A Diagonal Lattice Reduction Algorithm for MIMO Detection

Wen Zhang, Sanzheng Qiao, and Yimin Wei

***Abstract*—Recently, an efficient lattice reduction method, called the effective LLL (ELLL) algorithm, was presented for the detection of multiinput multioutput (MIMO) systems. In this letter, a novel lattice reduction criterion, called diagonal reduction, is proposed. The diagonal reduction is weaker than the ELLL reduction, however, like the ELLL reduction, it has identical performance with the LLL reduction when applied for the sphere decoding and successive interference cancelation (SIC) decoding. It improves the efficiency of the ELLL algorithm by significantly reducing the size-reduction operations. Furthermore, we present a greedy column traverse strategy, which reduces the column swap operations in addition to the size-reduction operations.**

***Index Terms*—Effective LLL algorithm, lattice reduction, LLL algorithm, MIMO systems.**

I. Introduction

LATTICE reduction has recently become a powerful tool for the detection of multiinput multioutput (MIMO) systems. For lattice-type modulation, the optimum maximum-likelihood (ML) decoding can be viewed as a *closest vector problem* (CVP) [1], [2], which can be solved exactly by the sphere decoding algorithm [3]. Also, there are many low-complexity approximate solvers, such as *zero-forcing* (ZF) decoding and *successive interference cancelation* (SIC) decoding [1], [4]. It is well known that lattice reduction can significantly improve the efficiency of the sphere decoding algorithm as well as the performance of approximate decoding algorithms [1], [2], [5]. Note that in some cases, such as the sphere decoding and SIC, it is sufficient to know only the information of the diagonal, rather than the whole matrix, of the upper triangular factor in the QR decomposition of the channel matrix. In [6], [7], a new reduction notion called *effective LLL* (ELLL) reduction, which is weaker than the LLL reduction but is sufficient for such applications, is presented.

In this letter, we propose a new lattice reduction criterion called *diagonal reduction*. The new reduction notion is even weaker than the ELLL reduction. Since it only imposes one single constraint on diagonal elements, it reduces the computational complexity. However, like the ELLL reduction, it achieves identical performance to the LLL reduction when applied to the sphere decoding and SIC. We present three algorithms based on the diagonal reduction. They differ mainly in the traverse of column exchanges. The first two algorithms adopt the traditional column traverse strategy [8], [9], and the fixed-structure column traverse strategy [10], respectively. The third algorithm uses a novel *greedy* column traverse strategy, which, in each iteration, firstly determines the urgency degree of column swap for all possible columns by applying a *decline rate* function, then selects the most urgent column to perform swap operations. Our simulation results show that the proposed three algorithms are more efficient than the ELLL algorithm [6], [7], since all of them reduce the number of size-reduction operations by about 50%. Moreover, the greedy column traverse strategy performs better than the previous two column traverse strategies, since it also reduces the number of column swap operations.

The rest of the letter is organized as follows. In Section II, we briefly introduce some background in lattice theory and the ELLL reduction. The new reduction criterion and its first two implementations are given in Section III. Section IV presents the greedy column traverse strategy. In Section V, we demonstrate our simulation results.

Notations: $\mathbf{H}^{\mathrm{H}}$ and $\det(\mathbf{H})$ denote the Hermitian conjugate transpose and the determinant of a matrix $\mathbf{H}$, $\Re(z)$ and $\Im(z)$ the real and imaginary parts of a complex number z, $\lfloor z \rceil$ the integer nearest to z, and $|z|$ the modulus $\sqrt{z\bar{z}}$ of z.

II. Lattice Basis Reduction

A. Some Basic Concepts

Given a matrix $\mathbf{H} \in \mathbb{C}^{m\times n} (n \leq m)$ of full column rank, then a *lattice* generated by $\mathbf{H}$ is defined by $L(\mathbf{H}) = \{\mathbf{Hz} : \mathbf{z} \in \mathcal{G}^n\}$, where $\mathcal{G}^n$ is the set of complex integer n-vectors $\mathcal{G}^n = \mathbb{Z}^n + i\mathbb{Z}^n$. The columns of $\mathbf{H}$ form a *basis* for the lattice $L(\mathbf{H})$. A matrix $\mathbf{Z} \in \mathcal{G}^{n\times n}$ is called *unimodular* if $|\det(\mathbf{Z})| = 1$. The columns of any matrix $\mathbf{H}'$ can form a basis for $L(\mathbf{H})$ if and only if there exists a unimodular matrix $\mathbf{Z}$ such that $\mathbf{H}' = \mathbf{HZ}$. A *lattice reduction algorithm* finds a unimodular matrix $\mathbf{Z}$ such that the columns of $\mathbf{HZ}$ are reasonably short in the sense of Euclidean length. Lattice reduction has now become a powerful tool for improving the performance of decoding algorithms, since it can significantly reduce the orthogonality defect and the condition number of the channel matrix [2]. It is proved

