Effective LLL Reduction for Lattice Decoding

Cong Ling  
Department of Electrical and Electronic Engineering  
Imperial College London  
London, United Kingdom  
E-mail: cling@ieee.org

Nick Howgrave-Graham  
NTRU Cryptosystems  
5 Burlington Woods  
Burlington, MA 02144, USA  
E-mail: nhowgravegraham@ntru.com

Abstract—The use of Lenstra-Lenstra-Lovasz (LLL) lattice reduction significantly improves the performance of zero-forcing (ZF) and successive interference cancellation (SIC) decoders in multi-input multi-output (MIMO) communications. Capitalizing on the observation that the decision region of SIC is determined by the Gram-Schmidt vectors rather than the basis itself, we propose the use of effective LLL reduction in SIC decoding, where size reduction is only performed for pairs of consecutive basis vectors. We establish the theoretical upper bound $O(n^4 \log n)$ on the average complexity of effective LLL reduction for the i.i.d. Gaussian model of MIMO fading channels, which is an order lower than previously thought. Moreover, an effectively LLL-reduced basis can easily be transformed into the standard LLL-reduced basis for the purpose of ZF decoding.

I. INTRODUCTION

In several problems of digital communications, the outputs are a linear combination of the inputs corrupted by additive noise. Let $x = (x_1, \ldots, x_n)^T$ be the $n \times 1$ data vector, where each symbol $x_i$ is chosen from a finite subset of the integer set $\mathbb{Z}$. With scaling and shifting, we have the generic $n \times n$ linear channel model

$$y = Bx + n$$

where $y, n \in \mathbb{R}^n$ denote the channel output and noise vectors, respectively, and $B \in \mathbb{R}^{n \times n}$ is the $n \times n$ full-rank matrix of channel coefficients.

For moderate to large values of $n$, decoding for such multi-input multi-output (MIMO) systems represents a challenging problem in communication theory. Recently, the theory of lattices and lattice reduction is emerging as a powerful tool for the ranges of $n$ in MIMO communications. In this paper, we shall develop strategies that have been extensively studied in literature. The most basic version of lattice decoding, with a provable complexity bound that is an order lower than that of the LLL algorithm. This is because a weaker version of LLL reduction without full size reduction is sufficient for SIC decoding.

For integral bases, it is well known that the LLL algorithm has complexity bound $O(n^4)$ [8]. The complexity of the LLL algorithm for real-valued bases is far less known. To our knowledge, the only complexity analysis of the real-valued LLL algorithm for a probabilistic model is due to Daude and Vallee [11]. However, they assumed basis vectors are drawn independently from the unit ball of $\mathbb{R}^n$, which does not hold in MIMO communications. In this paper, we shall develop an analysis for the typical MIMO channel model where the entries of $B$ are assumed to be i.i.d. standard normal.

Using effective LLL reduction may raise the concern of numerical stability, as the Gram-Schmidt coefficients will grow. However, with the accuracy present in most floating-point implementations, this does not seem to cause a problem for the ranges of $n$ in MIMO communications. A serious study on this issue (e.g., following [12]) is a topic of future research.

Notation: Let $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ be the inner product and Euclidean length in $\mathbb{R}^n$. $[x]$ denotes the integer closest to $x$. $(\cdot)^T$ and $(\cdot)^{-1}$ denote the matrix transpose and inverse, respectively.

II. APPROXIMATE LATTICE DECODING

For i.i.d. Gaussian noise, the ML decoder for the MIMO channel (1) is given by

$$\hat{x} = \arg\min_{x \in \mathcal{C}} \|y - Bx\|^2$$

where $\mathcal{C}$ stands for the transmitter codebook. Due to the exponential complexity of standard ML decoding, suboptimal strategies have been extensively studied in literature. The most obvious one is the linear ZF decoder $\hat{x} = [B^{-1}y]$ where the interference is totally suppressed. This, however, leads to
the detrimental effect of noise amplification. The effect can be alleviated by using SIC (also known as decision-feedback detection). Performing the QR decomposition $B = QR$, where $Q$ is an orthonormal matrix and $R$ is an upper triangular matrix, and multiplying (1) on the left with $Q^T$ we have $y^\prime = Q^T y = R x + n^\prime$. Since $R$ is triangular, the last symbol $x_N$ can be estimated as $\hat{x}_N = \lfloor y^\prime_N / r_{N,N} \rfloor$. Then the estimate is substituted to remove the interference term in $y^\prime_{N-1}$ when $x_{N-1}$ is being estimated. The procedure is continued until the first symbol is detected. That is, we have the following recursion:

