

# Performance Evaluation of Lower Complexity Hybrid-Fix-and-Round-LLL Algorithm for MIMO System

**Huazhang Lv<sup>1</sup>**

<sup>1</sup>5G Innovation Center, Network Technology Research Institute of China Unicom  
Beijing, China

[e-mail: lvhz7@chinaunicom.cn, huazhang0526@163.com ]

\*Corresponding author: Huazhang Lv

*Received July 12, 2017; revised December 27, 2017; accepted January 24, 2018;  
published June 30, 2018*

---

## **Abstract**

Lenstra-Lenstra-Lovász (LLL) is an effective receiving algorithm for Multiple-Input-Multiple-Output (MIMO) systems, which is believed can achieve full diversity in MIMO detection of fading channels. However, the LLL algorithm features polynomial complexity and shows poor performance in terms of convergence. The reduction of algorithmic complexity and the acceleration of convergence are key problems in optimizing the LLL algorithm. In this paper, a variant of the LLL algorithm, the Hybrid-Fix-and-Round LLL algorithm, which combines both fix and round measurements in the size reduction procedure, is proposed. By utilizing fix operation, the algorithmic procedure is altered and the size reduction procedure is skipped by the hybrid algorithm with significantly higher probability. As a consequence, the simulation results reveal that the Hybrid-Fix-and-Round-LLL algorithm carries a faster rate of convergence compared to the original LLL algorithm, and its algorithmic complexity is at most one order lower than original LLL algorithm in real field. Comparing to other families of LLL algorithm, Hybrid-Fix-and-Round-LLL algorithm can make a better compromise in performance and algorithmic complexity.

---

**Keywords:** LLL algorithm, lattice reduction, algorithmic complexity, size reduction, MIMO system

---

This research was supported by 5G network evolution, key technology research and business demonstration project (Z9B17ZU0R00009) of China Unicom.

## 1. Introduction

Lattice reduction (LR) [1]-[2] is a mainstream decoding technique. Its near-optimum performance comes at significantly lower complexity than ML detection. Due to this characteristic, LR decoding possesses significant advantages in MIMO communication [3]. A large part of the decoding problem in lattice reduction involves solving the closet vector problem (CVP) of a lattice [4]. In 1982, A. K. Lenstra, H. Lenstra, Jr., J., and L. Lovász first introduced the Lenstra-Lenstra-Lovász (LLL) algorithm [5] which was considered the most practical algorithm in lattice reduction. At the same time, the complexity of LLL algorithm is growing of polynomial. By means of the proximity factors method, a performance gap between LLL and ML decoding has been identified [6]-[7].

It is well known that the bound of operation complexity in LLL is  $O(n^4 \log n)$  [8]. With larger systems, reducing the whole complexity of LLL algorithm is still a challenge [9]. Effective LLL algorithm (E-LLL) [10] loosely imposes an ascending order on the diagonal elements of a channel matrix. E-LLL has a demonstrable complexity bound  $O(n^3 \log n)$  that is one order lower than the LLL algorithm. Another new lattice basis reduction, called diagonal reduction [11]. It only imposes one single constraint on the diagonal elements, the execution times of the column swap procedure are reduced. The Complex LLL algorithm (CLLL) [12] expands the definition of reduced basis to include more complex values. According to theoretical analysis, CLLL requires less arithmetic operations than LLL [13].

Due to the implementation of LLL algorithm suffers from variable run-time and complexity, modifications of the algorithmic architecture have been proposed. A fixed complexity LLL (fc-LLL) algorithm [14] introduces a deterministic structure for an easy implementation. In [15], the same thought of fixed complexity structure is expanded to LLL-Deep algorithm. A novel possible-swap-LLL (PSLLL) [16] algorithm tries to change the flows of the algorithm procedure. In recent years, the efficient greedy LLL algorithm [17], proposes a new relaxed Lovász condition to search the candidate set of LLL iterations. According to [17], a more computationally efficeient fixed complexity LLL (CE-fcLLL) algorithm is developed. In this paper, it introduce a power factor (PF) to measure the relation of reduced basis and transmit power of lattice-reduction-aided precoding. And based on this factor, it design a new LLL loop. In paper [19], it propose two greedy selection-based fcLLL algorithms. These algorithms can reduce more computaiton time in normal or large MIMO systems. Other details of development of LLL algorithm and MIMO detection techniques can be referred to in [20]-[22]

In this paper, a Hybrid-Fix-and-Round LLL (Hybrid-F&R-LLL) algorithm is introduced. It can significantly reduce the complexity of LLL algorithm and maintain a better compromise between performance and complexity. In the original LLL algorithm, a round measurement that rounds each element to the nearest integer is applied to parameter  $\mu$ . Round operation guarantees the accuracy of size reduction, however the duration of convergence will increase.

In the novel Hybrid-F&R-LLL algorithm, a strategy for combining fix and round measurement is introduced, and the new inserted fix measurement rounds the parameter  $\mu$  towards zero. This measurement will increase the probability of the parameter  $\mu$  obtaining a value of 0, a condition which when met means the size reduction procedure will be skipped. For example, for a large part of the duration, parameter  $\mu \in (0,1)$ , so it will get  $fix(\mu) = 0$  and

size reduction procedure is skipped. Since the size reduction procedure has a close connection to a basis of LLL reduced. In reality, the Hybrid-F&R-LLL algorithm aims at loosening the definition of a basis of LLL reduced. The value of threshold that basis conform to is extended, from  $\frac{1}{2}$  (in the original LLL) to 1 (in the Hybrid-F&R-LLL).

Due to the size reduction condition is modified in Hybrid-F&R-LLL algorithm, so the whole procedure and performance of algorithm are also changed. In this paper, LLL potential is introduced to measure the change of LLL power. In each iteration of algorithm, the change of LLL potential is computed. The algorithm has fast convergence characteristic will reduced the LLL potential at most. When the algorithm stopped, the LLL potential will achieve on a lower stage and may not be reduced anymore. According to analyses of LLL potential, it is known that skipping the size reduction procedure is sometimes beneficial for achieving the largest reduction in LLL potential. This will accelerate the speed of algorithmic convergence. In reality, Hybrid-F&R-LLL algorithm simplify the size reduction. The basis vector can directly process the column swap procedure. Hybrid-F&R-LLL can reduce by at least one order the total costs of arithmetic operation to  $O(\lambda n^3 \log B)$ . The complexity of

Hybrid-F&R-LLL algorithm is not fixed. Its complexity varies between  $O(3n^3 \log B)$  to  $O(7n^3 \log B)$ , the complexity of E-LLL. However, the Hybrid-F&R-LLL algorithm usually causes a loss in performance with respect to BER. With the help of the proximity factors method, an upper performance bound of the Hybrid-F&R-LLL algorithm has been

derived:  $\rho_{Hybrid-F\&R-LLL} \leq \left(\frac{4\beta}{3\beta-4}\right)^{n-1}$ . Compare to the other modification version of LLL

algorithm, the new hybrid version build a compromise between the algorithmic complexity and performance. The hybrid version improve the performance compared to Greedy-LLL and reduce the complexity nearly an order compared to E-LLL and Diagonal-LLL. For the higher order of antenna configuration, the analysis of performance bound and complexity may be a little bit different. The dominant parameter tranverse from the base to the exponent. For the details need to further study.

In the simulation part, this paper gives the simulation results for  $4 \times 4$  MIMO systems. It compares the SNR of each algorithm that needs on a fixed level of bit error rate (BER). The simulation results show that the performance of Hybrid-F&R-LLL is inferior to the LLL algorithm. However, CDF (Cumulative Distribution Function) is used to display convergence. The results demonstrate that the Hybrid-F&R-LLL can achieve a faster convergence compared with the LLL algorithm. Some well-known modification algorithms such as Greedy-LLL, Effective-LLL and Diagonal-LLL are also taken into account. Among these algorithms, Hybrid-F&R-LLL makes a better compromise between performance and algorithmic complexity.

The rest of the paper is organized as follows. Section 2 presents the basic conception of the MIMO system model, lattice reduction and LLL algorithm. New definition of Hybrid-F&R-LLL algorithm and theoretical statements will be discussed in Section 3. In Section 4 it tries to analyze the convergence characteristics and in Section 5 derive the performance bound of Hybrid-F&R-LLL with the help of proximity methods. The results of analyzing average complexity are included in Section 6. A discussions for high order antennas

configuration is listed in Section 7. Simulation results are demonstrated in Section 8 and finally, a brief conclusion is given in Section 9.

Notation: In this article,  $\Re(x)$  and  $\Im(x)$  denote the real and imaginary part of  $x$  respectively. The inner product in the complex Euclidean space between vectors  $u$  and  $v$  is defined as  $\langle u, v \rangle = u^H v$  and the Euclidean length is  $\|u\| = \sqrt{\langle u, u \rangle}$  in  $\mathfrak{R}^n$ . Symbol  $\lceil \cdot \rceil$  represents an arbitrary integer closest to  $x$  and the transpose, Hermitian transpose, inverse of a matrix  $H$  defined by  $H^T$ ,  $H^H$ ,  $H^{-1}$  respectively. Expectation of a matrix  $H$  is represented by  $E(H)$  and  $\sigma^2$  is the variance. Parameter  $C$  represents the complexity of algorithm and parameter  $\rho$  represents the upper performance bounds.

