The Shortest Vector Problem in Lattices with Many Cycles

Mårten Trolin

Department of Numerical Analysis and Computer Science,
Royal Institute of Technology, Stockholm, Sweden
marten@nada.kth.se

Abstract In this paper we investigate how the complexity of the shortest vector problem in a lattice Λ depends on the cycle structure of the additive group $\mathbb{Z}^n/\Lambda$. We give a proof that the shortest vector problem is **NP**-complete in the max-norm for n-dimensional lattices Λ where $\mathbb{Z}^n/\Lambda$ has $n-1$ cycles. We also give experimental data that show that the LLL algorithm does not perform significantly better on lattices with a high number of cycles.

Keywords: lattices, LLL algorithm, shortest vector problem

1 Introduction

Lattices were examined already in the middle of the 19th century, at that time mostly because of their connections to quadratic forms. The interest in algorithmic aspects of lattice problems started in the beginning of the 1980s.

In 1981, van Emde Boas [16] showed that finding the lattice point closest to a given point is **NP**-hard in ℓ_r-norm for any $r > 0$. In the following year, Lenstra, Lenstra and Lovász [7] published their lattice basis reduction algorithm, which is guaranteed to find a vector not more than an exponential factor longer than the shortest vector in polynomial time. This was a great achievement. Schnorr has improved approximation to a slightly sub-exponential factor [12].

The **NP**-hardness of the shortest vector problem in Euclidean norm was an open problem for a long time. It was proven to be **NP**-hard under randomized reductions by Ajtai in 1998 [2]. This result has been improved by several authors, and the strongest result today by Micciancio [8] is that the shortest vector problem is **NP**-hard to approximate within any factor smaller than $\sqrt{2}$ under randomized reductions. On the other hand, Goldreich and Goldwasser [4] have showed that this **NP**-hardness result cannot be extended to $\sqrt{n}$ unless the polynomial-time hierarchy collapses.

A lattice can be described either by a basis that spans the lattice, or as the solutions $\mathbf{x}$ of a set of modular equations $\langle \mathbf{a}_i, \mathbf{x} \rangle = 0 \pmod{k_i}$. Lattices can be classified after the cycle structure of the subgroup $\mathbb{Z}^n/\Lambda$. The number of cycles and the lengths of the cycles in this subgroup corresponds to the minimum number of equations and the moduli of the equations necessary to describe the lattice. Our main focus will be to investigate whether there is a difference in the complexity of computing short vectors between lattices with different number of cycles.

In 1996, Ajtai [1] published a paper in which it is showed that a random lattice from a certain set of lattices is at least as hard as a certain shortest vector problem in the worst case. This result has been improved in [3]. These n-dimensional lattices have n/c cycles, whereas a random lattice usually has one cycle. On the other hand, Schnorr and Paz [11] have showed in a worst-case result that any lattice can be approximated arbitrarily well by a lattice with one cycle. Schnorr's and Paz's result indicates that the lattices with one cycles are the hardest, whereas Ajtai's result gives evidence that also lattices with n/c cycles are hard (although the problem studied by Ajtai is of a different kind than the common shortest vector problem). This gives rise to the question on whether or not lattices with more cycles are easier, or whether the number of cycles is of no importance to the shortest vector problem.

As far as we know, it has never previously been investigated how the cycle structure affects the complexity of lattice problems. Except for the lattice created by Ajtai [1], the previously published reductions that we know about [2,8,16] contain no analysis of the cycle structure of the lattices used. We will show that even for a large number of cycles the shortest vector problem is hard in the max-norm. We also give experimental data that indicate that the LLL lattice basis reduction algorithm does not perform significantly better when applied on lattices with a high number of cycles. However, we still lack a theoretical explanation for these results.