$$\hat{x}_n = \left[ \frac{y^\prime_n - \sum_{j=n+1}^N r_{n,j} \hat{x}_j}{r_{n,n}} \right]$$

for $n = N, N-1, \ldots, 1$. For convenience, we shall deal with SIC with the ZF criterion only in this paper, although the minimum mean-square error (MMSE) criterion will further enhance the performance.

The MIMO channel model (1) admits a lattice formulation. A lattice in $\mathbb{R}^n$ is the set of all integer linear combinations of $n$ linearly independent vectors $b_1, \ldots, b_n$ [13]:

$$L \triangleq L(B) = \left\{ \sum_{i=1}^n x_i b_i \mid x_i \in \mathbb{Z} \right\}$$

where $B = [b_1, \ldots, b_n]$ is called the basis of $L$.

The basic idea behind approximate lattice decoding is to use lattice reduction in conjunction with conventional ZF and SIC decoders. With lattice reduction, the basis $B$ is transformed into a new basis consisting of near-orthogonal vectors $B' = BU$ where $U$ is a unimodular matrix, i.e., $U$ contains only integer entries and the determinant $\det U = \pm 1$. Indeed, we have the equivalent channel model

$$y = B' U^{-1} x + n = B' x' + n, \quad x' = U^{-1} x.$$

Then conventional decoders are applied on the reduced basis. The estimate $\hat{x}'$ is then transformed back into $\hat{x} = U \hat{x}'$. Since the equivalent channel is highly likely to be well-conditioned, the effect of noise amplification will be minimized.

We examine the decision regions of the respective decoders to characterize the error performance [9]. Fig. 1 illustrates the decision regions in a 2-D lattice. More generally, the decision region of ZF or SIC contains $2^n$ facets and we can define $n$ distances due to symmetry. We are particularly interested in the minimum distance, because it dominates the performance at high SNR.

The decision region of the ZF decoder is a fundamental parallelogram centered at the transmitted codeword. The minimum distance is given by $d_{ZF} = \min_{1 \leq i \leq n} \frac{1}{2} \| b_i \| \sin \theta_i$, where $\theta_i$ is the angle between $b_i$ and the linear space spanned by the other $n-1$ basis vectors. Meanwhile, the decision region of SIC is specified by the Gram-Schmidt (GS) vectors. We can compute the GS orthogonalization (O) by the recursion

$$\hat{b}_1 = b_1, \quad \hat{b}_i = b_i - \sum_{j=1}^{i-1} \mu_{i,j} \hat{b}_j, \quad \text{for } i = 2, \ldots, n$$

where $\mu_{i,j} = \langle b_i, \hat{b}_j \rangle / \| \hat{b}_j \|^2$. In matrix notation, GSO can be written as $B = \hat{B} [\mu_{i,j}]^T$, where $\hat{B} = [\hat{b}_1, \ldots, \hat{b}_n]$, and $[\mu_{i,j}]$ is a lower-triangular matrix with unit diagonal elements. In relation to the QR decomposition, one has $\mu_{i,i} = r_{i,i} / r_{i,i}$ and $\hat{b}_i = r_{i,i} \cdot q_i$ where $q_i$ is the $i$-th column of $Q$. The minimum distance of the SIC is $d_{SIC} = \min_{1 \leq i \leq n} \frac{1}{2} \| \hat{b}_i \|$. It is easy to see $d_{ZF} \leq d_{SIC}$. Namely, SIC always has better performance than ZF.

