Research Article

Loop-Reduction LLL Algorithm and Architecture for Lattice-Reduction-Aided MIMO Detection

Chun-Fu Liao,¹ Li-Wei Chai,² and Yuan-Hao Huang^{1,2}

¹ Department of Electrical Engineering, National Tsing-Hua University, Hsinchu 30013, Taiwan

² Institute of Communications Engineering, National Tsing-Hua University, Hsinchu 30013, Taiwan

Correspondence should be addressed to Yuan-Hao Huang, yhhuang@ee.nthu.edu.tw

Received 7 July 2011; Revised 30 September 2011; Accepted 14 October 2011

Academic Editor: Yin-Tsung Hwang

Copyright © 2012 Chun-Fu Liao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a loop-reduction LLL (LR-LLL) algorithm for lattice-reduction-aided (LRA) multi-input multioutput (MIMO) detection. The LLL algorithm is an iterative algorithm that contains many check and process operations; however, the traditional LLL algorithm itself possesses a lot of redundant check operations. To solve this problem, we propose a look-ahead check technique that not only reduces the complexity of the LLL algorithm but also produces the lattice-reduced matrix which obeys the original LLL criterion. Simulation results show that the proposed LR-LLL algorithm reduces the average number of loops or computation complexity. Besides, it also shortens the latency of clock cycles about 19.4%, 29.1%, and 46.1% for 4×4 , 8×8 , and 12×12 MIMO systems, respectively.

1. Introduction

To increase the transmission capacity, multiple-input multiple-output (MIMO) system has been proposed for the next generation wireless communication systems, and therefore the need for a high-performance and low-complexity MIMO detector becomes an important issue. The maximum likelihood (ML) detector is known to be an optimal detector; however, it is impractical for realization owing to its great computational complexity. Addressing this problem, researchers have proposed tree-based search algorithms, such as sphere decoding [1] and K-Best decoding [2], to reduce the complexity with near-optimal performance. On the other hand, channel matrix preprocessing technique, such as lattice-reduction-aided (LRA) detection [3], has been proposed to improve the MIMO detection performance.

The lattice reduction transforms the channel matrix into a more orthogonal one by finding a better basis for the same lattice so as to improve the diversity gain of the MIMO detector. The Lenstra-Lenstra-Lovász (LLL) algorithm is a well-known lattice reduction algorithm for its polynomial execution time. In the literature [4], the LLL algorithm is widely employed to improve the lattice-reduction MIMO detection or to reduce the MIMO detection complexity. However, the LLL algorithm has many redundant check

operations that have never been addressed in the literature. These redundant operations lead to many unnecessary computations and thus increase the processing latency and complexity. Therefore, we propose a look-ahead check technique to detect and avoid the unnecessary check operations in the LLL algorithm. This technique not only generates the lattice-reduced matrix which obeys the size reduction and LLL reduction in the original LLL algorithm but also applies to real- and complex-value LLL algorithm [5].

The remainder of this paper is organized as follows. Section 2 briefly describes the signal model for MIMO detection. In Section 3, we introduce the lattice-reduction-aided MIMO detection and the LLL algorithm. In Section 4, we demonstrate the proposed LR-LLL algorithm, and in Section 5 we present the simulation and analysis results. The corresponding hardware architecture and processing cycle counts estimation is shown in Section 6. Finally, we summarize our conclusions in Section 7.

2. System Model

A narrow-band $N_r \times N_t$ MIMO system consisting of N_t transmitters and N_r receivers can be modeled by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (1)$$

```

INPUT:  $\mathbf{Q}, \mathbf{R}$ , such that  $\mathbf{H} = \mathbf{QR}$ 
OUTPUT:  $\bar{\mathbf{Q}}, \bar{\mathbf{R}}, \mathbf{T}$  such that  $\mathbf{HT} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$ 
(1) Initialize  $\bar{\mathbf{Q}} = \mathbf{Q}, \bar{\mathbf{R}} = \mathbf{R}, \mathbf{T} = \mathbf{I}_N$ 
(2)  $k = 2$ 
(3) while  $k \leq N$ 
(4)   for  $p = k - 1, \dots, 1$ 
(5)      $\mu = \lceil \bar{\mathbf{R}}(p, k) / \bar{\mathbf{R}}(p, p) \rceil$ 
(6)     if  $\mu \neq 0$ 
(7)        $\bar{\mathbf{R}}(1 : p, k) \leftarrow \bar{\mathbf{R}}(1 : p, k) - \mu \bar{\mathbf{R}}(1 : p, p)$ 
(8)        $\mathbf{T}(:, k) \leftarrow \mathbf{T}(:, k) - \mu \mathbf{T}(:, p)$ 
(9)     end
(10)  end
(11)  if  $\delta |\bar{\mathbf{R}}(k - 1, k - 1)|^2 > |\bar{\mathbf{R}}(k, k)|^2 + |\bar{\mathbf{R}}(k - 1, k)|^2$ 
(12)    swap columns  $k - 1$  and  $k$  in  $\bar{\mathbf{R}}$  and  $\mathbf{T}$ 
(13)    calculate Givens rotation matrix  $\Theta = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$ 
            $a = \frac{\bar{\mathbf{R}}(k - 1, k - 1)}{\|\bar{\mathbf{R}}(k - 1 : k, k - 1)\|}$  and  $b = \frac{\bar{\mathbf{R}}(k, k - 1)}{\|\bar{\mathbf{R}}(k - 1 : k, k - 1)\|}$ 
(14)     $\bar{\mathbf{R}}(k - 1 : k, k - 1 : N) \leftarrow \Theta \bar{\mathbf{R}}(k - 1 : k, k - 1 : N)$ 
(15)     $\bar{\mathbf{Q}}(:, k - 1 : k) \leftarrow \bar{\mathbf{Q}}(:, k - 1 : k) \Theta^T$ 
(16)     $k \leftarrow \max\{k - 1, 2\}$ 
(17)  else
(18)     $k \leftarrow k + 1$ 
(19)  end
(20) end