## 2. MIMO System Model and Lattice Basis Reduction

### 2.1 MIMO System Model

Here it use a  $N \times M$  channel matrix to denote a MIMO system with  $M$  transmit antennas and  $N$  receive antennas. It is assumed that the transmitted signal at the  $m$ th transmit antenna is  $x_m$ , and the data received at the  $n$ th receive antenna is  $y_n$  ( $s_m \in \mathbb{Z} + j\mathbb{Z}$ ,  $y_n \in \mathbb{Z} + j\mathbb{Z}$ ). A common signal alphabet  $X$  is used for all  $x_m$ . Over the MIMO channel, the received signal vector is represented below:

$$y = H_r x + n \quad (1)$$

Matrix  $H_r$  consists of  $M \times N$  independent and identically distributed (*i.i.d*) complex Gaussian coefficients with zero mean and unit variance. Note that  $n$  is assumed to be a *i.i.d* complex Gaussian variable with unit variance. The variance is:  $E[nn^H] = 2\sigma^2 I$ .

### 2.2 Definition of Lattice Reduction

**Definition 1 (Definition 1.9 [22]):** Let  $n > 1$  and let  $x_1, x_2, x_3, \dots, x_n$  be a basis of  $R^n$ . The lattice with dimension  $n$  and basis  $x_1, x_2, x_3, \dots, x_n$  is the set  $L$  of all linear combinations of the basis vectors with integral coefficients:

$$L = \mathbb{Z}x_1 + \mathbb{Z}x_2 + \dots + \mathbb{Z}x_n = \left\{ \sum_{i=1}^n a_i x_i \mid a_1, a_2, \dots, a_n \in \mathbb{Z} \right\} \quad (2)$$

The basis can be seen as span or generate of lattice. Each lattice may be spanned by not only one basis. Different basis can be transferred by a unimodular matrix whose determinant is  $\pm 1$  and each matrix element is an integer.

**Definition 2 (Lemma 1.10 [22]):**  $x_1, x_2, x_3, \dots, x_n$  and  $y_1, y_2, y_3, \dots, y_n$  are two basis for the same lattice  $L \in R^n$ . Let  $X, Y$  be the  $n \times n$  matrix with  $x_i, y_j$  in row  $i$  for  $j = 1, 2, 3, \dots, n$ . The

conclusion is that it can use a unimodular matrix to link the two basis  $x_i, y_j$ .

$$Y = C \times X \quad (3)$$

The goal of lattice reduction is to find a unimodular matrix and then to simplify the basis. The unimodular matrix has more advantages compared to the original channel matrix including higher orthogonality and lower complexity. Once the unimodular matrix is obtained, the decoding procedure is listed below. Here it takes the LLL procedure as an example to explain the process of lattice reduction (definitions of the parameters are referred to in [Table 1](#)):

$$H_r = H_r' \times T_r \quad (4)$$

$$y = H_r \times T_r \times x + n \quad (5)$$

$$y = H_r \times c + n \quad (6)$$

Where  $c = T_r \times x$ ,  $T_r$  is a unimodular matrix and  $s = T_r^{-1}c$ . By applying the unimodular matrix to signal detection, the detection performance will be improved. However, the real problem then becomes how to find a good unimodular matrix.

### 2.3 Definition of Lattice Reduction

LLL algorithm is proposed to find a matrix with nearly orthogonal column vectors so as to generate the same lattice. In other words, LLL is proposed to find the unimodular matrix introduced in section 2.2. Lattice reduction can be performed for the M-basis MIMO system with the  $N \times M$  channel matrix. It concentrate on real-valued matrices for lattice reduction in a MIMO system.

First, derive the definition of LLL reduction under Gram-Schmidt orthogonalization. This definition was first introduced in [5]. If  $B = (b_1, b_2, \dots, b_n)$  is called the basis of  $L$ , a lattice is generated as the linear combination of integers of some set of linearly independent vectors. It can thus build the relationship between lattice  $L$  and basis  $B$  as equation (2) does.

$$L \triangleq L(B) = \left\{ \sum_{i=1}^n x_i b_i \mid x_1, x_2, \dots, x_n \in Z \right\} \quad (7)$$

The basis has Gram-Schmidt orthogonalization:

$$B = \hat{B} [\mu_{i,j}]^T \quad (8)$$

Where  $\hat{B} = [\hat{b}_1, \dots, \hat{b}_n]$  and  $[\mu_{i,j}]^T$  is a lower-triangular matrix with unit diagonal elements—the detailed procedure of Gram-Schmidt orthogonalization is omitted. A basis is LLL reduced if:

$$|\mu_{i,j}| \leq \frac{1}{2} \text{ for } 1 \leq j < i \leq n \quad (9)$$

$$\left| \hat{b}_i \right|^2 + \mu_{i,i-1}^2 \left| \hat{b}_{i-1} \right|^2 \geq \delta \left| \hat{b}_{i-1} \right|^2 \text{ for } 1 < i \leq n \tag{10}$$

Parameter  $\delta$  takes its value from interval  $\left(\frac{1}{4}, 1\right)$ . A small value of  $\delta$  leads to a fast convergence; whereas, a large value of  $\delta$  can lead to a slow convergence and better basis. The relationship displayed in (9) and (10) may be understood as an effective tool to derive the performance bound of the family of LLL algorithms. However, in the Monte Carlo simulation, it is suggested to frequently replace Gram-Schmidt orthogonalization with QR decomposition to speed up the simulation.

**Definition 3(LLL Reduction based on QR decomposition):** A basis  $H_r \in C^{m \times n}$  is LLL reduction process with parameter  $\delta \left(\frac{1}{4} < \delta < 1\right)$ , if the upper triangular factor  $R_r \triangleq [r_{i,j}]$  in its QR decomposition  $H_r = Q_r \times R_r$  satisfies:

$$\left| R_r(l, k) \right| \leq \frac{1}{2} \left| R_r(l, l) \right| \quad 1 \leq l < k \leq 2M \tag{11}$$

$$\delta R_r^2(k-1, k-1) \leq R_r^2(k, k) + R_r^2(k-1, k) \quad 1 \leq l < k \leq 2M \tag{12}$$

Where  $R_r(l, k)$  denotes the  $(l, k)$ th entry of  $R_r$ , also can be viewed as the element in  $l$ th row  $k$ th column.

Matrix  $Q_r$  is an orthogonal matrix. The different column vectors in  $Q_r$  expressed as  $q_i$  are mutually orthogonal. Also,  $q_i$  is an orthogonal basis where  $|q_i| = 1$ . And matrix  $R_r$  is an up-triangular matrix. The relationship between QR decomposition and Gram-Schmidt orthogonalization is shown below:

$$\mu_{j,i} = \frac{R_r(i, j)}{R_r(i, i)} \tag{13}$$

$$\hat{b}_i = R_r(i, i) \bullet q_i \tag{14}$$

The  $q_i$  is the  $i$ th column of  $Q_r$ , and vice versa. If set the square of the absolute value on both sides of the equation sign:

$$\left| \hat{b}_i \right| = \left| R_r(i, i) \right|^2 \tag{15}$$

This will be a valuable conclusion that will be used in the derivation of the union performance bound.

The inequality (12) is usually called Lovász condition. Parameter  $\delta$  is a real number that is arbitrarily chosen from  $\left(\frac{1}{4}, 1\right)$ , while  $\delta = \frac{3}{4}$  is considered the best complexity-quality trade-off coefficient. Lastly, the LLL algorithm generates a LLL-reduced matrix from the real-valued

channel matrix  $H_r$ .

**Table 1.** Process of LLL algorithm

	<b>Initialization:</b> <b>Input:</b> $H_r \in C^{m \times n}$ , $\delta \in \frac{3}{4}$ <b>Output:</b> $Q_r, R_r, T_r$ ( $Q_r, R_r$ are updated and $T_r$ becomes a unimodular matrix at last)
No.	Algorithm process
1	$qr(H_r) \rightarrow [Q_r, R_r]$
2	$n = \text{size}(H_r, 2)$
3	Initialization $T_r, T_r$ at first is an identity matrix. $T_r = I_n$ ;
4	$k=2$
5	while $k \leq n$
6	For $l=1:k-1$
7	$\mu = \text{round}(R_r(k-l, k)/R_r(k-l, k-l))$
8	If $\mu \neq 0$
9	$R_r(1:k-l, k) = R_r(1:k-l, k) - \mu \times R_r(1:k-l, k-l)$
10	$T_r(:, k) = T_r(:, k) - \mu \times T_r(:, k-l)$ ;
11	End if
12	End for
13	if $\delta R_r^2(k-1, k-1) > R_r^2(k, k) + R_r^2(k-1, k)$
14	swap the $(k-1)$ th and $k$ th columns in $R_r$ and $T_r$
15	find a Givens rotation $G$ to restore the upper triangular structure of $R$
16	$R_r(k-1:k, k-1:n) \leftarrow G R_r(k-1:k, k-1:n)$
17	$Q_r(:, k-1:k) \leftarrow Q_r(:, k-1:k) G^H$
18	$\max(k-1, 2) \rightarrow k$
19	Else
20	$k = k + 1$
21	End if
22	End while

Here should point out that the size reduction, indicated from line 6 to line 12 in **Table 1**, is a procedure that uses parameter  $\mu$  to adjust the element in matrix  $R_r$  and  $T_r$ . Givens rotation matrix  $G$  is generated by the following procedure:

$$G = \begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix} \quad (16)$$

$$\alpha = \frac{R_r(k-1, k-1)}{\|R_r(k-1:k, k-1)\|}, \beta = \frac{R_r(k, k-1)}{\|R_r(k-1:k, k-1)\|} \quad (17)$$

In specific, the expression on line 16 is written by a MATLAB statement where  $R_r(k-1:k, k-1:n)$  means it selects a matrix internal of the matrix  $R_r$  that consists of the  $k$ th to  $(k-1)$ th row and  $(k-1)$ th to  $n$ th column. Further, in line 17 of **Table 1**,  $Q_r(:, k-1:k)$  means it select a matrix internal of the matrix  $Q_r$  that consists of all the row elements, but is only limited to the  $(k-1)$ th to  $k$ th columns.