Manuscript received November 18, 2011; accepted March 10, 2012. Date of publication March 21, 2012; date of current version April 03, 2012. S. Qiao was supported in part by the Shanghai Key Laboratory of Contemporary Applied Mathematics. Y. Wei was supported by the National Natural Science Foundation of China under Grant 10871051, the Doctoral Program of the Ministry of Education under Grant 20090071110003, the 973 Program Project under Grant 2010CB327900, and by the Shanghai Education Committee (Dawn Project). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Eduard A. Jorswieck.

W. Zhang is with the School of Mathematical Sciences, Fudan University, Shanghai 200433, China (e-mail: zhangwen9801@gmail.com).

S. Qiao is with the Department of Computing and Software, McMaster University, Hamilton, ON L8S 4K1 Canada.

Y. Wei is with the School of Mathematical Sciences and Shanghai Key Laboratory of Contemporary Applied Mathematics, Fudan University, Shanghai, 200433 China.

Digital Object Identifier 10.1109/LSP.2012.2191614

in [5], [11] that LLL-reduction-aided decoding can achieve the full diversity order of a MIMO fading channel.

B. Effective LLL Reduction

Given a lattice generator matrix $\mathbf{H}$ and its QR decomposition $\mathbf{H} = \mathbf{QR}$, where $\mathbf{Q}$ has orthonormal columns and $\mathbf{R}$ is an upper triangular matrix with positive diagonal. Then from [5], [12] the efficiency of sphere decoding and the performance of SIC decoding is determined by the arrangement of the diagonal elements of $\mathbf{R}$. Based on this fact, a notion of effective LLL (ELLL) reduction, which is weaker than the LLL reduction, was proposed in [6] and generalized to the complex case in [7] to improve efficiency.

Definition 1 (Effective LLL Reduction [7]): A basis $\mathbf{H} \in \mathbb{C}^{m\times n}$ is effective LLL reduced with parameter $\omega(1/2 < \omega < 1)$, if the upper triangular factor $\mathbf{R} \triangleq [r_{i,j}]$ in its QR decomposition $\mathbf{H} = \mathbf{QR}$ satisfies

$$|\Re(r_{k-1,k})| \le \frac{1}{2}|r_{k-1,k-1}|,$$
$$|\Im(r_{k-1,k})| \le \frac{1}{2}|r_{k-1,k-1}|, \tag{1}$$

and

$$|r_{k-1,k}|^2 + r_{k,k}^2 \ge \omega r_{k-1,k-1}^2 \tag{2}$$

for all $1 < k \le n$.

The conditions in (1) and (2) are said to be *size-reduced* and *Lovász-reduced*, respectively. They loosely impose an ascending order on the diagonal elements of $\mathbf{R}$. The larger the parameter ω is, the better quality the reduced basis has. The ELLL algorithm, which adopts the traditional column traverse strategy [8], can be found in [6], [7].

III. Diagonal Reduction

A. Diagonal Reduction

From Definition 1, the ELLL reduction weakens the LLL reduction by removing requirement for the size-reduction of all off-diagonal elements. However, note that for sphere decoding and SIC, the size-reduction constraint (1) on all sub-diagonal elements $r_{k-1,k}$ is also unnecessary. Based on this fact, we propose an even weaker reduction criterion, which only requires the reduction of diagonal elements, to further improve the efficiency.

Definition 2 (Diagonal reduction): A basis $\mathbf{H} \in \mathbb{C}^{m\times n}$ is diagonal reduced with parameter $\omega(1/2 < \omega < 1)$, if the upper triangular factor $\mathbf{R}$ in its QR decomposition $\mathbf{H} = \mathbf{QR}$ satisfies

$$|r_{k-1,k} - \mu_k r_{k-1,k-1}|^2 + r_{k,k}^2 \ge \omega r_{k-1,k-1}^2 \tag{3}$$

for all $1 < k \le n$, where $\mu_k = \lfloor r_{k-1,k}/r_{k-1,k-1}\rceil$ [13].