Obviously, the decision region of the ML decoder has minimum distance $d_{ML} = \lambda (L)/2$, where $\lambda (L)$ is the length of the shortest vector of $L$. Reference [9] defined the proximity factors to measure the proximity between the distances of ML decoding and approximate lattice decoding:

$$\rho_{ZF} = \sup_{B \text{ reduced}} \frac{d_{2}^2}{d_{ZF}^2}, \quad \rho_{ML} = \sup_{B \text{ reduced}} \frac{\lambda^2 (L)}{d_{ML}^2}, \quad \rho_{SIC} = \sup_{B \text{ reduced}} \frac{d_{2}^2}{d_{SIC}^2}$$

where the supremum is taken over all bases $B$ satisfying a certain reduction criterion for any $n$-D lattices $L$. $\rho_{ZF}$ and $\rho_{SIC}$ quantify the worst-case loss of approximate lattice decoding in the minimum squared Euclidean distance. With lattice reduction they are upper-bounded by constants that are functions of $n$ alone. Good lattice reduction algorithms lead to small constants.

### III. Effective LLL Reduction

A basis is said to be LLL reduced if the following conditions hold [8]:

$$|\mu_{i,j}| \leq 1/2, \quad 1 \leq j < i \leq n;$$

$$\| \hat{b}_i + \mu_{i,i-1} \hat{b}_{i-1} \|^2 \geq \delta \| \hat{b}_{i-1} \|^2, \quad 1 < i \leq n.$$

The first condition is the size-reduced condition, while the second is known as the Lovasz condition. A size-reduced basis can be obtained by reducing each vector individually. The vector $b_k$ is size-reduced if $|\mu_{k,l}| \leq 1/2$ for all $l < k$. The subroutine shows how $b_k$ is size-reduced against $b_l$ ($l < k$). To size-reduce $b_k$, we call the subroutine for $l = k - 1$ down...
to 1. Size-reducing \(b_k\) does not affect the size reduction of the other vectors. Furthermore, it is not difficult to see that size reduction does not change the GS vectors.

It follows from the Lovasz condition (8) that for an LLL-reduced basis

\[
\|\hat{b}_i\|^2 \geq (\delta - \mu_{i,i-1}^2)\|\hat{b}_{i-1}\|^2 \geq (\delta - 1/4)\|\hat{b}_{i-1}\|^2, \quad (9)
\]
i.e., the lengths of GS vectors never drop too much.

Algorithm 1 describes the LLL algorithm as in [8]. It computes a reduced basis by performing size reduction and swapping in an iterative manner. If the Lovasz condition (8) is violated, the basis vectors \(b_k\) and \(b_{k-1}\) are swapped; otherwise it carries out size reduction to satisfy (7). The algorithm is known to terminate in a finite number of iterations for any given input basis \(B\). By an iteration we mean the operations within the “while” loop in Algorithm 1, which correspond to an increment or decrement of the variable \(k\).

**Subroutine** Pairwise Size Reduction

**Input:** Basis vectors \(b_k\) and \(b_l\) \((l < k)\)

GSO coefficient matrix \([\mu_{i,j}]\)

**Output:** Basis vector \(b_k\) size-reduced against \(b_l\)

Updated GSO coefficient matrix \([\mu_{i,j}]\)

if \(|\mu_{k,l}| > \frac{1}{2}\) then

\(b_k := b_k - [\mu_{k,l}]b_l\)

for \(j = 1, 2, ..., l - 1\) do

\(\mu_{k,j} := \mu_{k,j} - [\mu_{k,l}]\mu_{l,j}\)

\(\mu_{k,l} := \mu_{k,l} - [\mu_{k,l}]\)

**Algorithm 1** LLL Algorithm

**Input:** A basis \(B = [b_1, ..., b_n]\)

**Output:** The LLL-reduced basis

1: compute GSO \(\hat{B}[\mu_{i,j}]^T\)

2: \(k := 2\)

3: while \(k \leq n\) do

4: size-reduce \(b_k\) against \(b_{k-1}\)

5: if \(\|\hat{b}_k + \mu_{k,k-1}\hat{b}_{k-1}\|^2 < \delta\|\hat{b}_{k-1}\|^2\) then

