x_i or to replace it with $\tau_u(x_i)$. Since τ_u is bijective, this does not change the distribution on the points x_i. Hence, this modification does not change the output distribution of the sampling procedure. Next, observe that $u \in L$ implies $y_i \equiv x_i \equiv \tau_u(x_i) \mod L, i = 1, \ldots, N$. Hence, if we decide for each x_i whether to replace it with $\tau_u(x_i)$ at the end of step 1 rather than in step 1b, then this does not change the output distribution.

But if, without changing the output distribution, we can choose for each x_i whether to keep it or to replace it with $\tau_u(x_i)$ at the end of step 1, then making that decision for each x_i prior to the first time it is used in step 2 will also not change the output distribution. Furthermore, for each point x_i not used at all in step 2 we can choose whether to keep it or replace it with $\tau_u(x_i)$ at the end of step 2. But this is exactly the modification leading from the sampling procedure 3 to the modified sampling procedure 4.

For the further analysis only pairs (x_i, y_i) with $x_i \in C_1 \cup C_2$ are of interest because only for them the mapping τ_u is not the identity. In the following, three lemmata we will show that with high probability at the end of the sampling procedure 3 or the modified sampling procedure 4 the set $\mathcal{Z}$ contains at least 2^n pairs with this property. All lemmata are stated without proof. First, we need the probability, that a point x, which is chosen uniformly in $B^{(p)}(0,r)$, is contained in $C_1 \cup C_2$.

Lemma 7. *Let $u \in \mathbb{R}^n$ be a vector with $||u||_p = \rho$ and $\zeta > 0$. Define $C = B^{(p)}(0, (1/2)(1+\zeta)\rho) \cap B^{(p)}(u, (1/2)(1+\zeta)\rho)$. Then*

$$\frac{\mathrm{vol}(C)}{\mathrm{vol}\left(B^{(p)}\left(0, \frac{1}{2}(1+\zeta)\rho\right)\right)} \geq 2^{-n}\left(\frac{\zeta}{1+\zeta}\right)^n.$$

Next, we are interested in the number of points x_i, which are contained in $C_1 \cup C_2$, if we choose N points uniformly at random in $B^{(p)}(0,r)$.

Lemma 8. *Let $N \in \mathbb{N}$. By q denote the probability that a random point in $B^{(p)}(0,r)$ is contained in $C_1 \cup C_2$. If N points $x_1, \ldots, x_N$ are chosen uniformly at random in $B^{(p)}(0,r)$, then with probability larger than $1 - 4/(N \cdot q)$, there are at least $(q \cdot N)/2$ points $x_i \in \{x_1, \ldots, x_N\}$ with the property $x_i \in C_1 \cup C_2$.*

From the Lemmata 7 and 8 combined with Lemma 6 we obtain

Lemma 9. *Let L be a lattice and $M \subset \mathrm{span}(L)$ be a subspace. Furthermore, assume that in the first step of the sampling procedure 3 or of the modified sampling procedure 4 the number of points chosen is $N = ((1+\delta)/\delta)^n 2^{n+1}(2^n + z(R_0, \delta))$, where $z(R_0, \delta)$ is defined as in (2). Then at the end of step 2 of the sampling procedure 3 or the modified sampling procedure 4 the set $\mathcal{Z}$ contains with probability exponentially close to 1 at least 2^n pairs (x, y) with the property $x \in C_1 \cup C_2$.*

Theorem 9. *For every $0 < \epsilon \leq 2$ there exists a $\delta > 0$ such that the following holds: Given a lattice $L = \mathcal{L}(B)$, a subspace M of $\mathrm{span}(L)$, for which $2 \leq \lambda_M^{(p)}(L)$ and r satisfying (3), the modified sampling procedure 4 computes with probability exponentially close to 1 a vector $v \in L \backslash M$.*

Proof. We apply the modified sampling procedure with the same parameter as in Theorem 7, i. e. $\delta = (1/4)\epsilon$. Since $2 \leq \lambda_M^{(p)}(L)$, we have $r \geq 1/2$. By assumption $u \in L \backslash M$. If $y - x \in M$, then $y - \tau_u(x) = y - x \pm u \in L \backslash M$. The modified sampling procedure returns a vector $v \in L \backslash M$, if at the end of step 2 there exists a pair $(x, y) \in \mathcal{Z}$ with $x \in C_1 \cup C_2$ and one of the following conditions holds: $y - x \in M$ and in step 3 we replace x with $\tau_u(x)$ or $y - x \in L \backslash M$ and in step 3 we do not replace x with $\tau_u(x)$. In step 3 of the modified sampling procedure we decide for each pair $(x, y) \in \mathcal{Z}$ uniformly if we replace it or not. Using Lemma 9 the set $\mathcal{Z}$ contains with probability exponentially close to 1 at least 2^n pairs (x, y) with the property $x \in C_1 \cup C_2$. Therefore the probability, that the modified sampling procedure does not return a vector $v \in L \backslash M$, is bounded by 2^{-2^n}.