It is easy to verify that if $\mathbf{H}$ is ELLL reduced, then it must be diagonal reduced, while the converse is not true. For the quality of the diagonal reduction, we can derive from (3) that

$$\begin{aligned} r_{k,k}^2 &\ge \omega r_{k-1,k-1}^2 - |r_{k-1,k} - \mu_k r_{k-1,k-1}|^2 \\ &\ge (\omega - 1/2) r_{k-1,k-1}^2 \end{aligned} \tag{4}$$

for all $1 < k \le n$. By induction, we obtain

$$r_{j,j}^2 \le \beta^{i-j} r_{i,i}^2, \quad \text{for} \quad 1 \le j < i \le n \tag{5}$$

where $\beta = 1/(\omega - 1/2)$. Note that (5) also holds for the ELLL and is used to show that ELLL has the same performance as LLL in SIC decoding [5], [12]. It then follows that diagonal-reduction-aided SIC decoding has the same performance as LLL-reduction-aided SIC decoding.

```
Input: $\mathbf{Q} \in \mathbb{C}^{m\times n}$, $\mathbf{R} \in \mathbb{C}^{n\times n}$, $\omega$
Output: Updated $\mathbf{Q}$ and updated $\mathbf{R}$ that is diagonal reduced with
    parameter $\omega$ and a unimodular $\mathbf{Z}$ that reduces $\mathbf{R}$
 1: Initialization $\mathbf{Z} \leftarrow \mathbf{I}_n$
 2: $k \leftarrow 2$
 3: while $k \le n$ do
 4:     $\mu_k \leftarrow \lfloor \mathbf{R}(k-1,k)/\mathbf{R}(k-1,k-1) \rceil$
 5:     if the condition (3) is not satisfied then
 6:         if $\mu_k \neq 0$ then
 7:             $\mathbf{R}(1:k-1,k) \leftarrow \mathbf{R}(1:k-1,k) - \mu_k \mathbf{R}(1:k-1,k-1)$
 8:             $\mathbf{Z}(:,k) \leftarrow \mathbf{Z}(:,k) - \mu_k \mathbf{Z}(:,k-1)$
 9:         end if
10:         swap columns $k-1$ and $k$ in $\mathbf{R}$ and $\mathbf{Z}$
11:         find a Givens rotation $\mathbf{G}$ to restore the upper triangular
            structure of $\mathbf{R}$
12:         $\mathbf{R}(k-1:k, k-1:n) \leftarrow \mathbf{G}\mathbf{R}(k-1:k, k-1:n)$
13:         $\mathbf{Q}(:, k-1:k) \leftarrow \mathbf{Q}(:, k-1:k)\mathbf{G}^{\mathrm{H}}$
14:         $k \leftarrow \max(k-1, 2)$
15:     else
16:         $k \leftarrow k+1$
17:     end if
18: end while
```

Fig. 1. Diagonal reduction algorithm.

B. Two Implementations

The proposed weaker reduction criterion provides a possible implementation with potentially lower computational cost than the ELLL algorithm. Using the traditional column traverse strategy [6], [8], a generic implementation of diagonal reduction can be found in Fig. 1. In each iteration of the while-loop, the *diagonal reduction* (DR) algorithm firstly tests the diagonal reduction condition (3) in line 5, and then performs a selective size-reduction step (lines 6–9) followed by a column swap step (lines 10-13) in order to fulfill (3). In the case of a column swap, the algorithm backtracks one step ($k \leftarrow k-1$). It iterates until $\mathbf{R}$ satisfies the diagonal reduction criterion. In comparison to the ELLL algorithm, the DR algorithm reduces the number of size-reduction operations. Specifically, in each iteration, the ELLL algorithm performs a size-reduction whenever the size-reduction condition (1) is not fulfilled, whereas, in the DR algorithm, size-reduction is not performed unless the diagonal reduction condition (3) (line 5 of Fig. 1) and the size-reduction condition (1) (line 6 of Fig. 1) are not satisfied simultaneously.

In [7], [10], a fixed-structure column traverse strategy is proposed for the ELLL algorithm, resulting the *parallel effective LLL* (PELLL) algorithm. Analogously, by employing the fixed-structure column traverse strategy, a *modified diagonal reduction* (MDR) algorithm is shown in Fig. 2.