### 3. A Hybrid Fix&Round LLL (Hybrid-F&R-LLL) Algorithm

#### 3.1 Proposal of the Hybrid-F&R-LLL Algorithm

In this chapter, first propose a Hybrid-F&R-LLL algorithm. The details of algorithmic flow are listed in **Table 2**. The initialization of the algorithm is the same as for the LLL algorithm such as QR decomposition and generation of unimodular  $T_r$ . Before the algorithm performs size reduction, the Hybrid-F&R-LLL algorithm provides two kinds of preprocess strategy of parameter  $\mu$ : fix operation and round operation.

**Table 2.** Process of Hybrid-F&R-LLL algorithm

	<b>Initialization:</b> <b>Input:</b> $H \in C^{m \times n}, \delta \in \frac{3}{4}$ <b>Output:</b> $Q_r, R_r, T_r$ ( $Q_r, R_r$ are updated and $T_r$ becomes a unimodular matrix at last)
No.	Algorithm process
1	Do the procedures line 1 to line 4 in <b>Table 1</b>
2	while $k \leq n$
3	For $l = 1 : k - 1$
4	If $\frac{1}{2} < \frac{ R_r(k-l, k) }{ R_r(k-l, k-l) } < 1$
5	$\mu = \text{fix}(\frac{R_r(k-l, k)}{R_r(k-l, k-l)});$
6	else
7	$\mu = \text{round}(\frac{R_r(k-l, k)}{R_r(k-l, k-l)});$
8	End if
9	Doing size reduction line 8 to line 11 in <b>Table 1</b>
10	End for
11	The same strategy of column swap procedure line 13 to line 21 in <b>Table 1</b> .
12	End while

The fix operation function— $\text{fix}(X)$ —rounds the elements of  $X$  to the nearest integers towards 0, and the round operation— $\text{round}(X)$ —rounds the elements of  $X$  to the nearest decimal or integer. Based on the statistics, parameter  $\mu$  is mainly located at the interval of  $(0,1)$ . Applying fix measurement on parameter  $\mu$ ,  $\mu$  can obtain the value of 0 in each iteration. The threshold condition of size reduction  $\mu \neq 0$  cannot usually be satisfied, and once



the threshold condition of size reduction isn't satisfied, the size reduction procedure will be skipped. Thus, the fix operation has often been closely associated with the skipping of the size reduction procedure.

### 3.2 Derivation of New Definition under Hybrid-F&R-LLL algorithm

According to the algorithm process in **Table 2**, it may find that the value of  $\frac{|R_r(k-l,k)|}{|R_r(k-l,k-l)|}$  will decide the process of size reduction (line 8 to line 11 in **Table 1**). This

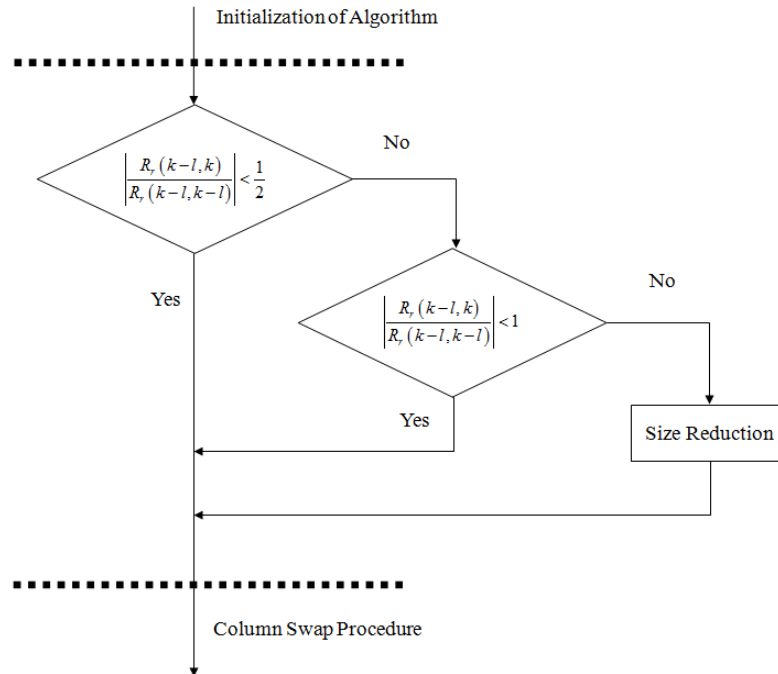
chapter will analyze the value of  $\frac{|R_r(k-l,k)|}{|R_r(k-l,k-l)|}$  and compare the algorithmic procedure

between the original LLL algorithm and the Hybrid-F&R-LLL algorithm. According to the definition in (11) and the procedure in **Table 2**:

$$|R_r(l,k)| \leq \frac{1}{2} |R_r(l,l)| \quad 1 \leq l < k \leq 2M \quad (18)$$

Divide the situation of the value of  $\frac{|R_r(k-l,k)|}{|R_r(k-l,k-l)|}$  into three parts for analyzing:

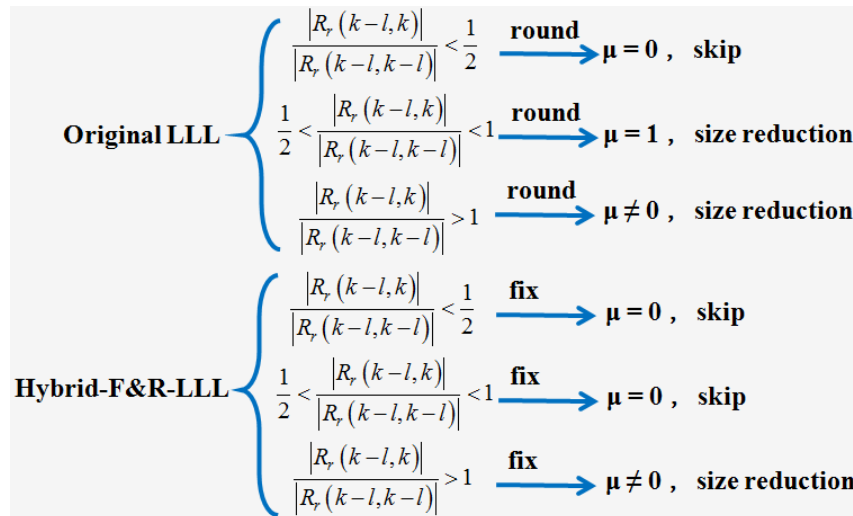
$$\frac{|R_r(k-l,k)|}{|R_r(k-l,k-l)|} < \frac{1}{2}, \quad \frac{1}{2} < \frac{|R_r(k-l,k)|}{|R_r(k-l,k-l)|} < 1 \quad \text{and} \quad \frac{|R_r(k-l,k)|}{|R_r(k-l,k-l)|} > 1.$$



**Fig. 1.** Algorithm Flow of Size Reduction in Hybrid-F&R-LLL algorithm

From **Fig. 1**, the introduction of *fix* operation helps to define which basis should perform the size reduction. In the original LLL algorithm, the basis which satisfies the

condition  $|R_r(l, k)| \leq \frac{1}{2}|R_r(l, l)|$  may be seen as a size reduced basis. However, in the Hybrid-F&R-LLL, the range of the size reduced basis has expanded to  $|R_r(l, k)| \leq |R_r(l, l)|$ . In **Fig. 2**, find that the only difference between the original LLL and Hybrid-F&R-LLL reduction is located at  $\frac{1}{2} < \frac{|R_r(k-l, k)|}{|R_r(k-l, k-l)|} < 1$ . So only  $\frac{|R_r(k-l, k)|}{|R_r(k-l, k-l)|} > 1$  size reduction will be applied.



**Fig. 2.** Comparison between Original LLL and Hybrid-F&R-LLL in Deciding Condition of Size Reduction

**Definition 4 (Hybrid-F&R-LLL Reduction based on QR decomposition):** A basis  $H_r \in C^{m \times n}$  is LLL-reduced with a reduction parameter  $\delta \left( \frac{1}{4} < \delta < 1 \right)$ , if the upper triangular factor  $R_r \triangleq [r_{i,j}]$  in its QR decomposition  $H_r = Q_r \times R_r$ , satisfies:

$$|R_r(l, k)| \leq |R_r(l, l)| \quad 1 \leq l < k \leq 2M \tag{19}$$

$$\delta R_r^2(k-1, k-1) \leq R_r^2(k, k) + R_r^2(k-1, k) \quad 1 \leq l < k \leq 2M \tag{20}$$

Where  $R_r(l, k)$  denotes the  $(l, k)$ th entry of  $R_r$ , it also can be viewed as the element in the  $l$ th row and  $k$ th column.

The definition of Hybrid-F&R-LLL reduction based on Gram-Schmidt orthogonalization can also be given below:

$$|\mu_{i,j}| \leq 1 \text{ for } 1 \leq j < i \leq n \tag{21}$$

$$\left| \hat{b}_i \right|^2 + \mu_{i,j}^2 \left| \hat{b}_{i-1} \right|^2 \geq \delta \left| \hat{b}_{i-1} \right|^2 \text{ for } 1 < i \leq n \tag{22}$$

#### 4. Analysis of Convergence Characteristics between Hybrid-Fix&Round-LLL and the Original LLL Algorithm

In this section, a definition of LLL potential will be used based on QR decomposition to verify whether the Hybrid-F&R-LLL Algorithm will converge faster than the original LLL algorithm.