By Theorem 8 the sampling procedure and the modified sampling procedure generate the same output distribution. Also, we have shown that we can restrict ourselves to instances of SAP with $2 \leq \lambda_M^{(p)}(L) < 3$ (Lemma 1). Hence, we get

Theorem 10. *There exists a randomized algorithm that for all ℓ_p norms, $1 \leq p \leq \infty$, solves* SAP *with approximation factor $1+\epsilon$, $0 < \epsilon \leq 2$ arbitrary, with probability exponentially close to 1. The running time of the algorithm is $((2+1/\epsilon)^n \cdot b)^{\mathcal{O}(1)}$, where b is the size of the input lattice and the subspace.*

Proof. Let $\delta = (1/4)\epsilon$. Using Lemma 1 we can assume: $2/3 < 2/\lambda_M^{(p)}(L) \leq 1$. Let $\kappa_0 = \log_{1+\delta}(2/3)$ and $\kappa_1 = 0$. Set $l := \lceil \log_{1+\delta} 2/\lambda_M^{(p)}(L) \rceil$, then $\kappa_0 \leq l \leq \kappa_1$ and $r := (1+\delta)^{2-l}$ satisfies the Equation (3), i.e., $(1/2)(1+\delta)\lambda_M^{(p)}(L) \leq r \leq (1/2)(1+\delta)^2\lambda_M^{(p)}(L)$. We apply the sampling procedure for each value $r = (1+\delta)^{2-l'}$ with $\kappa_0 \leq l' \leq \kappa_1$. Let $v_{l'} \in L \backslash M$ be the lattice point discovered by the sampling procedure started with $r = (1+\delta)^{2-l'}$, if any lattice point is discovered. The output will be the smallest $v_{l'} \in L \backslash M$. As we have seen, for the unique $l' = l$ such that $r = (1+\delta)^{2-l'}$ satisfies the Equation (3) the sampling procedure will find a $(1+\epsilon)$-approximation for SAP with probability exponentially close to 1.

We apply the sampling procedure $\left|\log_{1+\delta}(2/3)\right|$ times. By our choose of $\delta = (1/4)\epsilon$ the running time is $\left|\log_{1+\epsilon} 2/3\right| ((2+1/\epsilon) \cdot b)^{\mathcal{O}(1)} = ((2+1/\epsilon)^n \cdot b)^{\mathcal{O}(1)}$.

Combining this with the results from Section 3 we obtain:

Theorem 11. *There exist randomized algorithms that for all ℓ_p norms, $1 \leq p \leq \infty$, approximate* SVP, SMP, SIVP, *and* CVP *with probability exponentially close to 1. In case of* SVP, SMP, *and* SIVP *the approximation factor is $1+\epsilon$ for any $0 < \epsilon \leq 2$. For* CVP *the approximation factor is $1+\epsilon$ for any $0 < \epsilon < 1/2$. The running time of the algorithms is $((2+1/\epsilon)^n \cdot b)^{\mathcal{O}(1)}$, where b is the size of the input lattice and the subspace.*

5 Using the Sampling Procedure for Optimal Solutions

Theorem 12. *Let $L = \mathcal{L}(B)$ be a lattice and M be a subspace of* span(L)*, both of size b. Assume that there exist absolute constants c, ϵ such that the number of $v \in L \backslash M$ satisfying $||v||_p \leq (1+\epsilon)\lambda_M^{(p)}(L)$ is bounded by 2^{cn}. Then there exists an algorithm that solves* SAP *with probability exponentially close to 1. The running time is $(2^n \cdot b)^{\mathcal{O}(1)}$.*

Proof. To turn the $(1+\epsilon)$-sampling procedure into an exact algorithm, we use the sampling procedure 3 with the parameters $\delta = (1/4)\epsilon$ and $N = ((1+\delta)/\delta)^n 2^{n+1}\left(5 \cdot 2^{(c+1)n} + z(R_0, \delta)\right)$, where $z(R_0, \delta)$ is defined in (2). We only modify the output: We consider the two sets

$$O_1 := \{(y_i - x_i) - (y_j - x_j) | (x_i, y_i), (x_j, y_j) \in \mathcal{Z}\} \text{ and } O_2 := \{y_i - x_i | (x_i, y_i) \in \mathcal{Z}\}.$$

The output is a shortest lattice vector $v \in O_1 \cup O_2$ with $v \in L \backslash M$. The analysis and the running time of this sampling procedure are the same as in Section 4.