2 Definitions

A *lattice* Λ is the set $\{\sum_{i=1}^{n} \lambda_i \mathbf{b}_i \mid \lambda_i \in \mathbb{Z}\}$ where the vectors $\mathbf{b}_i \in \mathbb{Z}^n$ are linearly independent. The vectors $\mathbf{b}_i$'s are called a *basis* of the lattice, and the matrix $\mathbf{B}$ with the vectors $\mathbf{b}_i$'s as its rows is called a *basis matrix* for Λ. By $\Lambda(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ we mean the lattice spanned by the basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$. The determinant of a lattice is defined as $\det(\Lambda) = |\det(\mathbf{B})|$. The *$\ell_k$-norm* of a vector $\mathbf{v}$ is defined as $\|\mathbf{v}\|_k = \left(\sum_{i=1}^{n} |v_i|^k\right)^{1/k}$. We also define the *max-norm*, ℓ_∞-norm, as $\|\mathbf{v}\|_\infty = \max_{i=1}^{n} |v_i|$. We can see that the ℓ_2-norm is the Euclidean norm. In this report we will mainly consider the ℓ_2-norm and the ℓ_∞-norm. When we leave out the index we mean the ℓ_2-norm.

A vector $\mathbf{v} \in \Lambda$ is called a *shortest lattice vector* if $\|\mathbf{v}\| > 0$ and for every vector $\mathbf{u} \in \Lambda$ either $\|\mathbf{u}\| \geq \|\mathbf{v}\|$ or $\|\mathbf{u}\| = 0$. Moreover we define the length of a shortest vector in Λ as $\lambda(\Lambda) = \|\mathbf{v}\|$. We define the length of a basis as the length of the longest vector in the basis.

In the end of the 19th century, Minkowski [10] proved an upper bound on the length of the shortest vector in a lattice.

Theorem 1 (Minkowski's inequality). *Let $\Lambda \in \mathbb{Z}^n$ be an n-dimensional lattice. Then*

$$\lambda(\Lambda) \leq \sqrt{\gamma_n}(\det(\Lambda))^{1/n}$$

where γ_n is a constant.

The least constant γ_n is called Hermite's constant of rank n, and it has been proved that $\gamma_n \leq \frac{n}{\pi e}$ [5]. It is also known that $\gamma_n \geq \frac{n}{2\pi e}$.

An alternative way to describe a lattice is by giving a set of modular equation, that is, equations of the form $\langle \mathbf{a}_i, \mathbf{x} \rangle = 0 \pmod{k_i}$, where $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ are n-dimensional vectors. Any lattice can be written on this form. We will say that a lattice described by m modular equations with moduli $k_1, k_2, \ldots, k_m$ has m cycles of lengths $k_1, k_2, \ldots, k_m$, provided that the equations are on as simple form as possible. More exactly, we demand that the coefficients and the modulus are relative prime within each equation, and that $k_i | k_{i+1}$, $i = 1, 2, \ldots, m-1$. If we construct a lattice by modular equations such that the moduli do not have this property, we can always combine the equations to become equations of this form, and this representation is easy to compute.

The Smith normal form of a matrix [14] gives us the relation that the lengths of the cycles of a lattice is given by the *determinant divisors* of the basis matrix:

Theorem 2. *Let Λ be a lattice and $\mathbf{B}$ its basis matrix. Then the lengths of the cycles of Λ, $k_1, k_2, \ldots, k_n$ are given by*

$$k_i = \frac{d_i}{d_{i-1}}$$

where d_i is gcd *of all i-minors of $\mathbf{B}$ and $d_0 = 1$.*

3 Background and previous results

3.1 Complexity of finding short vectors

To the best of our knowledge, the first result of **NP**-hardness of calculating short vectors in a lattice was published by van Emde Boas in 1981 [16], where it is proved **NP**-hard to calculate the shortest vector in ℓ_∞-norm in a general lattice. The same problem for the ℓ_2-norm was long an open problem, until proven **NP**-hard under randomized reductions by Ajtai in 1998 [2]. Micciancio [8] improved this result by showing that it is **NP**-hard to approximate the shortest vector within a factor $\sqrt{2} - \varepsilon$ for any $\varepsilon > 0$ under randomized reductions.

3.2 On the cycle structure of lattices

We will now state a few theorems on the cycle structure. These are probably well known, and we will therefore omit the proofs. Please note that some of the cycle lengths k_i mentioned in the theorems may be 1.