LLL potential will be used as a tool to analyze the convergence properties of the LLL power. So, first introduce the LLL potential function which was first put forward in [5][10].

**Definition 5(Definition of LLL potential based on QR decomposition):** A positive real number  $D$  is defined as the LLL potential:

$$D = \prod_{i=1}^{M-1} d_i = \prod_{i=1}^{M-1} |R_r(i, i)|^{2(M-i)} \quad (23)$$

Where  $d_i = \det(L_i) = \prod_{j=1}^i |R_r(i, j)|^2$  and  $L_i$  is the sub-lattice spanned by column vectors  $q_1, q_2, \dots, q_i$  of matrix  $Q_r$ , which comes from the QR decomposition  $H_r = Q_r R_r$ .

LLL potential states the power of the LLL algorithm. While running the LLL algorithm, the LLL potential will be reduced until the algorithm terminates. With the aim of fast convergence and terminating, the reduction of  $D$  should be maximized in each iteration of the algorithm. This will be the main comparative method of the speed of convergence among the algorithms. The size reduction does not change the value of  $D$ , since the diagonal elements in  $R_r$  are unchanged. The value of  $D$  only changes after a column swap.

If an iteration occurs at index  $k$  the LLL potential will be updated by the column swap procedure because the diagonal elements in  $R_r$  have changed:

$$D_k = \prod_{\substack{i=1 \\ i \neq k-1, k}}^{M-1} |R_r(i, i)|^{2(M-1)} |R_r'(k-1, k-1)|^{2(M-K+1)} |R_r'(k, k)|^{2(M-K+1)} \quad (24)$$

Equation (24) indicates that the new potential consists of the original, unchanged diagonal elements in  $R_r$  and the two updated, swapped elements. Derive an equation based on  $D$  and  $D_k$ , which can be seen as a relationship between the values of the potential before and after the LLL iteration.

$$D_k = \frac{|R_r(k-1, k)|^2 + |R_r(k, k)|^2}{|R_r(k-1, k-1)|^2} D \quad (25)$$

Based on the precondition that it want to decrease the potential as much as possible, or for the algorithm to converge fast, the difference between  $D$  and  $D_k$  is as follows:

$$\Delta_k = D - D_k \quad (26)$$

$$\begin{aligned}\Delta_k &= \left( 1 - \frac{|R_r(k-1,k)|^2 + |R_r(k,k)|^2}{|R_r(k-1,k-1)|^2} \right) D \\ &= \left( 1 - \frac{|R_r(k-1,k)|^2}{|R_r(k-1,k-1)|^2} - \frac{|R_r(k,k)|^2}{|R_r(k-1,k-1)|^2} \right) D\end{aligned}\quad (27)$$

Index  $k$  is where that the largest decrease in potential occurs in the LLL iteration, and is defined as:

$$k = \arg \max_{2 \leq k \leq M} \Delta_k \quad (28)$$

According to line 8 to line 11 in **Table 1**, the diagonal elements in matrix  $R_r$  remain unchanged during size reduction, which means the term  $\frac{|R_r(k,k)|^2}{|R_r(k-1,k-1)|^2}$  remains unchanged.

If it wants  $\Delta k$  to be larger, the term  $\frac{|R_r(k-1,k)|^2}{|R_r(k-1,k-1)|^2}$  will be smaller. Thus, a fast reduction in LLL potential is obtained when  $|R_r(k-1,k)|^2$  is small. According to **Fig. 2**, it just only needs to explain the situation  $\frac{1}{2} < \frac{|R_r(k-l,k)|}{|R_r(k-l,k-l)|} < 1$ .

According to (28),  $|R_r(k-1,k)|^2$  is the key element that influences the reduction of LLL potential. Set the  $l=1$  for general analysis. Based on the statistics, parameter  $\mu$  is mainly located at the interval of (0,1). The conclusion can be expanded to other situations based on  $l=1$ .

$$\tilde{R}_r(k-1,k) = R_r(k-1,k) - R_r(k-1,k-1) \quad (\mu=1) \quad (29)$$

The term  $\tilde{R}_r(k-1,k)$  is the value obtained from size reduction after running the LLL algorithm. Compare the value of  $|\tilde{R}_r(k-1,k)|^2$  and  $|R_r(k-1,k)|^2$  using parameter  $\Delta R_r(k-1,k)$  which is used to measure the difference between them:

$$\Delta R_r(k-1,k) = |\tilde{R}_r(k-1,k)|^2 - |R_r(k-1,k)|^2 \quad (30)$$

$$= -2|R_r(k-1,k)| \cdot |R_r(k-1,k-1)| + |R_r(k-1,k-1)|^2 \quad (31)$$

Since  $\frac{|R_r(k-1, k)|}{|R_r(k-1, k-1)|} > \frac{1}{2}$ , then  $2|R_r(k-1, k)| > |R_r(k-1, k-1)|$ . Set square operation between equations:

$$|R_r(k-1, k-1)|^2 < 4|R_r(k-1, k)|^2 \quad (32)$$

The term  $|R_r(k-1, k-1)|^2$  and  $|R_r(k-1, k-1)|$  both have an upper bound. Use  $4|R_r(k-1, k)|^2$  and  $2|R_r(k-1, k)|$  to substitute  $|R_r(k-1, k-1)|^2$  and  $|R_r(k-1, k-1)|$ , respectively in (32):

$$\Delta R_r(k-1, k) < -4|R_r(k-1, k)|^2 + 4|R_r(k-1, k)|^2 = 0 \quad (33)$$

Therefore  $\Delta R_r(k-1, k)$  will always be less than 0 which means the term  $|R_r(k-1, k)|^2$  in the Hybrid-F&R-LLL algorithm is larger. Returning to (23):

$$\Delta_k = \left( 1 - \frac{|R_r(k-1, k)|^2}{|R_r(k-1, k-1)|^2} - \frac{|R_r(k, k)|^2}{|R_r(k-1, k-1)|^2} \right) D \quad (34)$$

It finds that in the Hybrid-F&R-LLL algorithm  $|R_r(k-1, k)|^2$  is larger than in the LLL algorithm and so it will cause a larger decrease  $\Delta_k$  in LLL potential. Then derive the conclusion that the Hybrid-F&R-LLL algorithm can maximize the reduction of the LLL potential.

The analysis above only hypothesizes for the index of  $k$  and  $k-1$ . Nevertheless, this conclusion can expand to all of the indexes.

**Proposal 4-1:** Hybrid-F&R-LLL algorithm converges faster than LLL algorithm.

## 5. Performance Bound of the Hybrid-F&R-LLL Algorithm Using the Proximity Factor Methods

### 5.1 Upper Performance Bounds under Hybrid-F&R-LLL by Proximity Factors

In this chapter, it will derive the performance bounds for the Hybrid-F&R-LLL algorithm with the proximity factors method which was first introduced in [7] by Professor CongLing. The general form of the upper performance bound is shown below:

$$\rho \leq (\omega \times \beta)^{n-1} \quad \left( \beta \triangleq \frac{1}{\delta - \frac{1}{4}} \right) \quad (35)$$

Parameter  $\omega$  measures performance. When  $\omega$  or  $\omega \times \beta$  become larger, the upper bound of the algorithm will become looser and performance will worsen.

The detailed procedure performance bound derivation of Hybrid-F&R-LLL algorithm is listed in Appendix A, here just gives the final expression:

$$\rho_{Hybrid-F\&R-LLL} \leq \left( \frac{4\beta}{3\beta-4} \right)^{n-1} \quad \left( \beta \triangleq \frac{1}{\delta - \frac{1}{4}} \right) \tag{36}$$

### 5.2 Comparison of Performance Bounds among LLL-SIC, Hybrid-F&R-LLL and ZF Algorithm by Proximity Factors

In [7], the performance bound of LLL-SIC (Successive Interference Cancellation) and ZF are given. SIC is a nonlinear aided operation applied to the LLL algorithm to eliminate interference in the detection results of the adjacent symbol.

**Table 3.** Performance Bounds of LLL-SIC, Hybrid-F&R-LLL and ZF Algorithm

Algorithm	Performance bound under proximity factors
LLL-SIC	$\rho_{LLL-SIC} \leq (\beta)^{n-1} \quad \left( \beta \triangleq \frac{1}{\delta - \frac{1}{4}}, \beta > \frac{4}{3} \right)$
Hybrid-F&R-LLL	$\rho_{Hybrid} \leq \left( \frac{4\beta}{3\beta-4} \right)^{n-1} \quad \left( \beta \triangleq \frac{1}{\delta - \frac{1}{4}}, \beta > \frac{4}{3} \right)$
ZF	$\rho_{ZF} \leq \left( \frac{9\beta}{4} \right)^{n-1} \quad \left( \beta \triangleq \frac{1}{\delta - \frac{1}{4}}, \beta > \frac{4}{3} \right)$
Greedy-LLL	$\rho_{Greedy-LLL} \leq (\beta)^{n-1} \quad (\beta = 2)$
Diagonal-LLL	$\rho_{Diagonal-LLL} \leq \left( \frac{4\beta}{4-\beta} \right)^{n-1} \quad \left( \beta \triangleq \frac{1}{\delta - \frac{1}{4}}, \beta > \frac{4}{3} \right)$

Compare the value of  $\beta$ ,  $\frac{9}{4}\beta$ , 2 and  $\frac{4\beta}{3\beta-4}$  :

$$\beta < \frac{4\beta}{3\beta-4} < 2 < \frac{9}{4}\beta, \quad \beta > \frac{4}{3} \tag{37}$$

From left to right, the performance bound is looser and the performance is inferior. So LLL-SIC offers the best performance and the Hybrid-F&R-LLL offers the second best. ZF can only achieve one order of diversity, thus its performance is worst of all. The Greedy-LLL algorithm gives a fixed value of  $\beta$ , thus the performance bound of Greedy-LLL is fixed. The entire conclusion conforms to the simulation results in Section 6.