Obviously, we can modify the sampling procedure in the same way as in Theorem 8 by using the mapping τ_u with respect to a shortest vector $u \in L \backslash M$. We obtain a modified sampling procedure similar to procedure 4 which generates the same output distribution as the original sampling procedure. Hence, we only need to analyze the success probability of the modified sampling procedure. We show that the modified sampling procedure computes with probability exponentially close to 1 the lattice vector u. In the following, consider the set $\mathcal{Z}$ after step 2 and before step 3 of the modified sampling procedure. We define the multiset $F := \{(x_i, y_i) \in \mathcal{Z} | x_i \in C_1\} \subseteq \mathcal{Z}$ and for $v \in L$ we set $F_v := \{(x_i, y_i) \in F | y_i - x_i = v\}$. As in Lemma 9, we can show, that F contains with probability exponentially close to 1 at least $5 \cdot 2^{(c+1)n}$ pairs. Next, we consider two cases: 1) There exists an $v \in L$ with $|F_v| \geq 2^n$ and 2) $|F_v| < 2^n$ for all $v \in L$.

In the first case, in step 3 we decide for each pair $(x, y) \in F_v$ uniformly whether we replace x with $\tau_u(x)$ or not. If there exist $(x_i, y_i), (x_j, y_j) \in F_v$ such that in step 3 the mapping τ is applied to x_i but not to x_j then $u \in O_1$. This event happens with probability $1 - 2 \cdot 2^{-2^n}$.

In the second case, we show that with probability exponentially close to 1 the vector u is contained in the set O_2. We do this by showing that after step 3 of the modified sampling procedure all vectors $v \in L \backslash M$ satisfying $||v||_p \leq (1+\epsilon)\lambda_M^{(p)}(L)$ are contained in O_2. In the following, we consider $\mathcal{F} := \{v \in L | \exists (x, y) \in F \text{ with } v = y - x\}$. Since $|F| > 5 \cdot 2^{(c+1)n}$ and $|F_v| < 2^n$ for all $v \in L$, we obtain $|\mathcal{F}| \geq 5 \cdot 2^{cn}$. Let $\mathcal{F}_1 := \mathcal{F} \cap M$. By assumption $|\mathcal{F} \backslash \mathcal{F}_1| \leq 2^{cn}$ and therefore $|\mathcal{F}_1| = |\mathcal{F}| - |\mathcal{F} \backslash \mathcal{F}_1| \geq 2^{cn+2}$. For all $v = y - x \in \mathcal{F}_1$ we have $y - \tau_u(x) \in L \backslash M$. Analogously to Lemma 8, we can show that with probability exponentially close to 1, for at least 2^{cn} elements $v = y - x$ in $\mathcal{F}_1$ we replace in step 3 the element x by $\tau_u(x)$. Hence, with probability exponentially close to 1 we get 2^{cn} elements $y - x - u \in L \backslash M$. All these elements have length at most $(1+\epsilon)\lambda_M^{(p)}(L)$. Combining this with $|(L \backslash M) \cap B^{(p)}(0, (1+\epsilon)\lambda_M^{(p)}(L))| < 2^{cn}$ we see that in this case the set O_2 contains all vectors $v \in L \backslash M$ of length at most $(1+\epsilon)\lambda_M^{(p)}(L)$.

To use the exact sampling procedure to solve Svp, Cvp, Smp and Sivp we need the following lemma, whose proof is almost identical to the proof of Lemma 2.

Lemma 10. *Let L be a lattice and $R > 0$. Then*

$$|B^{(p)}(0, R) \cap L| < \left(\left(2R + \lambda_1^{(p)}(L) \right) / \lambda_1^{(p)}(L) \right)^n .$$

In case of Svp we use Lemma 10 with $R = (1+\epsilon)\lambda_1^{(p)}(L)$ and get $|B^{(p)}(0, (1+\epsilon)\lambda_1^{(p)}(L)) \cap L| \leq (3+2\epsilon)^n = 2^{cn}$ for a $c \in \mathbb{N}$. Therefore, the assumptions of Theorem 12 are satisfied in case of Svp and we obtain the following.