Theorem 3. *Let $\Lambda \subseteq \mathbb{Z}^n$ be a lattice with cycle structure $\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \cdots \times \mathbb{Z}_{k_n}$. Then the lattice $d \cdot \Lambda$ has the cycle structure $\mathbb{Z}_{d \cdot k_1} \times \mathbb{Z}_{d \cdot k_2} \times \cdots \times \mathbb{Z}_{d \cdot k_n}$.*

The next theorem shows that we can always assume that the shortest cycle of a lattice has length 1.

Theorem 4. *Let $\Lambda \subseteq \mathbb{Z}^n$ be a lattice with cycle structure $\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2} \times \cdots \times \mathbb{Z}_{k_n}$, where $k_1 \leq k_i$, $i = 2, \ldots, n$. Then $\Lambda = k_1 \cdot \Lambda'$, where Λ' is a lattice with cycle structure $\mathbb{Z}_{k_2/k_1} \times \mathbb{Z}_{k_3/k_1} \times \cdots \times \mathbb{Z}_{k_n/k_1}$.*

3.3 Previous results on the cycle structure

Paz and Schnorr [11] have showed the following theorem, which essentially says that any lattice can be approximated arbitrarily well by a lattice described by a single modular equation. We will call these lattices *cyclic*.

Theorem 5. *Let $\Lambda \in \mathbb{Z}^n$ be a lattice. Then for every $\varepsilon > 0$ we can efficiently construct a linear transformation $\sigma_{\Lambda,\varepsilon} : \Lambda \to \mathbb{Z}^n$ such that $\sigma_{\Lambda,\varepsilon}(\Lambda)$ is a lattice and for some integer k*

1. $\forall \mathbf{u} \in \Lambda : \|\mathbf{u} - \sigma_{\Lambda,\varepsilon}(\mathbf{u})/k\| \leq \varepsilon \|\mathbf{u}\|$
2. *$\sigma_{\Lambda,\varepsilon}(\Lambda)$ has one cycle.*

This theorem implies that the cyclic lattices, in some sense, are the hardest ones in the worst case. If we know a way of finding short vectors in cyclic lattices, this would give us a method of finding short vectors in any lattice.

The average case/worst case connection described by Ajtai [1], the class of hard lattices consist of lattices with n/c cycles, where n is the dimension and c some constant.

These first results show that lattices with just one cycle are hard, but the latter seems to indicate that also lattices with relatively many cycles are hard. Hence it is natural to investigate the role of the cycle structure in complexity questions.

4 The LLL algorithm in practice

In this section we will give data about the performance of the LLL algorithm in practice when applied to lattices with different number of cycles. The intention is to find out whether or not the result of LLL depends on the cycle structure of the lattice.

In all experiments, version 4.3 of the NTL library [13] was used.

4.1 Construction of lattice instances

For the experiments, we need to construct lattices in such a way that we have control over the number of cycles in the lattices. The idea is to create a set of linear modular equations and compute the null space of this set of equations.

To create an n-dimensional lattice with m cycles we create an $m \times n$ matrix $\mathbf{A}$ and set

$$\Lambda = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \equiv \mathbf{0} \pmod{q}\} .$$

The elements of $\mathbf{A}$ are given by a shift register generator. In this register we create a stream of bits using

$$x_{i+1} = \sum_{j=1}^{l} a_j x_{i-j} \bmod 2 \ .$$

The parameter $\mathbf{a} = (a_1, a_2, \ldots, a_l)$ is a constant to be chosen. Solving for the null space gives us $n - m$ basis vectors. To ensure the basis contains n vectors, m rows of the matrix $q\mathbf{I}_n$ are added to the basis.

The dimensions of $\mathbf{A}$ determine the cycle structure of the lattice. With m rows in $\mathbf{A}$, we get a lattice with m cycles of length q. To make it possible to compare the different lattices with each other, they were created in such a way that their determinants were equal. By Minkowski's inequality (theorem 1), this implies that the length of a shortest vector has the same upper bound. Also the expected length of the shortest vector is the same. More precisely, given the dimension n and the determinant d, the lattices were created as

$$\Lambda_m = \left\{ \mathbf{x} \mid \mathbf{Ax} \equiv \mathbf{0} \pmod{p\left(d^{1/m}\right)} \right\}$$

where $p(x)$ is the smallest prime equal to or greater than x and $\mathbf{A}$ is an $m \times n$ matrix with random entries. Since the determinant is given by the product of the cycle lengths, we see that all the Λ_m have approximately the same determinant, which means that it makes sense to compare the results of the LLL algorithm on them.