## 6. Theoretic Upper Bound on the Average Complexity of Hybrid-F&R-LLL Algorithm

Ahead of analyzing, first give definition of the big  $O$  notion and other symbols.

**Definition 6(Big O notion [22]):** Suppose that  $f(x)$  and  $g(x)$  are functions with the same domain, which is a subset of the real numbers. The statement  $f(x) = O(g(x))$  means that for sufficiently large values of  $x$ , the quantity  $f(x)$  is at most a constant multiple of the quantity  $g(x)$ , in absolute value.

Now follow the procedure in paper [10], to derive the average complexity bound of the Hybrid-F&R-LLL Algorithm. The results of average complexity bound of the Hybrid-F&R-LLL Algorithm is listed in proposal 6-1:

**Proposal 6-1:** Define  $\lambda$  as the balance parameter between the original LLL and the Hybrid-F&R-LLL Algorithm. The total cost of the Hybrid-F&R-LLL Algorithm by using measurement flops is:

$$C_{\text{Hybrid-F\&R-LLL}} = O(\lambda \times n^3 \log B) \quad \lambda \in (3, 7) \quad (38)$$

When no size reduction procedure is operated  $\lambda = 3$ , and the lowest total costs of arithmetic operation are  $O(3n^3 \log B)$ . For the size reduction happens at index  $k$  and  $k-1$ , the worst-case scenario for the Hybrid-F&R-LLL Algorithm is that the algorithm procedure remains the same as the E-LLL algorithm. This time parameter  $\lambda = 7$  and the highest total costs of arithmetic operation are  $O(7n^3 \log B)$ , the same complexity as the ELLL algorithm.

When there exists a full-size reduction, the total costs will further extend to  $O(n^4 \log B)$ .

The detailed proof of proposal 6-1 is listed in Appendix B. The average complexity bounds of several algorithms in real field are listed in **Table 4** for comparison.

**Table 4.** Average Complexity Bounds of Several Algorithms in Real Field

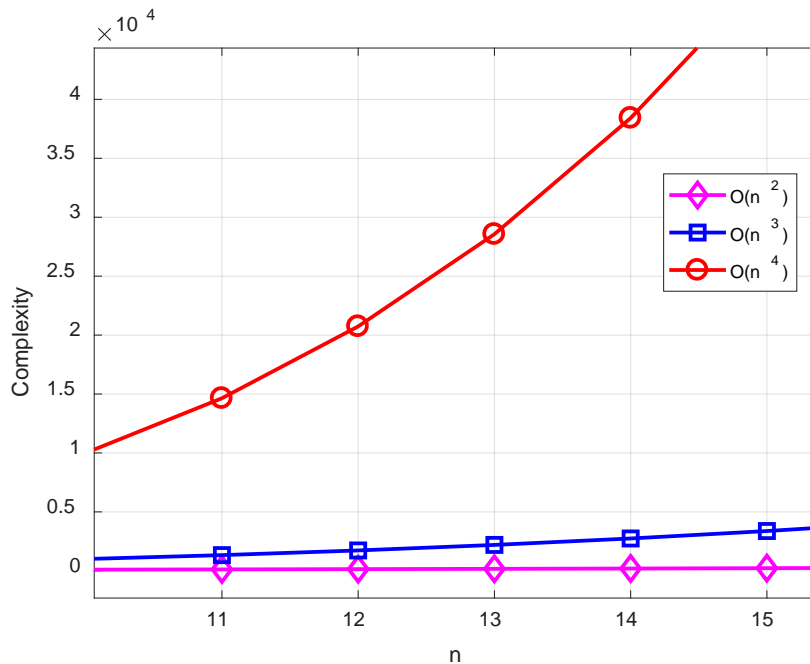
Algorithm	Field	Average Complexity Bounds
LLL	Real number field	$O(n^4 \log B)$
Hybrid-F&R-LLL	Real number field	$O(\lambda \times n^3 \log B) \quad \lambda \in (3, 7)$
Effective-LLL	Real number field	$O(7n^3 \log B)$
Diagonal-LLL	Real number field	$O\left(\frac{7}{2}n^3 \log B\right)$
Greedy-LLL	Real number field	$O\left(\frac{7}{2}n^2 \log B\right)$ (just column swap)

## 7. Discussions for High Order Antennas Configuration

All the antennas configuration selected in Section 8 is from  $2 \times 2$  to  $8 \times 8$ . And the simulation results in performance and complexity are confort to the analysis in Section 4 to Section 6. But when the MIMO system getting large, some difference in analysis should be point out.

For complexity, all the algorithmic complexity can be expressed as:  $O(\lambda n^a \log B)$ . The complexity of the algorithm is domined by the parameter  $\lambda$ ,  $a$  and  $n$ . In Section 6, this paper mainly discuss the influence of  $\lambda$ . If the size of MIMO system just below 8 transmitting antennas and 8 receiving antennas, it means  $n \leq 8$ . So  $\lambda$  has the same order as  $n$ . Parameter  $\lambda$ ,  $a$  and  $n$  both affects the algorithm complexity.

When  $n > 8$ , parameter  $n$  has the higher order than  $\lambda$ . This is the main difference compared to Section 6. Another difference that can not ignore is the parameter  $a$ , the exponent of  $O(\lambda n^a \log B)$ .



**Fig. 3.** Influence comparison of parameter  $a$  and  $n$  for algorithmic complexity

According to **Fig. 3**, it can be referred that for the same size of MIMO system, the higher of parameter  $a$ , the complexity is exponential growth. So for massive MIMO (higher order antenna configuration), the complexity is mainly domined by index  $a$  and the size of MIMO system. Among these algorithms, Greedy-LLL owns the lowest index  $a$  and it has the lowest complexity.

For performance, it is in a similar way. The performance bound can be written as  $\rho \leq (\omega \times \beta)^{n-1}$ . In Section 5, this paper discuss that the parameter  $\omega$  mainly influence the performance bound. But when the size of MIMO system getting large, term  $(\omega \times \beta)^{n-1}$  is exponential growth. The dominate item is not parameter  $\omega$ . It means that all the upper bound



of performance is:  $\rho \leq +\infty$ . All the algorithmic upper bound is the same, under the infinity. So it can not compare the performance bound referred to proximity factor. Analysis in higher order of antenna configuration is still a challenge. Proper measurements and method is for further study.

## 8. Simulation Results

In this chapter, it will use computer simulation techniques to verify the theoretical claims of LLL and its hybrid versions. Channel matrix  $H_r$  is assumed to remain constant over successive time intervals. And channel matrix  $H_r$  is known at the receiving terminal which is equivalent to ideal channel estimation at receiver side. White Gaussian noise  $n$  is randomly generated and assumed to be statistically independent and identically distributed. Inter-cell and intra-cell interference are ignored. Constellation mapping is settled with 16QAM. SNR is defined as the symbol energy per transmit antenna versus noise power spectral density. Channel realization is 10. Individually use performance gain to measure the performance of each algorithm at a fixed bit error rate (BER). At last in Section 8.4, a comparison between Hybrid-F&R-LLL and some latest achievement will be done. Hybrid-F&R-LLL still holds the characteristic of compromise between performance and complexity.

### 8.1 Average Number of Iterations and Computational Complexity

**Table 5.** Flops of each arithmetic operation in real field

Arithmetic operation	Flops needed
+ - × ÷	1
	1
$\sqrt{\quad}$	1
≤ and ≥	1
$round(\quad), fix(\quad)$	1

Each effective iteration of the different LLL algorithms is defined by the duration times of size reduction (line 6 to line 12 in **Table 1**) and the column swap procedure (line 13 to line 21 in **Table 1**).

In order to evaluate the computational complexity of different algorithms, it introduces the concept of real floating-point operations (flops). The flops of each arithmetic operation are defined in **Table 5**. In our algorithm, there are no complex operations where by the operation in the complex field is ignored.

The average number of iterations and computational complexity is shown in **Table 6**. (SR=size reduction procedure, CS=column swap procedure). In **Table 7**, make a contrast between the original LLL and other families of LLL algorithms. Select three other improved algorithms of the original LLL algorithm as a contrast. Diagonal-LLL [11], Effective-LLL [10] and Greedy-LLL [17] are all recent achievements in the field of lattice reduction. The idea of Greedy-LLL is first proposed in 2014 and the further improved is listed in [17] 2016.

