The Beauty and the Beasts—The Hard Cases in LLL Reduction

Saed Alsayigh[1(✉)], Jintai Ding[1], Tsuyoshi Takagi[2,3], and Yuntao Wang[4]

[1] Department of Mathematical Sciences, University of Cincinnati, Cincinnati, USA
alsayisd@mail.uc.edu, jintai.ding@gmail.com
[2] Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan
takagi@imi.kyushu-u.ac.jp
[3] CREST, Japan Science and Technology Agency, Kawaguchi, Japan
[4] Graduate School of Mathematics, Kyushu University, Fukuoka, Japan
y-wang@math.kyushu-u.ac.jp

Abstract. In this paper, we will systematically study who indeed are the hard lattice cases in LLL reduction. The "hard" cases here mean for their special geometric structures, with a comparatively high "failure probability" that LLL can not solve SVP even by using a powerful relaxation factor. We define the perfect lattice as the "Beauty", which is given by basis of vectors of the same length with the mutual angles of any two vectors to be exactly 60°. Simultaneously the "Beasts" lattice is defined as the lattice close to the Beauty lattice. There is a relatively high probability (e.g. 15.0% in 3 dimensions) that our "Beasts" bases can withstand the exact-arithmetic LLL reduction (relaxation factors δ close to 1), comparing to the probability (corresponding <0.01%) when apply same LLL on random bases from TU Darmstadt SVP Challenge. Our theoretical proof gives us a direct explanation of this phenomenon. Moreover, we give rational Beauty bases of 3 and 8 dimensions, an irrational Beauty bases of general high dimensions. We also give a general way to construct Beasts lattice bases from the Beauty ones. Experimental results show the Beasts bases derived from Beauty can withstand LLL reduction by a stable probability even for high dimensions. Our work in a way gives a simple and direct way to explain how to build a hard lattice in LLL reduction.

Keywords: Lattice · LLL reduction · Hard cases · Post-Quantum Cryptography

1 Introduction

As one of the compelling candidates in Post-Quantum Cryptography, Lattice-based cryptography is now a very hot topic due to all the versatile constructions based on the Learning With Errors (LWE) and the Ring Learning With Errors (RLWE) problems [10,17]. But to select a practical parameter, we must have a solid understanding on the hardness of the reduction algorithms. But as far

S. Obana and K. Chida (Eds.): IWSEC 2017, LNCS 10418, pp. 19–35, 2017.
DOI: 10.1007/978-3-319-64200-0_2

as we know, now there is a clear gap between the theoretical estimation of the approximation factor and the case of experiments. By now, not much work has been done in this direction and this serious gap can be a real roadblock for us to move forward in this direction. However there are some of related works in other direction. These works aim to analyze the computational complexity of reduction algorithms in low dimensional bases, as in Semaev [18] and Nguyen, Stehlé [14] for more details. There are also some previous works on γ-unique SVP problem. Some practical evidence shows that unique SVP is potentially easier as γ (the magnitude gap between the shortest and second independent shortest vector) becomes larger [5]. Much of the subsequent works concentrated on evaluating a more reasonable bound of γ, such that LLL reduction algorithm can derive a shortest vector successfully [8,11]. Oppositely, our work is to find "hard" cases, namely "Beasts", such that LLL can not find the shortest vector by a relatively high probability. In other words, our Beasts lattices give a potentiality to withstand a strong LLL reduction. Note that the reader should differentiate "hard" cases from the so-called "worst" cases concerning the computational complexity [1,14,16].

It is known that LLL can not guarantee a shortest vector even in the 3 dimensional lattices. In principle, when we do reduction to make the basis better and better, one reason we could not reach the best reduction is that in the LLL reduction process, we are essentially trying to do the best local reduction (2 vectors one time) to achieve the global reduction, which is very much related to local optimization and global optimization. Such a local method decides that there is a high probability that we will make the local decisions is not really right globally. What BKZ is doing is exactly to compensate such a defect, namely improve the global reduction by doing a better local reduction (instead of a local reduction in 2 dimension, a better reduction through searching in 3 or higher dimensions) [19]. For us, we want to find out what is really happening in such a local reduction. It turns out that the symmetry of the perfect lattice plays a key role here.

In this paper, we will open a new direction to look at this problem from a different angle, namely we will find out mathematically how we characterize the hard cases and explain from the point view of mathematical structure why they are hard. The reason behind is that we hope this will allow us first to fill the theoretical gap we mentioned above and further more this may give us new ideas on how to improve further the reduction algorithms. This is a direction which, we believe, was not explored before, and we do not know much about at the moment.

Our method is to start from low dimensional cases. Here through intuitive understanding and experiments, we realize that the key idea in low dimensions is related to the symmetry of the underlying lattice. By symmetry here, we mean the isometry group that keep the lattice invariant. We will first define a perfect lattice – bases vectors of 60° angles respectively with the same length, which is called "Beauty" in this paper. Then we can use this to explain that if we have a basis which is very close to the perfect lattice, which makes the decision to

find the shortest very hard even with very high precision as the exact-arithmetic reduction. Namely there are many pretenders of "almost good basis" for the good basis, which comes from the weakness of selecting a non-optimal parameter δ. This forms the barriers to find the good basis and therefore the shortest vector. Or even in more simple terms, there are many bases that could pass the LLL criterion to be a reduced basis. The "Beasts", namely the really hard cases, are the ones very near to the Beauty lattices.

It is practically impossible (in terms of probability) to randomly sample a good basis with the same length and close to 60° simultaneously, from the point of distribution. For this, we need some form of good probability experiments. We give a thorough analysis of 3 dimensional "Beauty" and "Beasts", which illuminate us how to construct higher dimensional cases. In experiments, we reduce our Beasts lattice bases using the exact-arithmetic LLL in NTL library. In our implementations, we consider the effects of some variable parameters of our constructed Beasts bases and LLL. Under the same condition, we also compare the failure probability for NTL rational LLL when it reduces random lattice bases from Darmstadt SVP Challenge. The experimental results show that our "Beasts" bases can keep a tough resistance to strong LLL (with relaxation factor δ close to 1), comparing with the random bases.

The paper is organized as follows. Section 2 covers notation and background on lattice and reduction algorithms. We give an explicit definition of Beauty and Beasts lattices in Sect. 3. In Sect. 4, the structures of the Beauty and the Beasts bases on 3 dimensional lattices are proposed, including the main theorem proof and experimental results. The exploratory structure of the Beauty and the Beasts bases for higher dimensional lattices and some experimental analysis are presented in Sect. 5. Finally some concluding remarks and future works are given in Section 6

2 Preliminaries

In this section we review some basic definitions and theorems related to some classical lattice algorithms.