An important factor is how the starting point for LLL is chosen. When we compute the basis matrix from the null space and add m rows of the form $q\mathbf{e}_k$ for unit vectors $\mathbf{e}_k$, we get a basis where the last rows are much shorter than the first ones. Micciancio [9] suggests the Hermite Normal Form (HNF) as a standard representation of a lattice. The HNF can be computed in polynomial time [15] and we can easily find an upper bound for the coordinates. The basis derived from the null space is in already in HNF, and for the results presented use this basis is used as starting point for LLL.

4.2 Result of the LLL algorithm

In the experiments, the LLL algorithm was executed with 75-dimensional lattices created as explained above as input. The algorithm was executed at least four times for each number of cycles, and the length of the output vector was noted. The result is given in figures 1. The number of iterations needed by the algorithm to finish is given in figure 2.

As we can see in figure 1, the length of the vector produced by the LLL algorithm does not seem to depend on the cycle structure of the lattice examined. From figure 2 it seems that the number of iterations needed by the LLL algorithm to finish in our experiments decreases with the number of cycles.

Since the starting point for a lattice Λ of dimension n with m cycles contains vectors of length $q \approx \sqrt[m]{\det(\Lambda)}$, we get a better starting point for a higher

Figure 1. Shortest vector found by LLL in 75-dimensional lattices with constant determinant

Figure 2. Number of iterations as function of the number of cycles for a 75-dimensional lattice

number of cycles. Experimental data indicate that once the LLL algorithm has reached a point where the length of the shortest vector is that of the starting point for lattice with a higher number of cycles, the progress of the algorithm is similar for both lattices.

5 Complexity of computing short vectors in a lattice with many cycles

We will now present an **NP**-completeness proof for lattices with a maximum number of cycles. We will prove that even if an n-dimensional lattice has $n-1$ cycles the problem of deciding whether there is a vector shorter than a given length in ℓ_∞-norm is **NP**-complete.

The problem that we will discuss is the following:

Definition 1. SVML_∞ *is the problem of finding a short vector in a lattice with maximal number of cycles. Let* $\Lambda \subseteq \mathbb{Z}^n$ *be a lattice which has* $n-1$ *cycles of equal length* q*, and let* $k \in \mathbb{Z}$*. Then* (Λ, k) *is a* YES*-instance if there exists* $\mathbf{v} \in \Lambda$ *such that* $\|\mathbf{v}\|_\infty \leq k$*, and a* NO*-instance otherwise.*

We will prove the following theorem

Theorem 6. SVML_∞ *is* ***NP****-complete.*

Proof. We first note that SVML_∞ is in **NP**. Given a vector $\mathbf{v}$ we can in polynomial time verify that $\|\mathbf{v}\|_\infty \leq k$ and that $\mathbf{v} \in \Lambda$ by solving the system of linear equations $\mathbf{Bx} = \mathbf{v}$ where $\mathbf{B}$ is a basis matrix of Λ and check that $\mathbf{x}$ is integral. This can be done in polynomial time.

Before we continue the proof, we introduce some notation. For any $a \in \mathbb{R}$, define

$$\{a\} := |a \bmod \mathbb{Z}| = \min_{k \in \mathbb{Z}}(|a-k|) \ .$$

Informally, $\{a\}$ is the distance from a to the closest integer. We also introduce a related notation for vectors. For any $\mathbf{v} \in \mathbb{R}^n$, define

$$\{\{\mathbf{v}\}\} := \|\mathbf{v} \bmod \mathbb{Z}^n\|_\infty = \max_{i=1}^{n}(\{v_i\}) \ .$$

$\{\{\mathbf{v}\}\}$ can be seen as the distance between $\mathbf{v}$ and the closest integral vector, given that we by distance mean the max-norm.