**Table 6.** Comparison of different algorithms and numerical stats on flops and iterations in real field

	2×2		Flops	4×4		Flops	8×8		Flops
	Average iterations			Average iterations			Average iterations		
	SR	CS	SR	CR	SR	CR			
LLL Algorithm	3.425	3.725	200.45	20.84	23.52	2230.933	123.067	108.5	20391
Hybrid-F&R-LLL	0.875	3.125	132.6	2.013	16.053	1074.48	2.000	57.867	6636.733
Diagonal-LLL	1.175	3.725	161.1	6.813	23.48	1673.47	25.867	108.5	13421.73
Greedy-LLL	0.55	1.775	98	2.16	7.813	612.947	4.533	16.533	1989.8
Effective-LLL	2.525	3.725	175.8	11.813	23.533	1787.44	39.533	108.533	14042.4

**Table 7.** Flops of Algorithm Saving to Original LLL Algorithm

	2×2	4×4	8×8
LLL Algorithm	0	0	0
Hybrid-F&R-LLL	33.85%	51.84%	67.45%
Diagonal-LLL	19.63%	24.99%	34.18%
Greedy-LLL	51.11%	72.525%	90.24%
Effective-LLL	12.30%	19.88%	31.13%

As can see, except for Greedy-LLL, the Hybrid-F&R-LLL algorithm terminates with much fewer iterations and flops compared to other families of LLL algorithms. This property of the Hybrid-F&R-LLL algorithm is obvious. The statistical results and the conclusions outlined in Section 4 demonstrate that Hybrid-F&R-LLL can indeed decrease the LLL potential by the greatest amount in each iteration. Also, from 2×2 up to 8×8 antenna size, the Hybrid-F&R-LLL algorithm owns the lower average computation flops. Although Greedy-LLL has the lowest complexity, its performance is also the worst. These results can be referred to in Fig. 5.

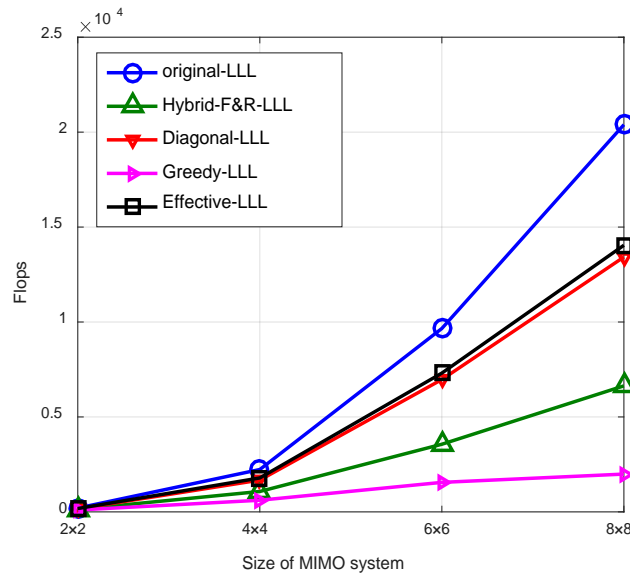
As the size of the MIMO system increases, as seen in Table 7, Hybrid-F&R-LLL can gradually save over 50% of the flops compared to the original LLL algorithm. In fact, it saves the second highest number of flops. An average complexity comparison based on the total number of flops among different sizes of MIMO system is shown in Fig. 4. Here it just gives a trend of the increase of operation flops in the detection algorithm. All the data is based on large numbers of experiments.

## 8.2 Performance Evaluation of Different Algorithms Based on Monte Carlo Simulation

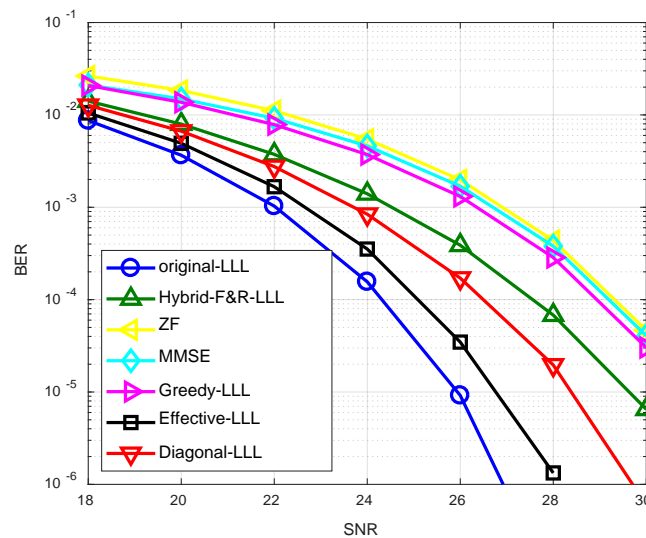
Here use a computer simulation technique to simulate the un-coded bit error rate (BER) performance of the family of LLL algorithms using MATLAB code. The simulation result of the 4×4 antenna configuration is shown in Fig. 5. The original LLL algorithm is seen as the best detection method because it can achieve full diversity. Minimum mean square error (MMSE) and zero-forcing (ZF) can only receive one order of diversity in a MIMO system. MMSE and ZF are simulated for comparison.

From the result plot in Fig. 5, the original LLL algorithm displays the best performance of

all. Whereas, E-LLL holds the best performance among the modified LLL algorithms. The performance gap between the E-LLL algorithm and the Diagonal-LLL algorithm at  $\text{BER}=10^{-4}$  is 1.44dB. E-LLL can achieve a 2.39dB performance gain compared to Hybrid-F&R-LLL. Greedy-LLL demonstrates a performance loss of 3.84dB compared to E-LLL at  $\text{BER}=10^{-4}$ . All these experimental results conform to the conclusion, which is derived in Section 5 especially that the performance bound of Hybrid-F&R-LLL is inferior to the original LLL algorithm. It can also derive the conclusion that the performance of the Greedy-LLL algorithm can only achieve the same order of as ZF and MMSE. Therefore, the order of diversity that Greedy-LLL can achieve is one. Due to this, the Greedy-LLL algorithmic performance is worse. The detailed BER of the different algorithms is shown in [Table 8](#).



**Fig. 4.** Complexity comparisons of different LLL algorithms from  $2 \times 2$  to  $8 \times 8$  MIMO systems



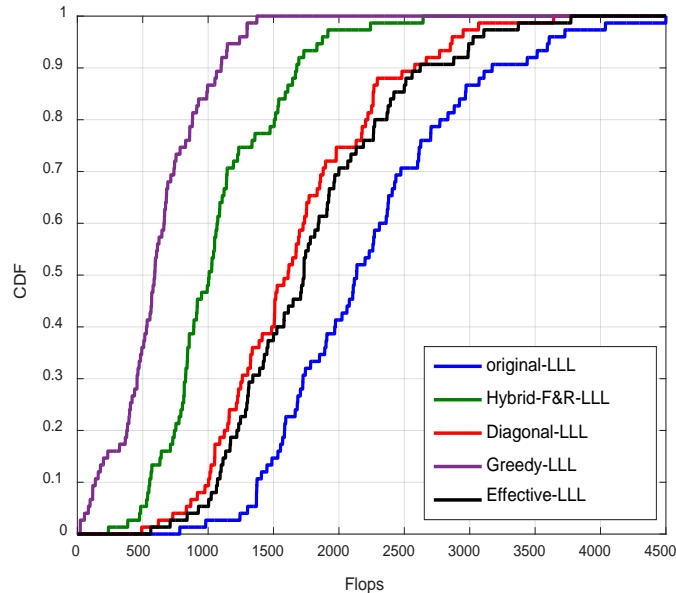
**Fig. 5.** BER Performance of LLL and Hybrid-F&R-LLL detectors in a 16QAM modulated  $4 \times 4$  MIMO system

**Table 8.** BER of Different Algorithm Based on Original LLL from SNR = 20 to 32

	20	22	24	26	28	30	32
LLL Algorithm	3.66E-03	1.03E-03	1.56E-04	9.18E-06	7.50E-08	0	0
Hybrid-F&R-LLL	1.38E-02	7.69E-03	3.55E-03	1.31E-03	3.61E-04	6.42E-05	6.43E-06
Greedy-LLL	2.08E-02	1.37E-02	7.85E-03	3.72E-03	1.31E-03	2.86E-04	3.03E-05
Effective-LLL	1.05E-02	4.93E-03	1.67E-03	3.53E-04	3.47E-05	1.33E-06	0
Diagonal-LLL	1.27E-02	6.67E-03	2.76E-03	8.41E-04	1.71E-04	1.96E-05	6.00E-07

### 8.3 Convergence Characteristic of Different Algorithms Based on Plotting of Cumulative Density Function

In Fig. 6, the cumulative density functions (CDF) of the number of iterations for completion of these five algorithms are shown. The initiation value of CDF is not the point of focus however. Only pay attention to the slope of the plots. Obviously, the algorithm, which has the steepest ascent curve, will have the fastest rate of convergence.



**Fig. 6.** Cumulative density function of the required flops to terminate the LLL and Hybrid-F&R-LLL algorithm for  $4 \times 4$  MIMO system

From the CDF function, it is apparent that the Hybrid-F&R-LLL converges faster than the original LLL algorithm and other families of LLL algorithms, except Greedy-LLL. With nearly 2500 flops, the Hybrid-F&R-LLL algorithm achieves almost 99% LLL basis reduction. However, with the original LLL algorithm, the number of flops is above 4000. Additionally, for Diagonal-LLL, Greedy LLL and E-LLL, the value of flops that achieves almost 99% LLL basis reduction is 3500, 1500, and 3800, respectively. Therefore, it can derive the conclusion here that Hybrid-F&R-LLL makes a better compromise between performance and algorithmic complexity.

