Probabilistic Analysis of LLL Reduced Bases

Michael Schneider, Johannes Buchmann, and Richard Lindner

Technische Universität Darmstadt, Department of Computer Science
Hochschulstraße 10, 64289 Darmstadt, Germany
mischnei@cdc.informatik.tu-darmstadt.de

Abstract. Lattice reduction algorithms behave much better in practice than their theoretical analysis predicts, with respect to output quality and runtime. In this paper we present a probabilistic analysis that proves an average case bound for the length of the first basis vector of an LLL reduced bases which reflects LLL experiments much better.
Keywords: lattice reduction, LLL, worst case bounds

1 Introduction

Lattice reduction is a useful tool in cryptanalysis. Different cryptosystems are broken using lattice reduction, e.g. knapsack systems [LO85,CJL+92] as well as RSA in special settings [May07]. Further on, factoring composite numbers and computing discrete logarithms is possible using lattice reduction [Sch91,May07].

The most famous algorithm for lattice reduction is the LLL algorithm by Lenstra, Lenstra and Lovász [LLL82]. Every lattice reduction algorithm used today is in some sense a variant of the LLL. Theoretically, the best algorithm to find short vectors is the *slide reduction* algorithm [GN08a]. Practically, the most promising algorithms are the L^2 algorithm by Nguyen and Stehlé [NS05] and the BKZ algorithm by Schnorr [SE94]. A comparison of lattice reduction algorithms can be found in [BLR08].

In this paper we present an average analysis that predicts the expected value of the first basis vector of an LLL reduced basis to be at most $1.0439^{(n-1)} \cdot |\det L|^{1/n}$ whereas the worst case analysis only yields the bound $1.078^n \cdot \det(L)^{1/n}$. To obtain this result, we assume that the Gram-Schmidt coefficients that arise in lattice reduction are random variables. The distribution is a polynomial of degree four. This distribution is deduced from experiments that we performed on random lattices chosen as in [GN08b,NS06] and on modular lattices like those of [BLR08]. Our new bound reflects the practical results of lattice reduction far better than the existing worst case bound and will be helpful in estimating cryptographic key sizes of lattice based systems.

First approaches concerning the gap between theory and practice were made in [NS06] and [GN08b]. Both papers analyse the practical behaviour of reduction algorithms by evaluating their experiments.

2 Preliminaries

Let $n, d \in \mathbb{N}$, $n \leq d$, $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^d$ linearly independent. Then $L(\mathbf{B}) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$ is the lattice spanned by $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. $L(\mathbf{B})$ has

Dagstuhl Seminar Proceedings 09221
Algorithms and Number Theory
http://drops.dagstuhl.de/opus/volltexte/2009/2126

dimension n, $\mathbf{B}$ is a basis of a lattice. Such a basis is uniquely determined up to unimodular transformations. We write L instead of $L(\mathbf{B})$ if it is clear which basis is concerned. The first successive minimum $\lambda_1(L)$ is the length of the shortest vector of a lattice. The lattice determinant $\det(L(\mathbf{B}))$ is defined as $\sqrt{\det(\mathbf{B}\mathbf{B}^t)}$. It is invariant under basis changes. For full-dimensional lattices ($n = d$) there is $\det(L(\mathbf{B})) = |\det(\mathbf{B})|$ for every basis $\mathbf{B}$.

Denote the Gram-Schmidt-orthogonalization (GSO) with $\mathbf{b}_i^* = \pi_i(\mathbf{b}_i)$ where $\pi_i(\mathbf{b}) \to \text{span}\,(\mathbf{b}_1 \dots \mathbf{b}_{i-1})^\perp$ is the orthogonal projection. The GSO is calculated via $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$ where $\mu_{i,j} = \mathbf{b}_i^T\mathbf{b}_j^* / \left\|\mathbf{b}_j^*\right\|^2$ for all $1 \le j \le i \le n$. We know that $\prod_{k=1}^{n} \|\mathbf{b}_k^*\| = |\det L|$.

Lattice reduction. Creating a basis consisting of short and nearly orthogonal vectors is the goal of lattice reduction. A more detailed notion of a reduced lattice is the following. A basis $\mathbf{b}_1, \dots, \mathbf{b}_n$ is called *LLL-reduced* with $\delta \in (\frac{1}{4}, 1]$, if $|\mu_{i,j}| \le 0.5$ for $1 \le j < i \le n$ and $\delta\|\mathbf{b}_{i-1}^*\|^2 \le \|\mathbf{b}_i^*\|^2 + \mu_{i,i-1}^2\|\mathbf{b}_{i-1}^*\|^2$ for $i = 2, \dots, n$ [LLL82].