2.1 Lattice Theory

Let linearly independent vectors set $(\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^{n \times m}$ be a basis B of lattice $L(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i, x_i \in \mathbb{Z}\}$. In our work, we use full-rank bases with row form vectors in lattice and we denote the i-th row vector of the basis by $B[i]$. The volume of L is given by the volume of fundamental domain $\mathcal{F}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{t_1\mathbf{b}_1 + t_2\mathbf{b}_2 + \cdots + t_n\mathbf{b}_n : 0 \leq t_i < 1\}$, which is equal to $\|\det(B)\|$. Generally if B is a full-rank matrix and $U_i (i \in [1, \infty))$ are unimodular matrices, $U_i \cdot B$ gives infinitely many bases of $L(B)$ since $\det(U_i) = \pm 1$ give the same volume.

The Euclidean norm of a vector $\mathbf{b} \in \mathbb{R}^m$ is $\|\mathbf{b}\|$. The Gram-Schmidt Orthogonal (GSO) basis $(\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ is given by the following:

(a) $\mathbf{b}_1^* = \mathbf{b}_1$
(b) $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij}\mathbf{b}_j^*$ for all $2 \leq i \leq n$ where $\mu_{ij} = \frac{<\mathbf{b}_i, \mathbf{b}_j^*>}{\|\mathbf{b}_j^*\|^2} (1 \leq j < i \leq n)$

The Gram-Schmidt Orthogonal (GSO) basis. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of $\mathbb{R}^m$ and let $(\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its GSO basis then

(a) $< \mathbf{b}_i, \mathbf{b}_j^* >= 0 \quad \forall 1 \leq i < j \leq n.$
(b) $span(\mathbf{b}_1^*, \ldots, \mathbf{b}_k^*) = span(\mathbf{b}_1, \ldots, \mathbf{b}_k) \quad \forall 1 \leq k \leq n.$
(c) $\|\mathbf{b}_k^*\| \leq \|\mathbf{b}_k\| \quad \forall 1 \leq k \leq n.$

Size-reduction. A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is called size-reduced with factor $\eta \geq \frac{1}{2}$ if its GSO family satisfies $|\mu_{ij}| \leq \eta$ for all $1 \leq j < i \leq d$.

Here it's important to know that η is usually $\frac{1}{2}$, but for the floating-point LLL one takes it at least slightly larger since the μ_{ij} will be known only approximately.

δ-reduction. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}_n^n$ be a basis of lattice L and let $(\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its GSO basis then $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is called δ-reduced basis if it satisfies:

(1) $|\mu_{ij}| \leq \eta$ for all $1 \leq j < i \leq n$ (Size Condition)
(2) $\|\mathbf{b}_i^* + \mu_{i,i-1}\mathbf{b}_{i-1}^*\|^2 \geq \delta\|\mathbf{b}_{i-1}^*\|^2$ for all $2 \leq i \leq n$ where $\frac{1}{4} < \delta < 1$. (Lovász Condition)

δ is called relaxation factor in LLL reduction algorithm.

Minkowski's minima [13]. Let L be a lattice with full-rank basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^{n \times n}$. We denote the Euclidean norm of the shortest vector in L as $\lambda_1(L)$. For all $1 \leq i \leq n$, Minkowski's ith minimum $\lambda_i(L)$ is defined as the minimum of $\max_{1 \leq j \leq i} ||\mathbf{b}_j||$ over all i linearly independent lattice vectors $(\mathbf{b}_1, \ldots, \mathbf{b}_n) \in L$.

SVP and γ-unique SVP. For a given basis $B \in \mathbb{R}^{n \times n}$, the Shortest Vector Problem (SVP) is to find the shortest non-zero vector in $L(B)$. It is called unique SVP, if $\lambda_1(L) \ll \lambda_2(L)$ is guaranteed in SVP. And the γ-unique SVP problem is scaling the bound by a positive multiple as $\gamma\lambda_1(L) < \lambda_2(L)$. The auxiliary condition can be seen as a bounded gap between the first Minkowski's minimum and the second Minkowski's minimum. It is known that if the gap is bigger, it is easier to find the shortest vector by a certain algorithm [5]. However, in our work we concentrate on constructing the hard bases with very small γ, such that by a high probability the LLL reduction can not find the shortest vector successfully.

2.2 Lattice Algorithms

Lagrange's algorithm. The Lagrange's algorithm can definitely solves the SVP in 2 dimensional lattice in polynomial time. Actually it finds a basis achieving the first two Minkowski's minima. In algorithmic principle the Lagrange's algorithm is similar to Euclids algorithm. Refer to [7] for more details.

LLL reduction. LLL reduction algorithm is a practical algorithm that is proved to terminate in a polynomial time and gives an δ-reduced basis [9]. However that is proven when $\delta \in (\frac{1}{4}, 1)$ and in many applications they used η slightly larger

Algorithm 1. LLL algorithm

Input: a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L, and a constant $\delta \in (\frac{1}{4}, 1)$.
Output: the output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L satisfies definition above for each i from 2 to d.

$i \leftarrow 2$
while $i \leq d$ **do**
  $\mathbf{b}_i \leftarrow \mathbf{b}_i - \sum_{j=i-1}^{1} \lceil \mu_{i,j} \rfloor \mathbf{b}_j$
  if $\|\mathbf{b}_i^*\|^2 \geq (\delta - \mu_{i,i-1}^2)\|\mathbf{b}_{i-1}^*\|^2$ **then**
    $i \leftarrow i + 1$
  else
    $swap(\mathbf{b}_i, \mathbf{b}_{i-1})$
    $i \leftarrow \max\{2, i - 1\}$
  end if
end while

than $\frac{1}{2}$ to guarantee termination in a practical time. We use $\eta = \frac{1}{2}$ in our work. We present a simple implementation to LLL algorithm (Algorithm 1). $\lceil \cdot \rfloor$ denotes rounding a real number to its nearest integer.

Setting δ strictly smaller than 1 can guarantee a polynomial computational complexity of LLL, with respect to the magnitude of the initial basis and the dimension n. Also according to proof of [2], in the so-called "ideal" or "optimal" case of $\delta = 1$, LLL is still polynomial with respect to the magnitude of the basis, but it is still an open problem whether it is also polynomial in the dimension n. In our work we consider the general case with δ close to 1 but strictly smaller than 1.