We prove that SVML_∞ is **NP**-hard by reducing from good simultaneous Diophantine approximation in ℓ_∞-norm, GDA_∞, which was proven **NP**-hard by Lagarias [6]. GDA_∞ is the following problem. Given a vector $\alpha \in \mathbb{Q}^n$ and integers N and s decide whether there exists an integer Q such that

$$1 \leq Q \leq N$$

and

$$\{\{Q\alpha\}\} \leq 1/s \ .$$

In other words, given a vector of rational numbers, we want to find good approximations to the components of this vector using rationals with a small common denominator.

We note that we can always assume that α is of the form

$$\alpha = \left(\frac{a_1}{b}, \frac{a_2}{b}, \ldots, \frac{a_n}{b} \right) \ .$$

Should α not be of this form, we can always rewrite all its components using the least common denominator.

We start by proving that a revision of GDA_∞, $\textsc{r}\mathrm{GDA}_\infty$, is **NP**-hard.

Definition 2. $\textsc{r}\mathrm{GDA}_\infty$ *is the following problem: Given integers s, q and N and a vector $\beta = (k_1, k_2, \ldots, k_n)/B \in \mathbb{Q}^n$ where $B = N^2 s(s-2)^q$, decide whether there exists an integer Q such that*

$$1 \le Q \le N^2(s-2)^q + \frac{N}{2}$$

and

$$\{\{Q\beta\}\} \le \frac{1}{s} + \frac{1}{2Ns(s-2)^q} \ .$$

Lemma 1. $\textsc{r}\mathrm{GDA}_\infty$ *is **NP**-hard.*

Proof. We reduce GDA_∞ to $\textsc{r}\mathrm{GDA}_\infty$. Let α, N and s be an instance of GDA_∞. Assume b is the common denominator of α. Let c be the least integer such that $1/s < c/b$. Now choose q as the least integer for which

$$\frac{1}{s} + \frac{1}{Ns(s-2)^q} < \frac{c}{b}$$

and

$$(s-2)^q > N$$

and let $B = N^2 s(s-2)^q$. In other words, we choose q so that there is no multiple of $1/b$ in the interval $[1/s, 1/s + N/B)$.

Let the vector

$$\beta' = (k_1, k_2, \ldots, k_n)/B$$

with integral components $k_1, k_2, \ldots, k_n$ be such that $\|\alpha - \beta'\|_\infty$ is minimized. This can be done in polynomial time using ordinary division. It is easy to see that $\|\alpha - \beta'\|_\infty \le 1/(2B)$.

We now define a new vector

$$\beta =$$

$$\left(\beta', \frac{1}{Ns}, \frac{1}{Ns(s-2)}, \ldots, \frac{1}{Ns(s-2)^q}, \frac{1}{N^2 s}, \frac{1}{N^2 s(s-2)}, \ldots, \frac{1}{N^2 s(s-2)^q} \right) \ ,$$

that is, we append some new elements to the vector β'.

We see that β, N, q and s form an instance of $\textsc{r}\mathrm{GDA}_\infty$. Since q is logarithmic in N, β is not more than polynomially larger than α. Also the bit size of the common denominator does not grow more than polynomially.

We want to prove that Q is a solution of this $\textsc{r}\mathrm{GDA}_\infty$ problem if and only if it is a solution of this original GDA_∞ problem.

Let Q be a solution of the $\textsc{r}\mathrm{GDA}_\infty$ instance. We want to prove that Q also is a solution of the original GDA problem.

We first prove that $Q \leq N$. We know that $1/(Ns)$ is a component of β. Since Q is a solution,

$$\left\{ Q \frac{1}{Ns} \right\} \leq \frac{1}{s} + \frac{1}{2s(s-2)^q} \ .$$

This implies that either

$$Q \frac{1}{Ns} \leq \frac{1}{s} + \frac{1}{2s(s-2)^q}$$

or

$$Q \frac{1}{Ns} \geq 1 - \left(\frac{1}{s} + \frac{1}{2s(s-2)^q} \right)$$

which can be rewritten as

$$Q \leq N + \frac{N}{2(s-2)^q}$$

or

$$Q \geq Ns - N - \frac{N}{2(s-2)^q} \ .$$

Since Q is integral, these two conditions imply that

$$Q \leq N$$

or

$$Q \geq N(s-1) \ .$$

We also have that $\frac{1}{Ns(s-2)}$ is a component of β. The corresponding calculations for this component show that $Q \leq N(s-2) < N(s-1)$ or $Q \geq N(s-2)(s-1)$. We can use the same reasoning for the components β_{n+1} up to β_{n+q+1} (remember that $\beta_{n+q+1} = 1/\left(Ns(s-2)^q\right)$), which shows that either