6: swap \(b_k\) and \(b_{k-1}\) and update GSO

7: \(k := k - 1\) \((k > 1)\)

8: else

9: for \(l = k - 2, k - 3, ..., 1\) do

10: size-reduce \(b_k\) against \(b_l\)

11: \(k := k + 1\)

A. Effective LLL Reduction

Since the decision region of the SIC decoder is determined by GS vectors \(\hat{B}_i\) only, and since size reduction does not change them, a weaker version of the LLL algorithm is sufficient for SIC decoding. This makes it possible to devise a variant of the LLL algorithm that is faster than the standard one.

From the above argument it seems that we would be able to remove the size reduction operations at all. However, this is not the case. To see why, let us recall the derivation of the proximity factor [9].

By induction of (9) we have

\[
\|\hat{b}_j\|^2 \leq \beta^{i-j}\|\hat{b}_i\|^2, \quad \text{for } 1 \leq j < i \leq n \quad (10)
\]

where \(\beta \triangleq 1/(\delta - 1/4)\). This along with the fact \(\|\hat{b}_1\| = \|b_1\| \geq \lambda(\mathcal{L})\) implies the following upper bound on the proximity factor

\[
\rho_{SIC} \leq \beta^{n-1}. \quad (11)
\]

Clearly, the condition

\[
|\mu_{i,i-1}| \leq 1/2, \quad 1 < i \leq n \quad (12)
\]
is essential in maintaining the performance of SIC decoding.

A basis satisfies condition (8) and (12) is called an effectively LLL-reduced basis in [10]. Effective LLL reduction terminates in exactly the same number of iterations, because size-reducing against other vectors has no impact on the Lovasz test.

Effective LLL reduction permits us to remove from Algorithm 1 the most expensive part, i.e., size-reducing \(b_k\) against \(b_{k-2}, b_{k-3}, ..., b_1\) (Lines 9-11). For integral bases, doing this may cause explosion of the (rational) GSO coefficients \(\mu_{i,j}, j < i - 1\), and the increase of bit lengths will likely offset the computational saving. This is nonetheless not a problem in MIMO decoding, since the basis vectors and GSO coefficients are represented by floating-point numbers after all. We use a model where floating-point operations take constant time, and accuracy is assumed not to perish. In practice this holds with sufficient arithmetic precision, so real-world implementations would be advised to use effective LLL reduction.

We emphasize that under this condition the effective and standard LLL algorithms have indistinguishable error performance in the application to SIC decoding, as asserted by Proposition 1.

**Proposition 1:** The SIC decoder with effective LLL reduction finds exactly the same lattice point as that with standard LLL reduction.

B. Transformation to Standard LLL-Reduced Basis

On the other hand, the ZF decoder does require the condition of full size reduction. The derivation of the proximity factor hinges on the lower bound on \(\theta_i\) for an LLL-reduced basis [9]

\[
\sin \theta_i \geq \left(\frac{2}{3\sqrt{\beta}}\right)^{n-1}. \quad (13)
\]

This lower bound along the trivial fact \(\|b_i\| \geq \lambda(\mathcal{L})\) leads to

\[
\rho_{ZF} \leq \left(\frac{9\beta}{4}\right)^{n-1}. \quad (14)
\]

It is the lower bound on \(\theta_i\) that requires full size reduction.

Nonetheless, we can easily transform an effective LLL-reduced basis into a fully LLL-reduced basis. To do so, we simply perform size reductions at the end to make the other coefficients \(|\mu_{i,j}| \leq 1/2\), for \(1 \leq j < i - 1, 2 < i \leq n\).
This is because, once again, such operations have no impact on the Lovasz condition.

There are likely multiple bases of the lattice \( L \) that are LLL-reduced. For example, a basis reduced in the sense of Korkin-Zolotarev (KZ) is also LLL-reduced. Proposition 2 shows that this version results in the same reduced basis as the LLL algorithm.

Proposition 2: Fully size-reducing an effectively LLL-reduced basis gives exactly the same basis as the standard LLL algorithm.