Let $\mathbf{R}$ be the upper triangular factor of the QR decomposition of $\mathbf{H}$, and denote M_1 and M_2 the minimum and maximum squared diagonal elements of $\mathbf{R}$. Then following [6], we can

```
Input: $\mathbf{Q} \in \mathbb{C}^{m \times n}$, $\mathbf{R} \in \mathbb{C}^{n \times n}$, $\omega$
Output: Updated $\mathbf{Q}$ and updated $\mathbf{R}$ that is diagonal reduced with
    parameter $\omega$ and a unimodular $\mathbf{Z}$ that reduces $\mathbf{R}$
Initialization $\mathbf{Z} \leftarrow \mathbf{I}_n$
while $\mathbf{R}$ is not diagonal reduced do
    for $k = 2, 3, ..., n$ do
        execute selective size-reduction followed by column swap
        and Givens rotation (lines 4-13 of Fig. 1)
    end for
end while
```

Fig. 2. Modified diagonal reduction algorithm.

prove that both the DR algorithm and the MDR algorithm terminate in at most $O(n^2 \log(M_2)/(M_1))$ iterations of the while-loop. For completeness, we recall the main steps of the proof. Consider the products

$$d_i = \prod_{j=1}^{i} r_{j,j}^2, \quad i = 1, 2, \ldots, n-1$$

and denote

$$D = \prod_{i=1}^{n-1} d_i. \tag{6}$$

Then each time when a column swap is performed, the corresponding d_i will be reduced by a factor smaller than $\omega < 1$, while the other products $d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_{n-1}$ remain unchanged. Consequently, D will be reduced by a factor smaller than ω. On the other hand, following [6], we can prove that the squared diagonal elements of $\mathbf{R}$ remain in the range $[M_1, M_2]$ throughout the execution of the algorithms. Thus, the lower and upper bounds of D are $M_1^{n(n-1)/2}$ and $M_2^{n(n-1)/2}$ respectively, which implies that the number of column swaps is at most $O(n^2 \log(M_2)/(M_1))$ and the proof is completed.

IV. A Greedy Diagonal Reduction Algorithm

From the complexity analysis in Section III-B, the efficiency of a diagonal reduction algorithm depends on the rate of decline in the size of D: the faster decline in the size of D, the fewer iterations are required and thus the more efficient the algorithm is. Therefore, it is desirable to find an appropriate column traverse strategy such that D can be reduced as quickly as possible. In the following, we propose a novel column traverse strategy, which can achieve such a goal in a greedy manner. That is, in each iteration, the new strategy selects a column from all possible columns for performing selective size-reduction and column swap, so that the decline in D is maximized.

Specifically, we define the *decline rate*:

$$\gamma_j = \frac{|r_{j-1,j} - \mu_j r_{j-1,j-1}|^2 + r_{j,j}^2}{r_{j-1,j-1}^2} \tag{7}$$

for each j: $1 < j \leq n$. In each iteration, we find k such that $\gamma_k = \min_j \gamma_j$. Suppose that $\gamma_k < \omega$, then we perform selective size-reduction and swap the columns $k-1$ and k. Accordingly, d_{k-1} is reduced to $\gamma_k d_{k-1}$, while the other products $d_1, \ldots, d_{k-2}, d_k, \ldots, d_{n-1}$ remain unchanged. Thus, D is reduced to $\gamma_k D$, achieving the maximum decline in the size of D among all possible column swaps. After that, we need to update the decline rates γ_j, which can be finished by just updating γ_{k-1} (if $k < 2$), γ_k, and γ_{k+1} (if $k < n$), since among the entries modified by the column swap only $r_{k-2,k-1}, r_{k-1,k-1}, r_{k-1,k}$, and $r_{k,k}$ are associated with the decline rates γ_j. The procedure is repeated until $\gamma_j \geq \omega$, for all $1 < j \leq n$, which means that the basis has been completely diagonal reduced. Based on such column traverse strategy, the *greedy diagonal reduction* (GDR) algorithm can be found in Fig. 3.

```
Input: $\mathbf{Q} \in \mathbb{C}^{m \times n}$, $\mathbf{R} \in \mathbb{C}^{n \times n}$, $\omega$
Output: Updated $\mathbf{Q}$ and updated $\mathbf{R}$ that is diagonal reduced with
    parameter $\omega$ and a unimodular $\mathbf{Z}$ that reduces $\mathbf{R}$
Initialization $\mathbf{Z} \leftarrow \mathbf{I}_n$
using (7) to compute $\gamma_j$, $1 < j \leq n$
$k \leftarrow \arg\min_{1<j\leq n} \gamma_j$
while $\gamma_k < \omega$ do
    execute selective size-reduction followed by column swap and
    Givens rotation (lines 4-13 of Fig. 1)
    update $\gamma_k$, $\gamma_{k-1}$ (if $k > 2$), and $\gamma_{k+1}$ (if $k < n$) using (7)
    $k \leftarrow \arg\min_{1<j\leq n} \gamma_j$
end while
```