Note that our work is easily confused from works in [3,14], which are trying to improve the LLL algorithm by using floating-point calculation or rounding technic, etc. We want to explore the hard cases for LLL from the geometrical structure of lattice bases, ignoring the efficiency of LLL so far. And also the new algorithm proposed in [18] is also a future work for us to modify our conjectures.

Schnorr-Euchner's enumeration algorithm. In 1994, Schnorr and Euchner proposed an enumeration algorithm to search the shortest vector of a lattice [19], which runs in exponential time and the cost is no more than $2^{O(n^2)}$. We denote it by "ENUM" in this paper. ENUM is the most efficient one in practice that can find the shortest vector successively in 100% but very heavy in high dimensions. There are some improvements for it as in [6]. However, we won't describe the details about ENUM here, because in our experiments we just modify if the LLL can produce a shortest vector by performing ENUM on the LLL-reduced basis.

3 The Beauty and the Beasts

Definition of Beauty Lattice. Let $B \in \mathbb{R}^{n \times n}$ be a full-rank basis then we say $L(B)$ is a *Beauty lattice* if its basis fulfils these two properties simultaneously:

1. its vectors are of the same length;
2. the angle between any two basis vectors is 60°.

Example: In $\mathbb{R}^3$, the lattice generated by

$$B = \begin{pmatrix} 1\ 1\ 0 \\ 0\ 1\ 1 \\ 1\ 0\ 1 \end{pmatrix}$$

is a Beauty lattice.

The corresponding basis B is called *Beauty basis*. In the following contents, we denoted the primitive n dimensional Beauty bases by $Beauty_n$ if the integer components in bases vectors are 1. We can get infinite Beauty bases by multiplying any $P \in \mathbb{R}$ by $Beauty_n$. We can see the Beauty lattice in $\mathbb{R}^2$ graphically in Fig. 1.

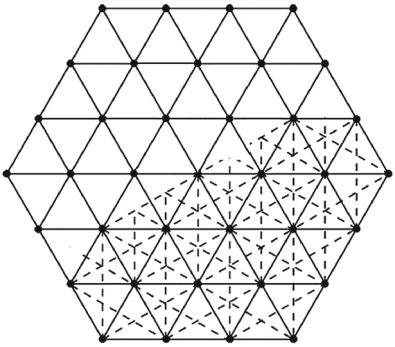

Fig. 1. The structure of Beauty lattices in $\mathbb{R}^2$.

It is folklore that for the symmetric property, the magnitudes of such basis vectors are all as same as the Minkowski's first minimum. In addition, the Beauty bases are already LLL reduced, namely the LLL algorithm will never fail to get the shortest vector on Beauty lattices.

Moreover, experiments show that if the first shortest vector is much shorter than the second one, then the lattice algorithms as LLL will succeed to reach the shortest vector with a much higher possibility [5]. Therefore we can conclude that in the low dimensions space, hard lattice or what we called the *Beast lattice* exist very close to the Beauty one.

Definition of Beast Lattice. A lattice $L(B)$ is called a *Beast lattice* if its basis vectors have the following two properties:

1. they are almost of the same length;
2. the angle between any two basis vectors is almost 60°.

We define the n dimensional *Beasts basis* by denotion $Beasts_n$, which is generated from $Beauty_n$. Obviously there are infinitely many beasts bases around one Beauty basis. We will give a method to build the 3-dimensional beasts bases

in Sect. 4.2 with concrete parameter settings and some theoretical proofs. In Sect. 5, we give the assessing structure for higher dimensional Beast bases.

Two dimensional case. Of course for two dimensional lattices, as the optimal cases of LLL ($\delta = 1$), definitely Lagrange's algorithm can find the vectors of Minkowski's minima for any structure of bases in polynomial time [7]. So our main focus in this paper is cases of 3 dimensions and higher.

4 Three Dimensional Case

In this section, we'll review the performance of classical rational LLL algorithm at first. Then we'll give our main theorem based on some theoretical analyses on a usual 3 dimensional Beasts lattices. And we simply evaluate the gap $\gamma = \lambda_2/\lambda_1$ in 3 dimensional Beasts. At last we show some experimental results to support our conjectures.

4.1 LLL Outputs a Shortest Vector?

As we know, the LLL algorithm gets stronger with the approximate factor δ asymptotically close to 1 in experiments. But for a given basis, it is folklore that LLL can not definitely output the shortest vector even when the dimension is 3. The Goldstein-Mayer bases [4] give us random bases in some sense. And it is used in Hermite Normal Form in the famous SVP Challenge operated by TU Darmstadt [20]. We generated 100,000 3 dimensional random bases from SVP Challenge and did some initial experiments using exact-arithmetic variants of LLL in NTL library [15]. Furthermore, we perform ENUM on the LLL reduced bases to check if the shortest vector is in the LLL reduced bases. We call LLL reduction "succeed" if the shortest one is in the output thereof. Figure 2 shows the number variety of LLL-failed cases in 3 dimensions using $\delta \in (0.25, 1)$.

4.2 Main Theorem

Now we will investigate why LLL reduction fails to get the shortest vector. We start our investigation within $\mathbb{R}^3$ – the 3 dimensional Beauty basis mentioned in Sect. 3. It obviously constitute a regular tetrahedron in 3 dimensional Euclidean space.

$$Beauty_3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

Note that to use a "strong" LLL, we set the parameter $\delta \in [0.9, 1)$. Although we do exam more than 100,000 random bases of $\mathbb{R}^3$ and found it almost succeed to get the shortest vectors, the reality is that LLL reduction can fail in a special cases. Our next theorem gives an examples of that failure cases:

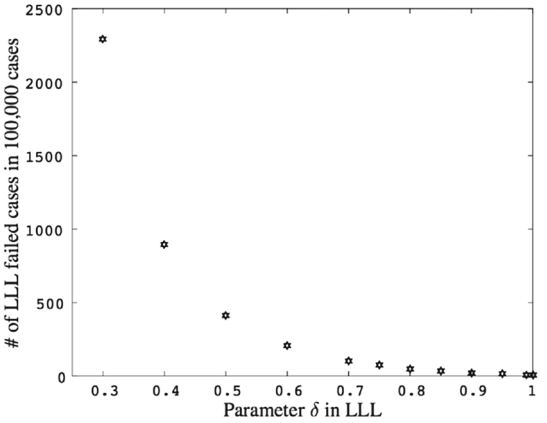

Fig. 2. The number of LLL failed cases in dimension 3 using same bases but different δ.