IV. Complexity Analysis and Experimental Results

In this Section, we evaluate the computational complexity in terms of floating-point operations (flops). The other operations such as looping and swapping are ignored. The complexity analysis consists of two steps. Firstly, we bound the average number of iterations. Secondly, we bound the number of flops of a single iteration.

The number \( K \) of iterations is exactly the number of Lovasz tests. Let \( K^+ \) and \( K^- \) be the numbers of positive and negative tests, respectively. Since \( k \) is incremented in a positive test and decremented in a negative test, and since \( k \) starts at 2 and ends at \( n \), we must have \( K^+ \leq K^- + (n-1) \). Thus it is sufficient to bound \( K^- \).

Following [8], we define a potential function

\[
D = \prod_{i=1}^{n-1} D_i = \prod_{i=1}^{n-1} \| \mathbf{b}_i \|^{2(n-i)}
\]

where \( D_i = \det L_i = \| \mathbf{b}_1 \| \| \mathbf{b}_2 \| \cdots \| \mathbf{b}_i \| \) and \( L_i \) is defined as the sublattice spanned by \( \mathbf{b}_1, \ldots, \mathbf{b}_i \).

During the size-reduction step, \( D \) does not change, since the GS vectors are unchanged. \( D \) only changes during the swapping step, where \( D_{k-1} \) is reduced by a factor \( < \delta - \mu^2_{k-1} \leq \delta \), whereas other \( D_i \)'s are unchanged (as the determinant of the sublattice \( L_i \) does not change). Hence \( D \) decreases by a factor \( \delta \) at most.

Let \( A = \max_i \| \mathbf{b}_i \|^2 \) and \( a = \min_i \| \mathbf{b}_i \|^2 \). The initial value of \( D \) can be bounded from above by \( A^{n(n-1)/2} \). For an integral basis, \( D \) is one at least. Consequently, one has

\[
K^- \leq \frac{n(n-1)}{2} \log A
\]

where the logarithm is taken to the base \( 1/\delta \) (and throughout this paper). This is how the commonly known \( O(n^4) \) complexity of the LLL algorithm arises for integral bases. To bound the number of iterations for a real-valued basis, we invoke the following lemma [8], [11].

Lemma 1: During the execution of the LLL algorithm, the maximum \( A \) is non-increasing while the minimum \( a \) is non-decreasing.

In other words, the LLL algorithm tends to reduce the interval \([a, A]\) where the lengths of GS vectors reside. From Lemma 1, we obtain

\[
K^- \leq \frac{n(n-1)}{2} \log \frac{A}{a}. \tag{16}
\]

Assuming that the basis vectors are i.i.d. in the unit ball of \( \mathbb{R}^n \), Daude and Vallee showed that the mean of \( K^- \) is upper-bounded by \( O(n^2 \log n) \) [11]. In this Section, we analyze the average number of iterations under the assumption that the entries of \( B \) are i.i.d. standard normal \( \mathcal{N}(0,1) \) distributed, which is a typical model for MIMO communication over fading channels.

We use the exact value of \( D_0 \) instead, defined as the initial value of \( D \), to bound the average number of iterations. It leads to a better bound than using the maximum \( A \) in (16). The lower bound on \( D \) is followed:

\[
D_l = \frac{n(n-1)}{2} \log a \leq D.
\]

Accordingly, the mean of \( K^- \) is bounded by

\[
E[K^-] \leq E \left[ \log \frac{D_0}{D_l} \right] = E[\log D_0] - E[\log D_l] = E[\log D_0] - \frac{n(n-1)}{2} E[\log a]. \tag{17}
\]

We shall bound the two terms separately.

The QR decomposition of an i.i.d. Gaussian random matrix has the following property: the squares of the diagonal elements \( r_{ii} \) of the matrix \( R \) are statistically independent \( \chi^2 \) random variables with \( n-i+1 \) degrees of freedom. Since \( r_{ii}^2 = \| \mathbf{b}_i \|^2 \), we have

\[
E[\log D_0] = \sum_{i=1}^{n-1} (n-i) E[\log \| \mathbf{b}_i \|^2] \leq \sum_{i=1}^{n-1} (n-i) \log E[\| \mathbf{b}_i \|^2] \tag{18}
\]

where the first inequality follows from Jensen’s inequality \( E[\log X] \leq \log E[X] \).