### 8.4 Simulation Results of Recent Achievement in Lattice Reduction

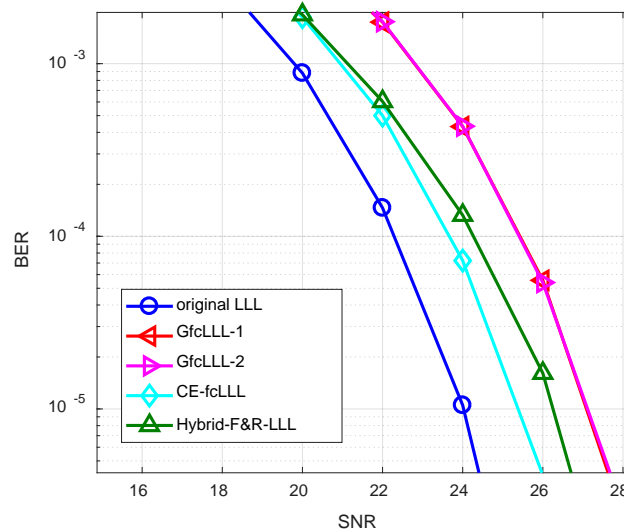
In this section, the selected algorithm includes Greedy-Fixed-Complexity LLL reduction algorithm [19]: GfcLLL(1) and GfcLLL(2). Moreover, it includes computation efficient fixed

complexity LLL algorithm (CE-fcLLL) [17]. All these achievements are published recent years on IEEE Communication Letters and IET Communications. The simulation parameters still keep the same as preamble. In this part, a new set of simulation data is selected due to the configuration of algorithm should be settled the same as families of Greedy fcLLL algorithm.

**Table 9.** Comparison of recent achievements and numerical stats on flops and iterations in real field

	2 × 2		Flops	4 × 4		Flops	8 × 8		Flops
	Average iterations			Average iterations			Average iterations		
	SR	CS	SR	CR	SR	CR			
LLL Algorithm	3.56	3.727	199.31	21.04	22.98	2190.75	122.98	106.83	20337
Hybrid-F & R-LLL	0.89	3.315	129.8	2.201	15.853	1061.83	2.16	56.94	6708.32
CE-fcLLL	1.772	2.011	176.17	10.78	12.7	1960.7	63.66	58.59	17921.04
GfcLLL-1	1	4	172	1.4	8	550	3	16	1693.4
GfcLLL-2	0.9	2.9	126.6	0.8	8	554.4	3.7	16	1421.6

From the statistical data in **Table 9**, GfcLLL-1 and GfcLLL-2 has the priority in complexity saving. These two algorithms use a traversal order based on a greedy selection strategy. It is motivated by increasing the success probability of the Babai point. GfcLLL-2 has different greedy selection strategy compared to GfcLLL-1. Between them, GfcLLL-2 can further reduce the complexity and the performance loss is not obvious. However, both of them performs bad in performance in **Fig. 7**. CE-fcLLL algorithm is designed under a new LLL loop and introducing new early termination conditions to reduce redundant and inefficient LR operations in fcLLL. However, CE-fcLLL algorithm needs full LLL loops and continually update the power factor in algorithm. During lots of iteration of LLL loops, the CE-fcLLL algorithm still experience the high level of computation. According to the **Table 9** and [17], it can just save little computation resource compared to LLL algorithm.



**Fig. 7.** BER Performance of LLL, Hybrid-F&R-LLL and families of fcLLL detectors in a 16QAM modulated 4 × 4 MIMO system

From the result plot in Fig. 7, the original LLL algorithm still displays the best performance of all. The performance gap between the LLL algorithm and the CE-fcLLL algorithm at  $\text{BER} = 10^{-5}$  is 1.34dB. CE-fcLLL can achieve a 0.88dB performance gain compared to Hybrid-F&R-LLL. GfcLLL-1 and GfcLLL-2 demonstrate performance loss of 1.73dB and 1.77dB respectively compared to CE-fcLLL at  $\text{BER} = 10^{-5}$ . Although the LLL and Hybrid version should set the parameter according to GfcLLL algorithm, the trend of BER is the same. In addition, in certain SNR, the value of BER may be a little bit different. The detailed BER of the different algorithms is shown in Table 10. Due to the mathematical expression of performance and algorithmic complexity have not been proposed in [17]-[19], the theoretical analysis is leaved to settle afterwards.

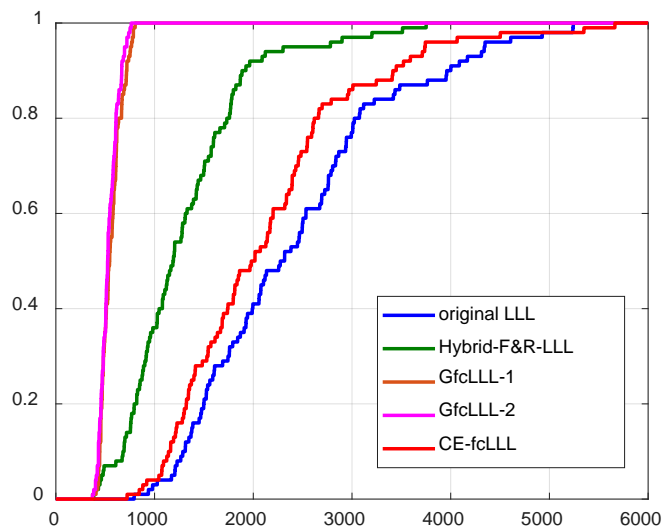


Fig. 8. Cumulative density function of the required flops to terminate the LLL, Hybrid-F&R-LLL and some new proposed algorithms for  $4 \times 4$  MIMO system

Table 10. BER of Different Algorithm Based on Original LLL from SNR = 18 to 30

	18	20	22	24	26	28	30
LLL Algorithm	3.00E-03	8.82E-04	1.46E-04	1.05E-05	1.25E-07	0	0
Hybrid-F&R-LLL	4.65E-03	1.92E-03	6.04E-04	1.33E-04	1.61E-05	3.75E-07	0
CE-fcLLL	4.83E-03	1.86E-03	5.00E-04	7.23E-05	4.13E-06	0	0
GfcLLL-1	8.93E-03	4.54E-03	1.74E-03	4.32E-04	5.56E-05	2.38E-06	0
GfcLLL-2	8.92E-03	4.55E-03	1.75E-03	4.34E-04	5.40E-05	2.63E-06	0

From the CDF function, this time, it is apparent that the GfcLLL-2 and GfcLLL-1 converges faster than other families of LLL algorithms. With nearly 700 flops, these two algorithms achieves almost 99% termination. However, according to Fig. 7 and Table 10, the recent achievements perform worst in performance. Additionally, for Hybrid-F&R-LLL, CE-fcLLL and original LLL, the value of flops that achieves almost 99% LLL basis reduction is 3800, 5500, and 6000, respectively. Therefore, it can derive the conclusion here that Hybrid-F&R-LLL still makes a better compromise between performance and algorithmic complexity. This conclusion is the same as in Section 8.3. It cannot both guarantee the characteristic of fast convergence and best performance the same time.

## 9. Conclusions

In this paper, a modified version of the LLL algorithm, the Hybrid-F&R-LLL is first proposed. The aim of designation of the Hybrid-F&R-LLL is that in each algorithm's iteration, the LLL potential can be reduced more than with any other. Thus, the algorithm directly changes the process of size reduction and the whole algorithm has a greater probability to skip size reduction. Additionally, deduce a performance bound for the Hybrid-F&R-LLL algorithm based on proximity factors. Its performance is poor contrasted with the original LLL algorithm. The theoretic upper bound on the average complexity and performance are also determined. Computer simulation results show that Hybrid-F&R-LLL algorithm can make a better compromise between performance and complexity. With some performance loss, the Hybrid-F&R-LLL algorithm can reduce more computation. At the same time, the Hybrid-F&R-LLL can produce faster converge than the original LLL algorithm. Compared to other new proposed algorithm, Hybrid-F&R-LLL can still holds a certain order of detection performance.

## Appendix A

### Proof of performance bound of Hybrid-F&R-LLL algorithm.

By the definition of Hybrid-F&R-LLL reduction based on Gram-Schmidt orthogonalization (21)-(22), it has:

$$\left| \hat{b}_i \right|^2 \geq \left( \delta - \mu_{i,i-1}^2 \right) \left| \hat{b}_{i-1} \right|^2 \geq (\delta - 1) \left| \hat{b}_{i-1} \right|^2 \quad \text{for } 1 < i \leq n \quad (39)$$

By induction in [6]:

$$\left| \hat{b}_j \right|^2 \leq \alpha^{i-j} \left| \hat{b}_i \right|^2 \quad \text{for } 1 \leq j < i \leq n \quad (40)$$

Where  $\alpha \triangleq \frac{1}{(\delta-1)} > -\frac{4}{3}$ . Substitute (40) into the Gram-Schmidt orthogonalization procedure:

$$\left| b_i \right|^2 = \left| \hat{b}_i \right|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \left| \hat{b}_j \right|^2 \quad (41)$$

Due to the fact that any two vectors following Gram-Schmidt orthogonalization are orthogonal, so obtain a new inequality:

$$\left| b_i \right|^2 \leq \left( 1 + \sum_{j=1}^{i-1} \frac{\alpha^{i-j}}{4} \right) \left| \hat{b}_i \right|^2$$

$$= \left( 1 + \frac{1}{4} \times \frac{\alpha^j - \alpha}{\alpha - 1} \right) \left| \hat{b}_i \right|^2 \quad (42)$$

For  $1 \leq j < i \leq n$ , replace index  $j$  and  $i$ , and substitute (40) with (42):

$$\left| b_j \right|^2 \leq \left( 1 + \frac{1}{4} \times \frac{\alpha^j - \alpha}{\alpha - 1} \right) \alpha^{i-j} \left| \hat{b}_i \right|^2 \quad (43)$$

According to the details of appendix B in [6], the equation (44) is established:

$$\left( 1 + \frac{1}{4} \times \frac{\alpha^j - \alpha}{\alpha - 1} \right) \leq \alpha^{j-1} \quad (44)$$

So, an exponential upper bound can be obtained:

$$\left| b_j \right|^2 \leq \alpha^{i-1} \left| \hat{b}_i \right|^2, \quad 1 \leq j \leq i \leq n \quad (45)$$