Fig. 3. Greedy diagonal reduction algorithm.

Note that the greedy column traverse strategy is also applicable to the ELLL reduction, since any implementation of the diagonal reduction can be easily extended to the ELLL reduction. The *greedy effective LLL* (GELLL) algorithm can be immediately obtained by adding "size-reduce all the sub-diagonal elements of $\mathbf{R}$" in the end of Fig. 3.

V. Simulation Results

In this section, we present our simulation results on comparing the efficiency of the proposed new reduction algorithms with the ELLL algorithm. All experiments were performed on matrices with random entries, drawn from an independently and identically zero-mean, unit variance complex Gaussian distribution. For each size, we generated 1000 random matrices and took an average. The parameter ω in the reduction algorithms was set to 0.99.

Fig. 4 depicts our results on the numbers of size-reduction operations and column swap operations performed by the reduction algorithms. The curves at the bottom show that the DR, MDR, and GDR algorithms performed about the same number of size-reduction operations, which is about 50% of the size-reduction operations performed by the ELLL. The numbers of the size-reduction operations performed by the PELLL and GELLL algorithms are roughly 85% and 70%, respectively, of that performed by the ELLL. For the column swap operations, since ELLL and DR, same between PELLL and MDR or between GELLL and GDR, use the same column traverse strategy, they performed the same number of column swap operations. As shown by the curves at the top of Fig. 4, the fixed-structure column traverse strategy reduces the number of column swap operations and the greedy strategy further reduces the column swap operations.

Although the greedy strategy reduces the number of column swap operations, it introduces overhead, such as the computation of the decline rates. To compare the overall complexity of

TABLE I
THE AVERAGE COMPLEXITY (IN FLOPS) OF THE ELLL, PELLL, GELLL, DR, MDR, AND GDR ALGORITHMS

n	ELLL	PELLL	GELLL	DR	MDR	GDR
2	128.46	129.46	132.01	117.40	118.40	113.86
3	517.90	506.70	505.22	477.62	473.52	457.27
4	1280.21	1226.32	1174.61	1189.82	1152.33	1085.71
6	4301.27	3999.46	3658.17	4013.27	3778.62	3457.06
8	9726.91	8912.25	7893.39	9097.26	8432.78	7531.62
10	16915.37	15449.68	13621.81	15823.74	14607.59	13070.11

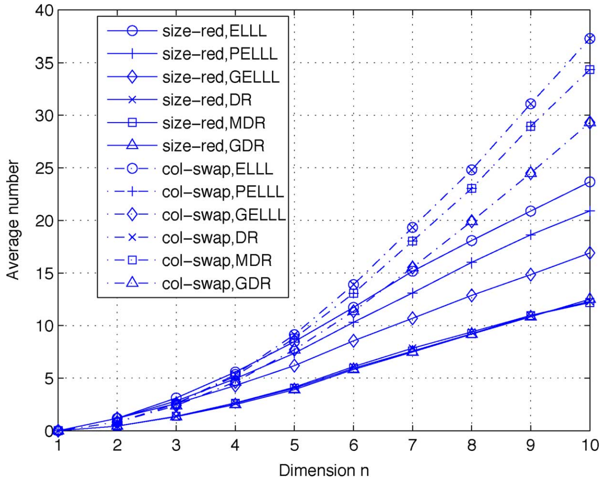

Fig. 4. Comparison of size-reduction and column swap operations performed by the ELLL, PELLL, GELLL, DR, MDR, and GDR algorithms.