Theorem 13. *Let $L \subset \mathbb{Q}^n$ be a lattice of size b. A shortest non-zero vector in L can be computed with probability exponentially close to 1. The running time is $(2^n \cdot b)^{\mathcal{O}(1)}$.*

Using Lemma 10, for SMP and CVP we can only show that the number of almost optimal solutions to SMP or CVP is single exponential in the rank of L if the n-th successive minimum $\lambda_n^{(p)}(L)$ or the distance D_t of target vector t to lattice L are bounded by $c\lambda_1^{(p)}(L)$ for some constant c. Hence, we get

Theorem 14. *Let $L \subset \mathbb{Q}^n$ be a lattice of size b. Assume that the n-th successive minimum $\lambda_n^{(p)}$ is bounded by $c\lambda_1^{(p)}$ for some constant $c \in \mathbb{N}$. Then the successive minima of L can be computed with probability exponentially close to 1. The running time is $(2^n \cdot b)^{\mathcal{O}(1)}$.*

Theorem 15. *Let $c > 0$ be some constant. Assume lattice $L \in \mathbb{Q}^n$ and target vector $t \in \mathrm{span}(L)$ are of size b. Assume furthermore, that $D_t \leq c\lambda_1^{(p)}(L)$. Then a vector $v \in L$ satisfying $||t - v||_p = D_t$ can be computed with probability exponentially close to 1. The running time is $(2^n \cdot b)^{\mathcal{O}(1)}$.*

Acknowledgment. We thank O. Regev for several stimulating discussions that greatly benefited the paper. Moreover, his lecture notes on the Ajtai, Kumar, Sivakumar algorithm for SVP [Reg04] were the starting point for our research.

References

[Ajt98] Ajtai, M.: The shortest vector problem in l_2 is NP-hard for randomized reductions. In: Proceedings of the 30th ACM Symposium on Theory of Computing, pp. 10–19. ACM Press, New York (1998)

[AKS01] Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proceedings of the 33th ACM Symposium on Theory of Computing, pp. 601–610. ACM Press, New York (2001)

[AKS02] Ajtai, M., Kumar, R., Sivakumar, D.: Sampling short lattice vectors and the closest lattice vector problem. In: Proceedings of the 17th IEEE Annual Conference on Computational Complexity - CCC, pp. 53–57. IEEE Computer Society Press, Los Alamitos (2002)

[Bab86] Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica 6(1), 1–13 (1986)

[Blö00] Blömer, J.: Closest vectors, successive minima, and dual HKZ-bases of lattices. In: Welzl, E., Montanari, U., Rolim, J.D.P. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 248–259. Springer, Heidelberg (2000)

[BS99] Blömer, J., Seifert, J.-P.: The complexity of computing short linearly independent vectors and sort bases in a lattice. In: Proceedings of the 21th Symposium on Theory of Computing, pp. 711–720 (1999)

[DFK91] Dyer, M., Frieze, A., Kannan, R.: A random polynomial time algorithm for approximating the volume of convex bodies. Journal of the ACM 38(1), 1–17 (1991)

[DKRS03] Dinur, I., Kindler, G., Raz, R., Safra, S.: Approximating CVP to within almost-polynomial factors in NP-hard. Combinatorica 23(2), 205–243 (2003)

[Kan87a] Kannan, R.: Algorithmic geometry of numbers. Annual Reviews in Computer Science 2, 231–267 (1987)

[Kan87b] Kannan, R.: Minkowski's convex body theorem and integer programming. Mathematics of Operations Research 12(3), 415–440 (1987)

[Kho05] Khot, S.: Hardness of approximating the shortest vector problem in lattices. Journal of the ACM (JACM) 52(5), 789–808 (2005)

[LLL82] Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261, 515–534 (1982)

[MG02] Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems - A Cryptographic Perspective. Kluwer Academic Publishers, Dordrecht (2002)

[Mic00] Micciancio, D.: The shortest vector in a lattice is hard to approximate to within some constant. SIAM Journal on Computing 30(6), 2008–2035 (2000)

[Reg04] Regev, O.: Lecture note on lattices in computer science, lecture 8: $2^{O(n)}$-time algorithm for SVP (2004)

[Sch87] Schnorr, C.-P.: A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science 53, 201–224 (1987)

[Sch94] Schnorr, C.-P.: Block reduced lattice bases and successive minima. Combinatorics, Probability & Computing 3, 507–522 (1994)