Let P be a big positive integer number and let $\epsilon_1, \epsilon_2, \epsilon_3$ and ϵ_4 be very small positive integers fulfilling

$$\begin{cases} P \in [2^{x-1}, 2^x - 1] & (x \in \mathbb{Z} \text{ and } x \geq y) \\ \epsilon_i \in [0, 2^y - 1] & (y \in \mathbb{Z} \text{ and } y \geq 2) \\ \delta = 1 - 2^{-z} & (z \in \mathbb{R} \text{ and } z \geq 3.32) \end{cases} \tag{1}$$

$$\epsilon_1 \geq \epsilon_2 > \epsilon_3 \geq \epsilon_4 \geq 0 \tag{2}$$

and

$$x - y - z \geq 1 \text{ and } y \leq z. \tag{3}$$

Then the basis

$$Beasts_3 = \begin{pmatrix} P - \epsilon_1 & P - \epsilon_2 & 0 \\ 0 & P - \epsilon_3 & P - \epsilon_4 \\ P & 0 & P \end{pmatrix}$$

is a 3 dimensional Beasts basis which can withstand exact-arithmetic LLL reduction by a high potentiality. The failure probability of LLL to find the shortest vectors in dimension 3 is stably 15.0%, relatively random bases thereof is <0.01%. Note that for generality we apply stochastic disturbance ϵ_i on each element of first two vectors in $P \cdot Beauty_3$. Certainly one can subtract ϵ_i from all elements or from just one element of each vector. Experimental results are same for all these Beasts.

Theorem 1. Let $Beasts_3$ be the basis defined above and let E be the unimodular matrix

$$E = \begin{pmatrix} 0 & -1 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

then $EBeasts_3$ is an LLL reduced basis with $L(EBeasts_3) = L(Beasts_3)$, but the shortest vector is not in $EBeasts_3$.

Proof. Let $Beasts_3 = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)^t$. At first, we declare that $\mathbf{b_1}$ is the shortest vector in $L(Beasts_3)$ and put the proof in Appendix A, for that in cases of constructed high Beasts bases, the first vector is definitely not the shortest one. Next we prove the second part of the theorem. Set

$$C = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)^t = EB = (\mathbf{b}_3 - \mathbf{b}_2, \mathbf{b}_1 - \mathbf{b}_2, \mathbf{b}_2)^t.$$

We have to show that C satisfies the LLL conditions where

$$C = \begin{pmatrix} P & -(P-\epsilon_3) & \epsilon_4 \\ P-\epsilon_1 & \epsilon_3-\epsilon_2 & -(P-\epsilon_4) \\ 0 & P-\epsilon_3 & P-\epsilon_4 \end{pmatrix}$$

First we will show that C is size-reduced matrix then we'll show it satisfies Lovász condition. Note that in our work, we use $\eta = \frac{1}{2}$ for size reduction of LLL.

$$\begin{aligned} \left|\mu^C_{2,1}\right| &= \left|\frac{< \mathbf{c}_2, \mathbf{c}_1 >}{< \mathbf{c}_1, \mathbf{c}_1 >}\right| \\ &= \left|\frac{P(P-\epsilon_1) - (P-\epsilon_3)(\epsilon_3-\epsilon_2) - (P-\epsilon_4)\epsilon_4}{P^2 + (P-\epsilon_3)^2 + \epsilon_4^2}\right| \\ &= \left|\frac{P^2 - P\epsilon_3(\epsilon_2-\epsilon_1)P + \epsilon_3(\epsilon_3-\epsilon_2) + \epsilon_4(\epsilon_4-P)}{2P^2 - 2P\epsilon_3 + \epsilon_4^2}\right| \\ &\leq \frac{P^2 - P\epsilon_3}{2P^2 - 2P\epsilon_3} \\ &\leq \frac{1}{2} = \eta \end{aligned}$$

$$\begin{aligned} \left|\mu^C_{3,1}\right| &= \left|\frac{< \mathbf{c}_3, \mathbf{c}_1 >}{< \mathbf{c}_1, \mathbf{c}_1 >}\right| \\ &= \left|\frac{-(P-\epsilon_3)^2 + \epsilon_4(P-\epsilon_4)}{P^2 + (P-\epsilon_3)^2 + \epsilon_4^2}\right| \\ &\leq \frac{P^2 - \epsilon_3 P}{2P^2 - 2\epsilon_3 P} \\ &\leq \frac{1}{2} = \eta \end{aligned}$$

To prove $|\mu^C_{3,2}| < \frac{1}{2}$, at first we need to get an approximate lower bound of $|\mu_{21}|$. Using the assumptions (1), (2) and (3), we get $x \geq 7$ and $|\mu_{21}| \geq 0.402$ can also be derived.

$$\begin{aligned} \left|\mu^C_{3,2}\right| &= \left|\frac{< \mathbf{c}_3, \mathbf{c}_2^* >}{< \mathbf{c}_2^*, \mathbf{c}_2^* >}\right| \\ &= \left|\frac{< \mathbf{c}_3, \mathbf{c}_2 > - \mu_{21} < \mathbf{c}_3, \mathbf{c}_1 >}{\|\mathbf{c}_2 - \mu_{21}\mathbf{c}_1\|^2}\right| \\ &= \left|\frac{(P-\epsilon_3)(\epsilon_3-\epsilon_2) - (P-\epsilon_4)^2 + \mu_{21}(P-\epsilon_3)^2 - \mu_{21}\epsilon_4(P-\epsilon_4)}{(P-\epsilon_1-\mu_{21}P)^2 + [\epsilon_3-\epsilon_2+\mu_{21}(P-\epsilon_3)]^2 + (P-\epsilon_4+\mu_{21}\epsilon_4)^2}\right| \\ &= \frac{1}{2}\left|\frac{(P-\epsilon_3)(\epsilon_3-\epsilon_2) - (P-\epsilon_4)^2 + \mu_{21}(P-\epsilon_3)^2 - \mu_{21}\epsilon_4(P-\epsilon_4)}{[(P-\epsilon_3)(\epsilon_3-\epsilon_2) - (P-\epsilon_4)^2 + \mu_{21}(P-\epsilon_3)^2 - \mu_{21}\epsilon_4(P-\epsilon_4)] + \zeta}\right| \end{aligned}$$