It follows immediately that:

$$\rho_{Hybrid-F\&R-LLL} \leq \alpha^{n-1} \quad (46)$$

Finally, build the relationship between parameter  $\alpha$  and  $\beta$ , and replace  $\alpha$  in (46) by the definition of  $\beta$ :

$$\rho_{Hybrid-F\&R-LLL} \leq \left( \frac{4\beta}{3\beta - 4} \right)^{n-1} \quad \left( \beta \triangleq \frac{1}{\delta - \frac{1}{4}} \right) \quad (47)$$

## Appendix B

### Proof of average complexity bound of Hybrid-F&R-LLL algorithm.

**Definition 7:** Define parameter  $B$  as the largest norm of basis and  $bb$  (in order to differ from the original basis  $b$ ) as the smallest norm of basis:

$$B = \max \left( \left| \hat{b}_1 \right|, \left| \hat{b}_2 \right|, \dots, \left| \hat{b}_n \right| \right) \quad (48)$$

$$bb = \min \left( \left| \hat{b}_1 \right|, \left| \hat{b}_2 \right|, \dots, \left| \hat{b}_n \right| \right) \quad (49)$$

The initial value of  $D$  can be bounded from above by  $B^{n(n-1)/2}$ . For an integral basis,  $D$  is one at least. Consequently, one has:



$$K^- \leq \frac{n(n-1)}{2} \log \frac{B}{bb} \quad (50)$$

where the logarithm is taken to the base  $\frac{1}{\delta}$ . All these narration is according to [10]. LLL algorithm tends to reduce the lengths of Gram-Schmidt vectors. The average complexity of the LLL reduction algorithm depends on the distribution of the random basis matrix  $R_r$ .

Updating the Gram-Schmidt orthogonalization coefficients during the column swap procedure costs  $6(n-k)+7 \leq 6n-5$  flops, whereas pair wise size reduction in the subroutine for  $(k, k-1)$  costs  $(2n+2(k-1))$ .

However, in the Hybrid-F&R-LLL Algorithm, the range of the size-reduced basis expands to  $\frac{|R_r(k-l, k)|}{|R_r(k-l, k-l)|} < 1$  and there is a low probability that the algorithm will complete the size

reduction procedure. When the entire basis satisfies  $\frac{|R_r(k-l, k)|}{|R_r(k-l, k-l)|} < 1$ , size reduction is directly ignored and may result in a fast convergence. This is the extreme situation. Thus, in this situation the total cost is:

$$\begin{aligned} C &\geq (6n-5)K^- + 0 \times (2n+2(k-1)) + 2n^3 \\ &\geq (6n-5) \frac{n(n-1)}{2} \log 6n + 2n^3 \\ &\geq 3n^3 \log 6n + 2n^3 \end{aligned} \quad (51)$$

The last term  $2n^3$  is the initialization of Gram-Schmidt orthogonalization vectors and parameter  $K^-$ , which is  $K^- \leq \frac{n(n-1)}{2} \log \frac{B}{bb}$ . This result assumed that the lowest average complexity bounds of the Hybrid-F&R-LLL Algorithm is  $O(3n^3 \log B)$ :

$$C_{Hybrid-F\&R-LLL} \geq O(3n^3 \log B) \quad (52)$$

For the worst case, all the basis vectors meets the condition  $\frac{|R_r(k-l, k)|}{|R_r(k-l, k-l)|} < 1$ . This will achieve the same procedure of E-LLL algorithm when size reduction happens at index  $(k, k-1)$ . So the highest average complexity bounds of Hybrid-F&R-LLL Algorithm is  $O(7n^3 \log B)$ :

$$C_{Hybrid-F\&R-LLL} \leq O(7n^3 \log B) \quad (53)$$

Introduce a parameter  $\lambda$  to balance the arithmetic cost between the standard E-LLL algorithm and the Hybrid-F&R-LLL Algorithm:

$$O(3n^3 \log B) \leq C_{\text{Hybrid-F\&R-LLL}} \leq O(7n^3 \log B) \quad (54)$$

## References

- [1] W. H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint," *IEEE Trans. Inf. Theory*, vol. 40, pp. 1591–1600, Sept. 1994. [Article \(CrossRef Link\)](#)
- [2] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1639–1642, July 1999. [Article \(CrossRef Link\)](#)
- [3] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. of Globecom'02*, Taipei, China, pp. 17–21, Nov. 2002. [Article \(CrossRef Link\)](#)
- [4] L. Babai, "On Lovász's lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, pp. 1–13, 1986. [Article \(CrossRef Link\)](#)
- [5] Lenstra AK, Lenstra HW, Lovász L Factoring polynomials with rational coefficients. *Math Ann* 261, pp. 515–534, 1982. [Article \(CrossRef Link\)](#)
- [6] C. Ling, "Towards characterizing the performance of approximate lattice decoding in MIMO communications," in *Proc. of International Symposium on Turbo Codes*, Munich, Germany, Apr. 3-7, 2006. [Article \(CrossRef Link\)](#)
- [7] Cong Ling, "On the Proximity Factors of Lattice Reduction-Aided Decoding," *IEEE Trans. Signal Processing*, vol.59 no.6, pp. 2795-2808, June 2011. [Article \(CrossRef Link\)](#)
- [8] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "LLL lattice-basis reduction achieves the maximum diversity in MIMO systems," in *Proc. of IEEE Int. Symp. Inf. Theory (ISIT)*, Adelaide, Australia, 2005. [Article \(CrossRef Link\)](#)
- [9] C. Ling. H. Mow, and N. Howgrave-Graham, "Variants of the LLL algorithm in digital communications: Complexity analysis and fixed complexity implementation," *Mathematics*, 2010, [Article \(CrossRef Link\)](#)
- [10] C. Ling and N. Howgrave-Graham, "Effective LLL reduction for lattice decoding," in *Proc. of IEEE Int. Symp. Inf. Theory (ISIT)*, Nice, France, Jun. 2007. [Article \(CrossRef Link\)](#)
- [11] Wen Zhang, Sanzheng Qiao, and Yimin Wei. HKZ and Minkowski, "A Diagonal Reduction Algorithms for MIMO Detection," *IEEE Transactions on Signal Processing*, vol. 60, No. 11, 2012. [Article \(CrossRef Link\)](#)
- [12] Gan YH, Ling C, Mow WH, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Trans Signal Process*, vol. 57, pp. 2701–2710, 2009. [Article \(CrossRef Link\)](#)
- [13] Ma X, Zhang W, "Performance analysis for MIMO systems with lattice-reduction aided linear equalization," *IEEE Trans Communication*, vol. 56, pp. 309–318, 2008. [Article \(CrossRef Link\)](#)
- [14] M. S. H. Vetter, V. Ponnampalamand and P. Hoeher, "Fixed complexity LLL algorithm," *IEEE Trans. Signal Process*, vol. 57, no. 4, pp. 1634–1637, Apr. 2009. [Article \(CrossRef Link\)](#)
- [15] Cong Ling, Wai Ho Mow, "Reduced and Fixed-Complexity Variants of the LLL Algorithm for Communications," *IEEE Trans. Communications*, vol.61 no.3, pp 1040-1050, March 2013. [Article \(CrossRef Link\)](#)
- [16] K. Zhao, Y. Li, H. Jiang, and S. Du, "A low complexity fast lattice reduction algorithm for MIMO detection," in *Proc. of IEEE Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, Sydney, Australia, Sep. 2012, pp. 1612–1616. [Article \(CrossRef Link\)](#)
- [17] Qingsong Wen, and Xiaoli Ma, "Efficient Greedy LLL Algorithm for Lattice Decoding," *IEEE Transactions on Wireless Communications*, vol.15,No.5, pp.3560-3572, May.2016. [Article \(CrossRef Link\)](#)
- [18] Wei Wang, Meixia Hu, Yongzhao Li, Hailin Zhang, Zan Li, "Computationally efficient fixed complexity LLL algorithm for lattice-reduction-aided multiple-input-multiple-output precoding," *IET Communications*, vol.10, No. 17, pp.2328-2335, July.2016. [Article \(CrossRef Link\)](#)

- [19] Jinming Wen, Xiao-Wen Chang, "A Greedy Selection-Based Approach for Fixed-Complexity LLL Reduction," *IEEE Communication Letters*, Vol.21, No.9, pp.1965-1968, Sep.2017. [Article \(CrossRef Link\)](#)
- [20] Dirk Wubben, Dominik Seethaler, Joakim Jalden, Gerald Matz, "Lattice Reduction," *IEEE Trans. Signal Processing Magazine*, vol.28 issue.3, pp 70-91, April 2011. [Article \(CrossRef Link\)](#)
- [21] L.Bai and J.Choi, "Low-Complexity-MIMO Detection. Springer Science Business Media," *LLC* 2012. [Article \(CrossRef Link\)](#)
- [22] Murray R. Bremner, "Lattice Basis Reduction: An Introduction to the LLL Algorithm and Its Applications," *CRC Press*, LLC, 2012.



**Huazhang Lv** He received his M.S. degree from Communication University of China (CUC), Beijing, in 2017. He received his B.S. degree from CUC, Beijing, in 2014. Now he works in 5G Innovation Center of Network Technology Research Institute of China Unicom, No.9 Shouti South Road, Beijing, 100048. His main work content is standardization of Mobile Edge-Computing (MEC) and 5G beam forming in ETSI and 3GPP. In addition, his main research field is LTE-Advanced L2/L3 layer protocol and 5G NR technology. At school, his devotes himself to the development error control coding and MIMO detection based on lattice reduction algorithm. He has published over 10 research papers indexed in EI and SCI journal.