Hard lattice problems. There are several problems on lattices that are supposed to be or proven to be hard [MG02]. The most famous problem is the shortest vector problem (SVP). The goal of γ-SVP is to find an (approximate) shortest non-zero vector in the lattice, namely a vector $\mathbf{0} \neq \mathbf{v} \in L$ with $\|\mathbf{v}\| \le \gamma\lambda_1(L)$, where $\gamma \ge 1$ is the approximation factor. It is possible to formulate the problem in every norm, the most usual norm is the euclidean norm, that we are using throughout this paper.

As the length of the shortest vector $\lambda_1(L)$ might not be known, it might be hard to control the approximation factor of SVP. Therefore it is common practice to use the Hermite-SVP variant: given a $\gamma \ge 1$, find a non-zero vector $\mathbf{v} \in L$ with $\|\mathbf{v}\| \le \gamma \cdot (\det L)^{1/n}$. Having reduced a basis $\mathbf{B}$ one can easily calculate the reached Hermite factor using $\gamma_{\text{Hermite}} = \|\mathbf{b}_{\min}\| / (\det L)^{1/n}$.

The γ-SVP was solved by Lenstra, Lenstra and Lovász in [LLL82] for factors γ exponential in the lattice dimension n. Their LLL algorithm requires $\mathcal{O}(n^3 \log M)$ arithmetic operations (M is a function of the input size, i.e. $M = \max_{i=1,\dots,n}(\|\mathbf{b}_i\|^2, D_i)$ and $D_i = \det(L(\mathbf{b}_1, \dots, \mathbf{b}_i))^2$) and outputs a basis whose first vector approximates the shortest lattice vector with an approximation factor exponential in the lattice dimension. More concretely, it can be proved that $\|\mathbf{b}_1\| \le (4/3)^{(n-1)/4} \cdot \det(L)^{1/n}$ [LLL82].

In [SE94] the authors introduce the idea of deep inserting the size-reduced vector into the basis. The same paper presents the BKZ algorithm, that is a blockwise variant of the LLL algorithm. BKZ is today's best algorithm for lattice reduction in practice. Using blocksize β it reaches a lattice vector with length $\|\mathbf{b}_1\| \le (\gamma_\beta)^{\frac{n-1}{\beta-1}} \cdot \lambda_1$ [Sch94], where γ_β is the Hermite constant in dimension β.

Practical behaviour. In practice however, lattice reduction algorithms behave much better than theory would suppose: in the average case they find much shorter vectors than theoretical worst case bounds suggest. In [GN08b] Gama

and Nguyen give a practical analysis using the established implementation of Shoup's NTL library [Sho]. The authors state that a Hermite factor of 1.01^n and an approximation factor of 1.02^n in high lattice dimension (e.g. dimension 500) is within reach today, but a Hermite factor of 1.005^n in dimension around 500 is totally out of reach.

3 LLL on the Average Revisited

Remember that in an LLL reduced basis it is required that $|\mu_{i,j}| \leq 0.5$ for $0 \leq j < i \leq n$. For the theoretical LLL analysis one assumes [LLL82] that all $\mu_{i,i-1}$ match the worst case, i.e. $|\mu_{i,i-1}| = 0.5$. Our new idea is to replace this assumption by a more realistic, probabilistic distribution of the Gram-Schmidt coefficients. The challenge is to find a suitable distribution function of those random variables. In [NS06] the authors state that the coefficients $\mu_{i,i-1}$ are not uniformly distributed in the area $[-0.5, 0.5]$.

In order to get a better impression of the distribution of the $\mu_{i,i-1}$ we performed some experiments on random lattices like those used in [NS06] and [GN08b] as well as on the modular lattices of [BLR08]. For all experiments we used a parameter $\delta = 0.99$. It turns out that the values $\mu_{i,i-1}$ are distributed along a polynomial of degree four, namely $p(x) = 53.85692x^4 + 1.57202x^2 + 0.19579$ in the range $[-0.5, 0.5]$. Figure 1 shows the experimental data and the fitting polynomial $p(x)$. The polynomial $p(x)$ is created such that $\int_{-\infty}^{\infty} p(x)\,dx = 1$, which allows us to use $p(x)$ as density function of a probability distribution.

3.1 Expectation Values of $\|\mathbf{b_1}\|$

We are now using the probability distribution given by $p(x)$ to give a better bound on LLL reduced bases. The first part of the proof is quite similar to the analysis of the LLL bound [LLL82].