We denote ζ the long remainders. From $P \geq 2^{x-1} \geq 2^{y+z} \geq 2^{y+4}$ we can get

$$\frac{\epsilon_1}{P} < \frac{2^y - 1}{2^{x-1}} < \frac{2^y - 1}{2^{y+4}} < \frac{1}{16}(1 - \frac{1}{2^y}) < \frac{1}{16}. \tag{4}$$

Hence we can get the following lower bound of $|\mu_{21}|$ as:

$$\begin{aligned}
|\mu_{21}| &= \left| \frac{< \mathbf{c}_2, \mathbf{c}_1 >}{< \mathbf{c}_1, \mathbf{c}_1 >} \right| \\
&= \left| \frac{P(P-\epsilon_1) + (\epsilon_2 - \epsilon_3)(P-\epsilon_3) - \epsilon_4(P-\epsilon_4)}{P^2 + (P-\epsilon_3)^2 + \epsilon_4^2} \right| \\
&= \left| \frac{P^2 + (\epsilon_2 - \epsilon_1 - \epsilon_3 - \epsilon_4)P + \epsilon_3^2 - \epsilon_2\epsilon_3 + \epsilon_4^2}{2P^2 - 2P\epsilon_3 + \epsilon_3^2 + \epsilon_4^2} \right| \\
&> \frac{P^2 - 3\epsilon_1 P - \epsilon_1^2}{2P^2 + 2\epsilon_1^2} \\
&= \frac{1}{2} \cdot (1 - \frac{2\epsilon_1^2}{P^2 + \epsilon_1^2} - \frac{3P\epsilon_1}{P^2 + \epsilon_1^2}) \\
&> \frac{1}{2} - (\frac{\epsilon_1}{P})^2 - \frac{3}{2}\frac{\epsilon_1}{P} \\
&> \frac{1}{2} - (\frac{1}{16})^2 - \frac{3}{2} \cdot \frac{1}{16} \\
&\approx 0.402
\end{aligned}$$

Using this result, we compute $\zeta > 0$ and finally we can derive $|\mu_{3,2}^C| \leq \frac{1}{2} = \eta$. During the all proof we omit some factors since we assumed that the epsilons' values are very small respect to P value. So the omitted value doesn't affect in the sum value.

Next we want to show that the vectors of C also satisfy Lovász condition. From Gram-Schmidt theorem, we have

$$\|\mathbf{b}_1 - \mathbf{b}_2\|^2 = \|(\mathbf{b}_1 - \mathbf{b}_2)^*\|^2 + (\mu_{2,1}^C)^2 \|(\mathbf{b}_3 - \mathbf{b}_2)^*\|^2.$$

Hence we can get

$$\|(\mathbf{b}_1 - \mathbf{b}_2)^*\|^2 = \|\mathbf{b}_1 - \mathbf{b}_2\|^2 - (\mu_{2,1}^C)^2 \|(\mathbf{b}_3 - \mathbf{b}_2)^*\|^2. \tag{5}$$

Now we should proof that $\|(\mathbf{b}_1 - \mathbf{b}_2)^*\|^2 - \delta\|\mathbf{b}_3 - \mathbf{b}_2\|^2 \geq 0$ as follows.

$$\begin{aligned}
&\|(\mathbf{b}_1 - \mathbf{b}_2)^*\|^2 - \delta\|\mathbf{b}_3 - \mathbf{b}_2\|^2 \\
&= ((P-\epsilon_1)^2 + (\epsilon_3 - \epsilon_2)^2 + (P-\epsilon_4)^2) - \delta(P^2 + (P-\epsilon_3)^2 + \epsilon_4^2) \qquad (6) \\
&= 2(1-\delta)P^2 + 2(\delta\epsilon_3 - \epsilon_1 - \epsilon_4)P + \epsilon_1^2 + (\epsilon_3 - \epsilon_2)^2 + (1-\delta)\epsilon_4 - \delta\epsilon_3^2.
\end{aligned}$$

Note that z should take big value such that $\delta \to 1$. We consider the dominative part of Eq. (6). Finally, using our assumption (1) and (2), and from formula (5), we can get

$$\begin{aligned}
\|(\mathbf{b}_1 - \mathbf{b}_2)^*\|^2 &= \|\mathbf{b}_1 - \mathbf{b}_2\|^2 - (\mu_{2,1}^C)^2 \|(\mathbf{b}_3 - \mathbf{b}_2)^*\|^2 \\
&\geq \delta\|\mathbf{b}_3 - \mathbf{b}_2\|^2 - (\mu_{2,1}^C)^2 \|(\mathbf{b}_3 - \mathbf{b}_2)^*\|^2 \\
&\geq \delta\|\mathbf{b}_3 - \mathbf{b}_2\|^2 - (\mu_{2,1}^C)^2 \|\mathbf{b}_3 - \mathbf{b}_2\|^2 \\
&\geq (\delta - (\mu_{2,1}^C)^2)\|\mathbf{b}_3 - \mathbf{b}_2\|^2.
\end{aligned}$$

Similarly, we have the equation:

$$\|\mathbf{b}_3\|^2 = \|\mathbf{b}_3^*\|^2 + (\mu_{3,1}^C)^2\|(\mathbf{b}_3 - \mathbf{b}_2)^*\|^2 + (\mu_{3,2}^C)^2\|(\mathbf{b}_1 - \mathbf{b}_2)^*\|^2$$

and we can proof that it satisfies the Lovász condition by our assumptions above.

Therefore C satisfies $\mu_{i,j}^C \leq \eta$ and Lovász condition too. Hence C is LLL reduced and by the construction process we know that $\|C[i]\| > Beasts_3[1] = \lambda_1[L]$, for $\forall i \in \{1, 2, 3\}$.

Finally, it is easy to see that $EBeasts_3$ is a basis for $L(Beasts_3)$ therefore $L(EBeasts_3) = L(Beasts_3)$. □

Theorem 1 means that even if we apply a strong LLL reduction to $EBeasts_3$, LLL fails to get the shortest vector in our parameter setting. (1), (2) and (3) are the most important conditions in our theoretical proof. Moreover, the theoretical conjecture perfectly matches our experimental results.