$$Q \leq N$$

or

$$Q \geq N(s-2)^q(s-1) - \frac{N}{2} \ .$$

We do the same thing with $\beta_{n+q+2} = 1/(N^2 s)$, which gives us that either

$$Q \leq N^2$$

or

$$Q > N^2(s-2) \ .$$

Since $(s-2)^q > N$ this implies together with the previous results that $Q > N^2(s-2)$ unless $Q \leq N$. Going through the remaining components we finally get that

$$Q \leq N$$

or

$$Q \geq N^2(s-2)^q(s-1) - \frac{N}{2} \ .$$

Since we in the definition of the problem stated that $Q \leq N^2(s-2)^q + N/2$, the only remaining possibility is that $Q \leq N$.

We now prove that $\{\{Q\alpha\}\} \leq 1/s$. We observe that $\{\{Q\alpha\}\} = k/b$ for some integer k (remember that b is the common denominator of α). We know that $\{Q\beta'\} \leq \{Q\beta\} \leq 1/s + 1/(2B)$. Since the distance between α and β' is at most $1/(2B)$ we can conclude that

$$\begin{array}{l} \{\{Q\alpha\}\} \leq \{\{Q\beta'\}\} + Q\frac{1}{2B} \\ \qquad\qquad \leq \frac{1}{s} + \frac{1}{2Ns(s-2)^q} + \frac{1}{2Ns(s-2)^q} \\ \qquad\qquad = \frac{1}{s} + \frac{1}{Ns(s-2)^q} \end{array} \ .$$

But, as we just stated, the approximation error in α is always a multiple of $1/b$ and since we have chosen q such that $1/s + 1/(Ns(s-2)^q)$ does not pass a $1/b$ boundary, this must indeed be

$$\{\{Q\alpha\}\} \leq \frac{1}{s} \ .$$

This concludes the proof that Q is a solution of the GDA$_\infty$ instance if it is a solution of the rGDA$_\infty$ instance.

Now assume that Q is a solution of the GDA$_\infty$ instance. This means that $\{Q\alpha\} \leq 1/s$ and $Q \leq N$. We first note that $\{Q\beta_i\} \leq Q\beta_i \leq N\beta_i \leq 1/s$ for $i = n+1, \ldots, n+2q+2$, i.e., the appended components. This means that we only need to consider β'. We know that $\|\alpha - \beta'\|_\infty \leq 1/(2B)$, which means that

$$\{\{Q\beta'\}\} \leq \{\{Q\alpha\}\} + Q\frac{1}{2B} \leq \frac{1}{s} + \frac{1}{2Ns(s-2)^q}$$

and we can conclude that Q is a solution of the rGDA$_\infty$ problem.

This proves that the reduction is correct. We now turn to the proof of the **NP**-hardness of SVML$_\infty$. We do this by reducing from rGDA$_\infty$. Let β, N and s be an instance of rGDA$_\infty$, with $\beta = \left(\frac{k_1}{B}, \frac{k_2}{B}, \ldots, \frac{k_n}{B}\right)$. We create the lattice with the following $(n+1) \times (n+1)$ matrix as its basis matrix (the basis vectors are the rows of the matrix)

$$\mathbf{A} = \begin{pmatrix} 1/B & k_1/B & k_2/B & \ldots & k_n/B \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix} \ .$$

If we multiply this lattice by B we get an $(n+1)$-dimensional integral lattice. According to theorem 2, the lattice $B \cdot \mathbf{A}$ has n cycles of length B. This means that $B \cdot \mathbf{A}$ and $B/s + B/(2Ns(s-2)^q)$ is an instance of SVML$_\infty$.

We now want to prove that this SVML$_\infty$ instance is a YES instance if and only if the original RGDA$_\infty$ is a YES instance.