Theorem 1. *Suppose that a basis $[\mathbf{b}_1 \dots \mathbf{b}_n]$ is chosen arbitrarily and an LLL algorithm is performed on the basis. Suppose that after LLL-reduction, the $\mu_{i,i-1}$ are independent random variables and their probability distribution is given by the polynomial $p(x)$. Then the expectation of the norm of the first lattice vector after LLL reduction is*

$$E(\|\mathbf{b}_1\|) \leq 1.0439^{(n-1)} \cdot |\det L|^{1/n}\,. \tag{1}$$

Proof. We start with an LLL-reduced basis $\mathbf{B}$: $\delta \left\|\mathbf{b}^*_{i-1}\right\|^2 \leq \mu^2_{i,i-1} \left\|\mathbf{b}^*_{i-1}\right\|^2 + \|\mathbf{b}^*_i\|^2 \,\forall i = 2, \dots, n$. With that[1] $\left\|\mathbf{b}^*_{i-1}\right\|^2 \leq (\delta - \mu^2_{i,i-1})^{-1} \|\mathbf{b}^*_i\|^2 \,\forall\, i = 2, \dots, n$. Repeating this gives us

$$\|\mathbf{b}^*_1\|^2 \leq \prod_{i=1}^{k-1} (\delta - \mu^2_{i,i-1})^{-1} \|\mathbf{b}^*_k\|^2 \quad \forall\, k = 1, \dots, n\,.$$

[1] $\delta - \mu^2_{i,i-1}$ is positive since $|\mu_{i,i-1}| \leq 0.5$ and $\delta > 0.25$

Multiplying both sides for $k = 1, \ldots, n$ leads to

$$\|\mathbf{b}_1^*\|^{2n} \leq \prod_{k=1}^{n} \big(\prod_{i=1}^{k-1} (\delta - \mu_{i,i-1}^2)^{-1} \big) \|\mathbf{b}_k^*\|^2 = \prod_{k=1}^{n} \big(\prod_{i=1}^{k-1} (\delta - \mu_{i,i-1}^2)^{-1} \big) \cdot \underbrace{\prod_{k=1}^{n} \|\mathbf{b}_k^*\|^2}_{=|\det L|^2} .$$

Calculating the logarithm we get

$$\begin{aligned} \ln(\|\mathbf{b}_1^*\|^{2n}) &\leq \ln\Big(\prod_{k=1}^{n} \big(\prod_{i=1}^{k-1} (\delta - \mu_{i,i-1}^2)^{-1} \big) \Big) + \ln(|\det L|^2) \\ &= -\sum_{k=1}^{n} \sum_{i=1}^{k-1} \ln(\delta - \mu_{i,i-1}^2) + 2\ln(|\det L|) \\ \Leftrightarrow \quad \ln(\|\mathbf{b}_1^*\|) &\leq -\frac{1}{2n} \sum_{k=1}^{n} \sum_{i=1}^{k-1} \ln(\delta - \mu_{i,i-1}^2) + \frac{1}{n} \ln(|\det L|) \end{aligned}$$

We now calculate the expectation value:

$$E(\ln(\|\mathbf{b}_1^*\|) \leq -\frac{1}{2n} \sum_{k=1}^{n} \sum_{i=1}^{k-1} E(\ln(\delta - \mu_{i,i-1}^2)) + \frac{1}{n} \ln(|\det L|)$$

Using the polynomial $p(x)$ as density function for the random $\mu_{i,i-1}$ we get

$$E(\ln(\delta - \mu_{i,i-1}^2)) = \int_{-\infty}^{\infty} \ln(\delta - x^2) p(x)\, dx = 2 \int_{0}^{1/2} \ln(\delta - x^2) p(x)\, dx$$

For $\delta = 1$, the expectation value $E(\ln(\delta - x^2))$ becomes -0.172 (c.f. full version). This leads to the following:

$$E(\ln(\|\mathbf{b}_1^*\|) \leq \frac{0.172}{2n} \frac{n(n-1)}{2} + \frac{1}{n} \ln(|\det L|) = 0.043(n-1) + \frac{1}{n} \ln(|\det L|)\,.$$

This leads to

$$\|\mathbf{b}_1\| \approx \exp(E(\ln(\|\mathbf{b}_1^*\|)) = \exp(0.043)^{(n-1)} \cdot |\det L|^{1/n} = 1.0439^{(n-1)} \cdot |\det L|^{1/n}\,.$$

□

The original LLL worst case bound for $\delta = 1$ is

$$\|\mathbf{b}_1\| \leq (4/3)^{(n-1)/4} \cdot \det(L)^{1/n} \approx 1.078^n \cdot \det(L)^{1/n}\,. \tag{2}$$

There exist bases for whom this bound is tight. In [NS06] Nguyen and Stehlé show experimentally that practical L^3 algorithms reach an average value of

$$\|\mathbf{b}_1\| \approx 1.02^n \cdot \det(L)^{1/n}\,. \tag{3}$$

The experiments of Gama and Nguyen [GN08b] show the same behaviour of LLL algorithms. Figure 2 gives an illustration of these norm-values, i.e. the Figure shows the first part and leaves out the $\det(L)^{1/n}$-part. It is easy to see that our expectation value is much closer to the average case than the worst case bound.