Moreover, using the condition (4): $\frac{\epsilon_2}{P} \leq \frac{\epsilon_1}{P} \leq \frac{1}{16}$, we can easily get the gap between λ_1 and λ_2 in our $Beasts_3$ bases:

$$\gamma = \frac{\lambda_2}{\lambda_1} \leq \frac{\sqrt{2P^2}}{\sqrt{(P-\epsilon_1)^2 + (P-\epsilon_2)^2}} \leq \sqrt{\frac{2P^2}{2P^2 - 2(\epsilon_1+\epsilon_2)P}} = \frac{1}{\sqrt{1 - \frac{\epsilon_1+\epsilon_2}{P}}} \leq \sqrt{\frac{8}{7}} \approx 1.069.$$

4.3 Experimental Results in 3 Dimensional Beasts Lattices

In our experiments, we implemented our main program using C++ language and calculated the random unimodular matrices from Magma computational algebra system [12]. About the size of x, we also follow the strategy of TU Darmstadt SVP Challenge [20], which takes the size of entries of n dimensional random lattice basis as $10 \cdot n$ bits. In our experiments, we also considered the size of entries and constructing technique in unimodular matrices. The experimental results show that with the size of entries increasing, the chance for LLL to find the shortest vector is bigger. Since we compare the performance of our Beasts bases with the randomly generated bases from SVP Challenge, where both original elements are $10 \cdot n$ bits length, the elements in unimodular matrices should be as small as possible (as size smaller than n and close to 0). Because in LLL reductions, sometimes the shortest vector in the reduced basis is not in the first position, for the sake of fairness, we use ENUM algorithm to check if the shortest one is in the whole reduced bases or not.

Now we will show two experimental results. The first one is when we apply the 100,000 unimodular matrices to randomize 100,000 generated Beasts bases, the LLL (with $\delta \to 1$) fails to solve SVP by 15.0% probability under our parameters' assumption (1), (2) and (3). And this failure probability will become 18.3% if we apply the unimodular matrices on the same Beasts basis again. However for random bases generated from TU Darmstadt SVP Challenge [20], this failure probability is <0.01% for $\delta \in [0.9, 1)$. This result means our constructed 3 dimensional Beasts lattice can withstand powerful LLL much better than random lattices.

The second experiment is on the failed cases as $EBeasts_3$ in our main theorem. In Table 1, we can see that under our parameters' assumption (1), (2) and (3), the failed cases will keep a stable probability that LLL can not find the shortest vector with $\delta \to 1$. Simultaneously this result validates that the LLL failed cases are all LLL reduced as what we proved on $EBeasts_3$.

Table 1. Some experimental results using our constructed 3-dimensional Beast bases. Note that we call LLL failed if the shortest vector was not in the LLL-reduced basis.

x	y	z	δ	LLL failed prob.
10	2	6.64	$1-10^{-2}$	100%
20	2	16.6	$1-10^{-5}$	100%
30	9	19.9	$1-10^{-6}$	100%
30	5	23.25	$1-10^{-7}$	100%
30	2	26.5	$1-10^{-8}$	100%

Here the parameter y is the critical value to make sure the LLL fail by 100%. Corresponding to the relaxation factor δ closer to 1 means our Beast bases can withstand a stronger LLL reduction.

5 High Dimensional Cases

In this section, we will study the hard cases in higher dimensions. We extend the 3 dimensional Beasts, keeping their two properties, which is the basis vectors are almost the same length and their angles are almost 60°. At first we observe the performance of LLL reduction in high dimensions. Then we give a rational version of 8 dimensional Beauty basis and a general irrational Beauty for higher dimensional bases. Easily we can build the corresponding Beasts bases from these Beauty bases. According with our conjectures in dimension 3, our experimental results show that the constructed Beast bases are stronger than random lattices to resist LLL reduction.

5.1 Success Probability of LLL in High Dimensions

Similar to the experiments in Sect. 4.1, we also did experiments for higher dimensions until 32. For each dimension, we used 100,000 random bases generated from TU Darmstadt SVP Challenge [20] and we fixed $\delta = 0.99$ in our experiments. From Fig. 3 we can see that the exact-arithmetic LLL can solve SVP by an extremely high success probability within 15 dimensions.

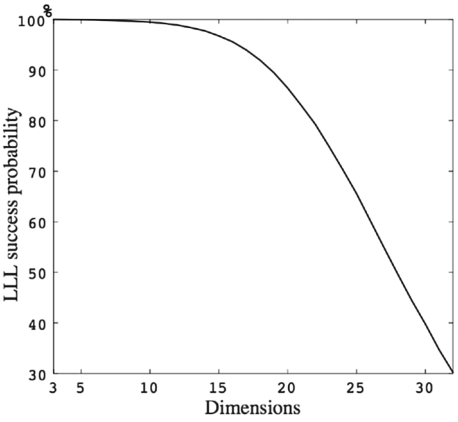

Fig. 3. Experimental results of random bases generated from TU Darmstadt SVP Challenge.

5.2 Build 8 Dimensional Beasts from Rational Beauty Bases

Our defined Beauty bases is a configuration of vectors in the Euclidean space fulfilling certain geometrical properties, namely vectors are sampled from the isometry group. Root system is known as a group somehow satisfying the symmetric property. Observation from the simple roots for root system E_8 in the odd coordinate inspired us to exploit a rational Beauty basis with integral vectors (with elements equal to 1) and components 1/2, see $Beauty_8$ as follows.

$$Beauty_8 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \in \mathbb{Q}^{8\times 8}.$$

Following the 3 dimensional cases, we construct Beast bases as $Beasts_8$. The parameter settings carry forward the conditions (1), (2) and (3). Simultaneously extend $i \in [1, 4]$ to $i \in [1, 22]$ as $\epsilon_1 \geq \epsilon_2 > \epsilon_3 \geq \epsilon_4 \cdots \geq \epsilon_{22}$.

$$Beasts_8 = \begin{pmatrix} P-\epsilon_1 & P-\epsilon_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ P-\epsilon_3 & 0 & P-\epsilon_4 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ P-\epsilon_{13} & 0 & 0 & 0 & 0 & 0 & 0 & P-\epsilon_{14} \\ \lceil \frac{P}{2} \rfloor - \epsilon_{15} & \lceil \frac{P}{2} \rfloor - \epsilon_{16} & \lceil \frac{P}{2} \rfloor - \epsilon_{17} & \lceil \frac{P}{2} \rfloor - \epsilon_{18} & \lceil \frac{P}{2} \rfloor - \epsilon_{19} & \lceil \frac{P}{2} \rfloor - \epsilon_{20} & \lceil \frac{P}{2} \rfloor - \epsilon_{21} & \lceil \frac{P}{2} \rfloor - \epsilon_{22} \end{pmatrix}.$$