It remains to determine \( E[\log a] \), which is negative since \( E[\log a] \leq \log E[a] \) and \( E[a] \leq 1 \). The cumulative distribution function (cdf) \( F_a(x) \) of the minimum \( a \) can be written as

\[
F_a(x) = 1 - \prod_{i=1}^{n} [1 - F_i(x)]
\]

where \( F_i(x) \) denotes the cdf of a \( \chi^2 \) random variables with \( i \) degrees of freedom. As \( n \) tends to infinity, \( F_a(x) \) approaches a limit \( cdf \bar{F}_a(x) \) that is not a function of \( n \). Since \( a \) decreases with \( n \), \( E[\log a] \) is necessarily bounded from below by its limit:

\[
E[\log a] \geq \int_{0}^{\infty} \log x d\bar{F}_a(x) \equiv -\log 6 \tag{19}
\]

where the integration has been numerically evaluated. A fully analytic treatment will be pursued in the future.
Substituting (18),(19) into (17), we obtain
\[ E[K^+] \leq \frac{n(n-1)}{2} \log 6n. \] (20)

Updating the GSO coefficients [8] during the swapping step costs \(6(n-k)+7 \leq 6n-5\), \((k \geq 2)\) flops, whereas pairwise size reduction in the subroutine for \((k, k-1)\) costs \(2n+2(k-1) \leq 4n-2\) flops. Testing the Lovasz condition as (9) costs 3 flops each time. Besides, the initial GSO costs \(2n^3\) flops. Therefore, the total cost is bounded by
\[ C \leq (6n-5)K^+ + (4n-2+3)(K^- + K^+) + 2n^3 \]
\[ \leq (6n-5)\frac{n(n-1)}{2} \log 6n \]
\[ + (4n+1)n(n-1) \log 6n \]
\[ + (n-1) \log 6n \]
\[ \leq 7n^3 \log 6n + 2n^3. \] (21)

This completes the complexity analysis of the effective LLL reduction. To obtain a fully reduced basis, we further run pairwise size reduction for \(l = k - 2\) down to 1 for each \(k = 3, \ldots, n\). The additional number of flops required is bounded by
\[ \sum_{k=3}^{n} \sum_{l=1}^{k-2} (2n+2l) = \sum_{k=3}^{n} [2n(k-2) + (k-1)(k-2)] \]
\[ = \frac{4}{3} n(n-1)(n-2) \leq \frac{4}{3} n^3. \] (22)

On the other hand, full size reduction is always performed during the execution of the original LLL algorithm. During each step, the extra number of flops is no more than the inner sum of (22), which is less than \(3n^2\). Thus, on top of (21), it will incur up to the following number of flops
\[ 3n^2K^+ \leq 3n^2 \left[ \frac{n(n-1)}{2} \log 6n + (n-1) \right] \] (23)
which is thought to be dominant. This results in the assumed \(O(n^4 \log n)\) time-complexity bound of the standard LLL algorithm.

Finally, in Fig. 2, we show the theoretic upper bounds and experimental results for effective and standard LLL algorithms. Clearly there is room to improve the analysis. This is because our work is in fact a blend of worst- and average-case analysis, and the resultant theoretic bound is unlikely to be sharp. But nonetheless, the experimental data exhibit cubic growth with \(n\), thereby supporting the \(O(n^3 \log n)\) bound. On the other hand, surprisingly, the real complexity of standard LLL reduction is not much higher. We observed that this is because of the small probability to execute Lines 9-11 of the standard LLL algorithm (which were thought to dominate the complexity.) For example, the probability is only 2% when \(n = 20\). Such issues will be studied in future research.

ACKNOWLEDGMENT
The authors would like to thank W. H. Mow and Y. H. Gan for helpful discussions on complexity analysis.

REFERENCES