Assume that the SVML$_\infty$ instance is a YES instance, i.e., there is a Q such that

$$\max\left\{Q\frac{1}{B}, \{\{Q\beta\}\}\right\} \leq \frac{1}{s} + \frac{1}{2Ns(s-2)^q}$$

which implies that

$$\{\{Q\beta\}\} \leq \frac{1}{s} + \frac{1}{2Ns(s-2)^q}$$

and

$$Q \leq N^2(s-2)^q + \frac{N}{2}$$

so Q is a solution of the RGDA$_\infty$ instance.

Assume that the RGDA$_\infty$ instance is a YES instance. Then there is a $Q \leq N^2(s-2)^q + N/2$ such that

$$\{\{Q\beta\}\} \leq \frac{1}{s} + \frac{1}{2Ns(s-2)^q} \ .$$

We can calculate

$$Q\frac{1}{B} \leq \frac{1}{s} + \frac{1}{2Ns(s-2)^q}$$

which implies that

$$\max\left\{Q\frac{1}{B}, \{\{Q\beta\}\}\right\} \leq \frac{1}{s} + \frac{1}{2Ns(s-2)^q} \ ,$$

i.e., the SVML$_\infty$ instance is a YES instance.

This concludes the proof that SVML$_\infty$ is **NP**-complete.

Acknowledgements

I would like to thank Johan Håstad for valueable feedback and ideas during the preparation of this paper. I would also like to thank the anonymous referees for pointing out mistakes and possible improvements.

References

1. M. Ajtai. Generating Hard Instances of Lattice Problems. *Proc. 28th ACM Symposium on Theory of Computing*, pages 99–108, 1996.
2. M. Ajtai. The shortest vector problem in ℓ_2 is **NP**-hard for randomized reductions. *Proc. 30th ACM Symposium on the Theory of Computing*, pages 10–19, 1998.
3. J-Y. Cai and A. Nerurkar. An Improved Worst-Case to Average-Case Connection for Lattice Problems. *Proc. 38th IEEE Symposium on Foundations of Computer Science*, pages 468–477, 1997.
4. O. Goldreich and S. Goldwasser. On the limits of non-approximability of lattice problems. *Journal of Computer and System Sciences*, Academic Press, 60(3):540–563, 2000. Can be obtained from `http://www.eccc.uni-trier.de/eccc`.
5. Kabatjanskii and Levenshtein. Bounds for Packings on a Sphere and in Space. *Problems of Information Transmission 14*, 1:1–17, 1978.
6. J.C. Lagarias. The Computational Complexity of Simultanous Diophantine Approximation Problems. *SIAM Journal of Computing*, 14:196–209, 1985.
7. A.K. Lenstra, H.W. Lenstra and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen* 261:515–534, 1982.
8. D. Micciancio. The Shortest Vector in a Lattice is Hard to Approximate within Some Constant. *Proc. 39th IEEE Symposium on Foundations of Computer Science*, 1998, 92–98.
9. D. Micciancio. Lattice Based Cryptography: A Global Improvement. Technical report, Theory of Cryptography Library, 1999. Report 99-05. Can be obtained from `http://eprint.iacr.org`.
10. H. Minkowski. Über die positiven quadratischen Formen und über kettenbruchähnliche Algorithmen. *Crelles Journal für die Reine und Angewandte Mathematik*, 107:278–297, 1891.
11. A. Paz and C.P. Schnorr. Approximating Integer Lattices by Lattices with Cyclic Lattice Groups. *Automata, languages and programming (Karlsruhe)*, 1987, 386–393.
12. C.P. Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
13. V. Shoup. NTL: A Library for doing Number Theory. Can be obtained from `http://www.shoup.net`.
14. H.J.S. Smith. On Systems of Linear Indeterminate Equations and Congruences. *Philosophical Transactions of the Royal Society of London*, 151:293–326, 1861.
15. A. Storjohann and G. Labahn. Asymptotically Fast Computation of Hermite Normal Forms of Integer Matrices. *ISAAC '96*, 1996, 259–266.
16. P. van Emde Boas. Another **NP**-complete partition problem and the copmlexity of computing short vectors in lattices. Technical Report 81-04. Mathematics Department, University of Amsterdam, 1981. Can be obtained from `http://turing.wins.uva.nl/~peter`.