5.3 How to Build Beasts in General High Dimensions

Start from 3 dimensional cases, we want to add vector to get a Beauty lattice basis in $\mathbb{R}^4$. By basic calculation, we can easily get,

$$Beauty'_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{5}}{2} \end{pmatrix}.$$

We denote the $Beauty'_n$ to distinguish from the rational bases. Analogously using $Beasts'_n$ to stand the Beasts generated from $Beauty'_n$. By construction, we can get a full-rank Beauty lattice generated by the basis as follows.

$$Beauty'_n = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{\sqrt{5}}{2} & 0 & 0 & \cdots & 0 & \cdots & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2\sqrt{5}} & \sqrt{\frac{6}{5}} & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2\sqrt{5}} & \sqrt{\frac{1}{30}} & \cdots & \frac{1}{\sqrt{i(i-1)}} & \sqrt{\frac{i+1}{i}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2\sqrt{5}} & \sqrt{\frac{1}{30}} & \cdots & \cdots & \cdots & \frac{1}{\sqrt{n(n-1)}} & \sqrt{\frac{n+1}{n}} \end{pmatrix} \in \mathbb{R}^{n \times n}$$

Hence we know now how to build a Beauty lattice which can be easily converted to a hard lattice as the following:

1. Pick a large integer P.
2. Multiply $Beauty_n$ by the integer P and round it to the ceiling integer.
3. Subtract a random matrix $\Sigma = [\epsilon_{ij}]$ where ϵ_{ij} sufficient small integer.

Or we can simply take the form from the following basis. Parameter settings follow the 3 dimensional cases in our main theorem in Sect. 4.2 and $\epsilon_1 \geq \epsilon_2 > \epsilon_3 \geq \epsilon_4 \cdots \geq \epsilon_n$.

$$Beasts'_m = \begin{pmatrix} P-\epsilon_{11} & P-\epsilon_{12} & 0 & 0 & 0 & \cdots & 0 & \cdots 0 \\ 0 & P-\epsilon_{22} & P-\epsilon_{23} & 0 & 0 & \cdots & 0 & \cdots 0 \\ P & 0 & P & 0 & 0 & \cdots & 0 & \cdots 0 \\ \lceil \frac{P}{2} \rfloor - \epsilon_{41} & \lceil \frac{P}{2} \rfloor - \epsilon_{42} & \lceil \frac{P}{2} \rfloor - \epsilon_{43} & \lceil P\frac{\sqrt{5}}{2} \rfloor - \epsilon_{44} & 0 & \cdots & 0 & \cdots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lceil \frac{P}{2} \rfloor - \epsilon_{i1} & \lceil \frac{P}{2} \rfloor - \epsilon_{i2} & \lceil \frac{P}{2} \rfloor - \epsilon_{i3} & \lceil \frac{P}{2\sqrt{5}} \rfloor - \epsilon_{i4} & \cdots & \lceil \frac{P}{\sqrt{i(i-1)}} \rfloor - \epsilon_{i,i-1} & \lceil P\sqrt{\frac{i+1}{i}} \rfloor - \epsilon_{i,i} & \cdots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lceil \frac{P}{2} \rfloor - \epsilon_{n1} & \lceil \frac{P}{2} \rfloor - \epsilon_{n2} & \lceil \frac{P}{2} \rfloor - \epsilon_{n3} & \lceil \frac{P}{2\sqrt{5}} \rfloor - \epsilon_{n4} & \cdots & \cdots & \cdots & \lceil P\sqrt{\frac{n+1}{n}} \rfloor - \epsilon_{n,n} \end{pmatrix}$$

5.4 Experimental Results of High Dimensional Cases

As same as the procedure in Sect. 4.3, we generate 100,000 bases for each dimension and randomize them by equal amount of unimodular matrices from Magma [12]. The bit length x of P is also set as $x = 10 \cdot n$. The parameter settings

Table 2. Some experimental results of higher dimensional Beast bases. Parameter settings $x = 10 \cdot n$, $z = log_2(1-\delta)^{-1}$ and y are under the conditions in Theorem 1.

Dim n	LLL failure prob. on the random bases			LLL failure prob. on the $Beasts'_n$		
	Ex1. $\delta = 1 - 10^{-1}$	Ex2. $\delta = 1 - 10^{-2}$	Ex3. $\delta = 1 - 10^{-3}$	Ex1. $\delta = 1 - 10^{-1}$	Ex2. $\delta = 1 - 10^{-2}$	Ex3. $\delta = 1 - 10^{-3}$
4	0.057%	0.035%	0.022%	30.3%	30.3%	30.3%
5	0.129%	0.068%	0.048%	39.4%	39.4%	39.4%
6	0.240%	0.107%	0.103%	40.5%	40.5%	40.5%
7	0.350%	0.173%	0.157%	43.5%	43.5%	43.5%
8	0.542%	0.256%	0.229%	47.2%	47.2%	47.2%
9	0.79%	0.400%	0.359%	50.7%	50.7%	50.7%
10	1.14%	0.507%	0.485%	53.2%	53.2%	53.2%

fulfil the above sections respectively. Note that for all cases we perform ENUM algorithm to check if the shortest one is in the reduced bases or not.

At first, the LLL ($\delta \in [0.9, 1)$) failure probability solving SVP on $Beasts_8$ expanded lattices is stably 48.4%. However for the same magnitudes of vectors in TU Darmstadt SVP Challenge generated random lattices, the LLL failure probability is $\leq 0.542\%$, see Table 2.

Simultaneously Table 2 also shows that the $Beasts'_n$ bases constructed from irrational $Beauty'_n$ possess a comparative high probability to withstand exact-arithmetic version of LLL reduction with relaxation factor $\delta \in [0.9, 1)$. Moreover the LLL failure probability keep stable with the increasing of δ, which means LLL becomes more powerful. Especially the close result for $Beasts_8$ (48.4%) and $Beasts'_8$ (47.2%) give a support for our conjecture of Beasts bases.