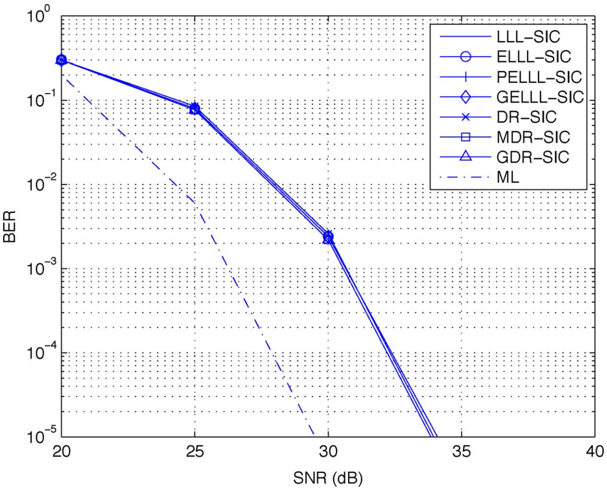

Fig. 5. Simulated BER of SIC aided by the LLL, ELLL, and diagonal reduction for 64-QAM over an 8 × 8 MIMO fading channel.

the algorithms, we experimented on the floating-point operations (flops)[1] carried out by the algorithms. Table I shows that in both cases of ELLL and DR, the greedy column traverse strategy improves the overall computation and the results are consistent with Fig. 4. This implies that the overhead introduced by the greedy strategy is insignificant.

We also investigated the quality of the reduced bases produced by our algorithms in terms of the bit-error-rate (BER) performance of SIC decoding. Specifically, using a 64-QAM constellation, Fig. 5 depicts the simulated BER curves of lattice-reduction-aided SIC over an 8 × 8 uncoded MIMO fading channel. We found that SIC aided by the six reduction algorithms had identical BER performance with that aided by the LLL algorithm. This is consistent with the quality analysis presented in Section III-A.

VI. CONCLUSION

In this letter, we propose a new lattice reduction criterion called diagonal reduction. The new reduction notion is weaker than the ELLL reduction, but has identical performance with the LLL reduction when applied to SIC decoding. We also present a a greedy column traverse strategy to enhance the performance. Simulation results show that the diagonal reduction notion can reduce the number of size-reduction operations by about 50% when compared with the ELLL algorithm. Moreover, the greedy column traverse strategy can reduce the number of column swap operations.

REFERENCES

[1] W. H. Mow, "Universal lattice decoding: Principle and recent advances," *Wireless Commun. Mobile Comput., Special Issue on Coding and Its Application toWireless CDMA Systems*, vol. 3, pp. 553–569, Aug. 2003.

[2] D. Wübben, D. Seethaler, J. Jaldén, and G. Marz, "Lattice reduction: a survey with applications in wireless communications," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 70–91, May 2011.

[3] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[4] L. Babai, "On Lovász's lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, pp. 1–13, 1986.

[5] C. Ling, "On the proximity factors of lattice reduction-aided decoding," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2795–2808, Jun. 2011.

[6] C. Ling and N. Howgrave-Graham, "Effective LLL reduction for lattice decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Nice, France, Jun. 2007.

[7] C. Ling, W. H. Mow, and N. Howgrave-Graham, "Variants of the LLL algorithm in digital communications: Complexity analysis and fixed-complexity implementation," *IEEE Trans. Inf. Theory* [Online]. Available: http://arxiv.org/abs/1006.1666, submitted for publication

[8] A. K. Lenstra, H. Lenstra, Jr., J. , and L. Lovász, "Factorizing polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515–534, Dec. 1982.

[9] Y. H. Gan, C. Ling, and H. M. Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2701–2710, July 2009.

[10] H. Vetter, V. Ponnampalam, M. Sandell, and P. A. Hoeher, "Fixed complexity LLL algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1634–1637, 2009.

[11] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "LLL reduction achieves the receive diversity in MIMO decoding," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4801–4805, Dec. 2007.

[12] C. Ling, "Towards characterizing the performance of approximate lattice decoding," in *Proc. Int. Symp. Turbo Codes/Int. Conf. Source Channel Coding'06*, Munich, Germany, Apr. 2006.

[13] W. Zhang, Y. Wei, and S. Qiao, A Delayed Size-Reduction Technique for Speeding Up the LLL Algorithm Dept. Comput. Software, McMaster Univ., Hamilton, ON, Canada, Tech. Rep. CAS-11-02-SQ, 2011.

[1] Flop count: Complex multiplication, 6 flops; complex addition, 2 flops; complex rounding, 2 flops; real addition/multiplication/division/max, 1 flop.