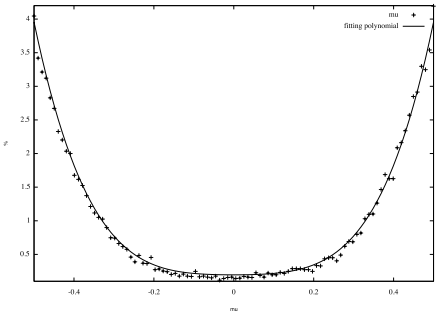

Fig. 1. Distribution of the values $\mu_{i,i-1}$ and the fitting polynomial $p(x)$

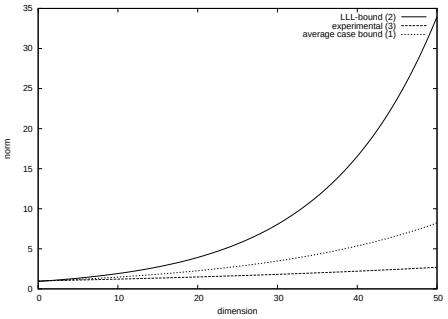

Fig. 2. Comparison of the norm bounds: LLL worst case bound (2), experimental value (3) of [NS06] and new average case bound (1). The $\det(L)^{1/n}$-part is omitted.

4 Further Work

To our knowledge there is no further theoretical analysis of the deep insertion variant [SE94] concerning worst case bounds. The best upper bound known is the standard LLL-bound. Practically the algorithm was observed in [BW02,NS06] and [GN08b]. The average Hermite factor reached by Deep-LLL is observed to be 1.012^n (maximal insertion depth not given) in [NS06] and 1.011^n with maximal insertion depth 50 in [GN08b], respectively.

It might be possible to apply our probabilistic analysis to the Deep-LLL and to the BKZ algorithm in order to prove bounds for the deep insertion variant or improve the known BKZ bound.

It remains an open problem to show a more precise analysis of the Gram-Schmidt coefficients after LLL reduction. The description of our experiments in this abstract is quite short and has to be extended. The polynomial distribution that we assumed can only be seen as an approximation. The main difficulty in this analysis is the fact that the random variables $\mu_{i,j}$ are dependent on each other, i.e. $\mu_{i,i-1}$ and $\mu_{i+1,i}$ both depend on $\mathbf{b}_i$ and can therefore not be considered to be independent random variables.

References

[BLR08] Johannes Buchmann, Richard Lindner, and Markus Rückert. Explicit hard instances of the shortest vector problem. In *Post-Quantum Cryptography (PQCrypto) 2008*, Lecture Notes in Computer Science, pages 79–94. Springer-Verlag, 2008.

[BW02] Werner Backes and Susanne Wetzel. Heuristics on lattice basis reduction in practice. *ACM Journal of Experimental Algorithmics*, 7, 2002.

[CJL+92] Matthijs J. Coster, Antoine Joux, Brian A. LaMacchia, Andrew M. Odlyzko, Claus-Peter Schnorr, and Jacques Stern. Improved low-density subset sum algorithms. *Computational Complexity*, 2:111–128, 1992.

[GN08a] Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within mordell's inequality. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2008*, pages 207–216. ACM Press, 2008.

[GN08b] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *Advances in Cryptology — Eurocrypt 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer-Verlag, 2008.

[LLL82] Arjen Lenstra, Hendrik Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 4:515–534, 1982.

[LO85] J. C. Lagarias and Andrew M. Odlyzko. Solving low-density subset sum problems. *J. ACM*, 32(1):229–246, 1985.

[May07] Alexander May. Using LLL-reduction for solving RSA and factorization problems, 2007. A survey for the LLL+25 conference.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[NS05] Phong Q. Nguyen and Damien Stehlé. Floating-point LLL revisited. In *Advances in Cryptology — Eurocrypt 2005*, pages 215–233, 2005.

[NS06] Phong Q. Nguyen and Damien Stehlé. LLL on the average. In *Algorithmic Number Theory Symposium — ANTS*, pages 238–256, 2006.

[Sch91] Claus-Peter Schnorr. Factoring integers and computing discrete logarithms via diophantine approximations. In *EUROCRYPT*, pages 281–293, 1991.

[Sch94] Claus-Peter Schnorr. Block reduced lattice bases and successive minima. *Combinatorics, Probability & Computing*, 3:507–522, 1994.

[SE94] Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.

[Sho] Victor Shoup. Number theory library (NTL) for C++. `http://www.shoup.net/ntl/`.