6 Conclusions

We studied the hard cases in LLL reduction, as we called them "Beasts", whose vectors fulfilling two properties: 1. the vectors are almost the same length; 2. the angles of the vectors are almost 60°. They are generated from the corresponding "Beauty" bases "exactly" instead of "almost". Our experimental results showed the randomized Beasts bases have a potentiality to withstand the LLL reduction. We started from 3 dimensional cases and gave a complete theoretical proof for this phenomenon. We also give a 8 dimensional rational Beauty and the general irrational Beauty for dimensions higher than 3. Comparing with the same magnitude of TU Darmstadt SVP Challenge random bases, our generated Beasts can stably withstand the exact-arithmetic LLL with relaxation factor $\delta \in [0.9, 1)$. However, we can't assert categorically that the Beasts are only our constructions. So it is an intriguing open problem if there are other hard cases in LLL reduction.

A Supplementary proof in main theory

Let $Beasts_3 = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)^t$. We prove that $\mathbf{b_1}$ is the shortest vector in $L(Beasts_3)$, i.e. $\|\mathbf{b}_1\| = (\lambda_1(L)) \leq \|a\mathbf{b}_1 + b\mathbf{b}_2 + c\mathbf{b}_3\|$ for any coefficients set $(a, b, c) \in \mathbb{Z}^3 \backslash \{(0, 0, 0)\}$. We observe any non-zero vector $\mathbf{b} = a\mathbf{b}_1 + b\mathbf{b}_2 + c\mathbf{b}_3$ in $L(Beasts_3)$. The square norm of $\mathbf{b}$ is as

$$\begin{aligned}
\|\mathbf{b}\|^2 &= \|a\mathbf{b}_1 + b\mathbf{b}_2 + c\mathbf{b}_3\|^2 \\
&= [a(P-\epsilon_1) + cP]^2 + [a(P-\epsilon_2) + b(P-\epsilon_3)]^2 + [b(P-\epsilon_4) + cP]^2 \\
&\geq (a(P-\epsilon_1) + c(P-\epsilon_1))^2 + [a(P-\epsilon_2) + b(P-\epsilon_2)]^2 + [b(P-\epsilon_1) + c(P-\epsilon_1)]^2 \\
&= [(a+b)^2 + (b+c)^2](P-\epsilon_1)^2 + (a+b)^2(P-\epsilon_2)^2.
\end{aligned}$$

Since a, b, and c are integers and not equal to 0 at the same time, the smallest case for the right side of inequality is when

$$a = b = 0 \text{ and } |c| = 1.$$

However in this case $\|\mathbf{b}\| = \|\mathbf{b}_3\| > \|\mathbf{b}_1\|$ satisfying our theorem. So we consider a "looser" condition for the right side as

$$a + b = 0 \text{ and } |c| \in \mathbb{Z}^+_{>1}.$$

From it we derive

$$\begin{aligned}
\|\mathbf{b}\|^2 &\geq [(c-a)^2 + (c+a)^2](P-\epsilon_1)^2 \\
&\geq 4(P-\epsilon_1)^2 \\
&= (P-\epsilon_1)^2 + (P-\epsilon_2)^2 + 3(P-\epsilon_1)^2 - (P-\epsilon_2)^2 \\
&= \|\mathbf{b}_1\|^2 + 2P(P - 3\epsilon_1 + \epsilon_2) + 3\epsilon_1^2 - \epsilon_2^2.
\end{aligned}$$

From the inequality (4) we get $P - 3\epsilon_1 > 0$. Hence

$$\begin{aligned}
\|\mathbf{b}\|^2 &\geq \|\mathbf{b}_1\|^2 + 2P(P - 3\epsilon_1 + \epsilon_2) + 3\epsilon_1^2 - \epsilon_2^2 \\
&\geq \|\mathbf{b}_1\|^2
\end{aligned}$$

for $\epsilon_1 \geq \epsilon_2$. We finish the proof of $\|\mathbf{b}_1\| = \lambda_1(L)$.

References

1. Akhavi, A.: Worst-case complexity of the optimal LLL algorithm. In: Gonnet, G.H., Viola, A. (eds.) LATIN 2000. LNCS, vol. 1776, pp. 355–366. Springer, Heidelberg (2000). doi:10.1007/10719839_35
2. Akhavi, A.: The optimal LLL algorithm is still polynomial in fixed dimension. Theor. Comput. Sci. **297**(1–3), 323 (2003)
3. Bi, J., Coron, J.-S., Faugère, J.-C., Nguyen, P.Q., Renault, G., Zeitoun, R.: Rounding and chaining LLL: finding faster small roots of univariate polynomial congruences. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 185–202. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54631-0_11
4. Goldstein, D., Mayer, A.: On the equidistribution of Hecke points. Forum Mathematicum **15**(2), 165–189 (2003)

5. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78967-3_3
6. Gama, N., Nguyen, P.Q., Regev, O.: Lattice enumeration using extreme pruning. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 257–278. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_13
7. Lagrange, L.: "Recherches d'arithmétique". Nouv. Mém. Acad. (1773)
8. Luzzi, L., Othman, G.R., Belfiore, J.C.: Augmented lattice reduction for MIMO decoding. IEEE Trans. Wireless Commun. **9**(9), 2853–2859 (2010)
9. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
10. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13190-5_1
11. Luzzi, L., Stehlé, D., Ling, C.: Decoding by embedding: correct decoding radius and DMT optimality. IEEE Trans. Inf. Theory **59**(5), 2960–2973 (2013)
12. Magma computational algebra system. http://magma.maths.usyd.edu.au/magma/
13. Minkowski, H.: Geometrie der Zahlen (1910)
14. Nguyen, P.Q., Stehlé, D.: Low-dimensional lattice basis reduction revisited. ACM Trans. Algorithms **5**(4) (2009)
15. Victor Shoup's NTL library. http://www.shoup.net/ntl/
16. Nguyen, P.Q., Vallée, B. (eds.): The LLL Algorithm - Survey and Applications. Information Security and Cryptography. Springer, Berlin Heidelberg (2010)
17. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93 (2005)
18. Semaev, I.: A 3-dimensional lattice reduction algorithm. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 181–193. Springer, Heidelberg (2001). doi:10.1007/3-540-44670-2_13
19. Schnorr, C.P., Euchner, M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math. Program. **66**(1–3), 181–199 (1994)
20. TU Darmstadt lattice challenge. http://www.latticechallenge.org/

http://www.springer.com/978-3-319-64199-7

Advances in Information and Computer Security
12th International Workshop on Security, IWSEC 2017,
Hiroshima, Japan, August 30 – September 1, 2017,
Proceedings
Obana, S.; Chida, K. (Eds.)
2017, XII, 243 p. 52 illus., Softcover
ISBN: 978-3-319-64199-7