```

ALGORITHM 1: LLL algorithm.

where $\mathbf{x} \in \mathbb{A}^{N_t}$ is the transmitted signal vector, $\mathbf{y} \in \mathbb{C}^{N_r}$ is the received signal vector, $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{N_t}]$ represents a flat-fading channel matrix, and $\mathbf{n} \in \mathbb{C}^{N_r}$ is the white Gaussian noise with variance σ_n^2 . All the vectors $\mathbf{h}_i$ are independent and identically distributed complex Gaussian random vectors with zero means and unity variances. Set $\mathbb{A}$ consists of the constellation points of the QAM modulation. Then, we re-formulate the equivalent real channel matrix as follows:

$$\mathbf{y}_r = \begin{bmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{bmatrix} = \mathbf{H}_r \mathbf{x}_r + \mathbf{n}_r$$

$$= \begin{bmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{bmatrix} \begin{bmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \Re(\mathbf{n}) \\ \Im(\mathbf{n}) \end{bmatrix}. \quad (2)$$

Then, the dimension of $\mathbf{H}_r$ becomes $N \times M$, where $M = 2N_t$ and $N = 2N_r$. The vectors $\mathbf{y}_r$ and $\mathbf{n}_r$ belong to $\mathbb{R}^N$ and $\mathbf{x}_r \in \mathbb{A}^M$.

The QR decomposition is often applied in the pre-processing of the MIMO detection because it provides decoding efficiency. Then, the channel matrix $\mathbf{H}_r$ can be expressed by

$$\mathbf{H}_r = \mathbf{Q}_r \mathbf{R}_r, \quad (3)$$

where $\mathbf{Q}_r \in \mathbb{R}^{N \times M}$ is an orthogonal matrix and $\mathbf{R}_r \in \mathbb{R}^{M \times M}$ is an upper triangular matrix. By multiplying $\mathbf{Q}_r^T$ on both sides of (2), we can obtain

$$\hat{\mathbf{y}}_r = \mathbf{Q}_r^T \mathbf{y}_r = \mathbf{R}_r \mathbf{x}_r + \mathbf{Q}_r^T \mathbf{n}_r, \quad (4)$$

where $\mathbf{Q}_r^T \mathbf{n}_r$ is white Gaussian. In addition, we adopt column-norm-based sorted QR decomposition (SQRD) [6] because it not only enhances detection performance but also reduces the computational complexity of the lattice reduction [7].

3. Lattice Reduction

A lattice $\mathbb{L}$ is defined as $\{t_1 \mathbf{h}_{r1} + t_2 \mathbf{h}_{r2} + \dots + t_N \mathbf{h}_{rN} \mid t_1 \dots t_N \in \mathbb{Z}\}$, where $\{\mathbf{h}_{r1}, \dots, \mathbf{h}_{rN} \in \mathbb{R}^N\}$ are the basis vectors. The lattice reduction algorithm aims to find a unimodular matrix $\mathbf{T}$ ($|\det \mathbf{T}| = 1$ and all elements of $\mathbf{T}$ are integers) such that a more orthogonal $\hat{\mathbf{H}}_r = \mathbf{H}_r \mathbf{T}$ has the same lattice as $\mathbf{H}_r$. Then, the signal model becomes

$$\mathbf{y}_r = \mathbf{H}_r \mathbf{x}_r + \mathbf{n}_r = \hat{\mathbf{H}}_r \mathbf{T}^{-1} \mathbf{x}_r + \mathbf{n}_r = \hat{\mathbf{H}}_r \mathbf{s} + \mathbf{n}_r. \quad (5)$$

If $\mathbf{x}_r \in \mathbb{Z}^N$, $\mathbf{T}^{-1} \mathbf{x}_r = \mathbf{s} \in \mathbb{Z}^N$. In practice, the transmitted signals $\mathbf{x}_r$ do not belong to an integer set; however, we can still transform the signals $\mathbf{x}_r \in \mathbb{A}^N$ into an integer set by linear operations such as scaling and shifting.

Several lattice-reduction algorithms are described in the literature, and the LLL algorithm [11] is the most popular approach. Because QR preprocessing is often employed in the MIMO detector, the LLL algorithm is then modified for $\mathbf{Q}$ and $\mathbf{R}$ matrices [12], as shown in Algorithm 1. In the literature, lines (4) to (19) are often defined as a loop that can be decomposed into two parts: (1) lines (4) to (10) deals with the size reduction operations; and (2) lines (11) to (19) handle LLL reduction operations. The number of iterations performed in the size reduction depends on the index k , and the LLL reduction operation may increase or decrease the index k depending on the result of the LLL reduction check ($\delta |\bar{\mathbf{R}}(k - 1, k - 1)|^2 > |\bar{\mathbf{R}}(k, k)|^2 + |\bar{\mathbf{R}}(k - 1, k)|^2$). Therefore, the number of loops certainly depends on the values in the $\mathbf{R}$ matrix, and thus the processing latency varies for different channel matrices. Moreover, we find that most of the computational complexity is contributed by the operations when the check conditions ($\mu \neq 0$ for size reduction and $\delta |\bar{\mathbf{R}}(k - 1, k - 1)|^2 > |\bar{\mathbf{R}}(k, k)|^2 + |\bar{\mathbf{R}}(k - 1, k)|^2$ for LLL reduction check) are satisfied; that is, the size and

TABLE 1: The percentage of no operation after index k is decreased.

Antenna number ($N \times N$)	$N = 2$	$N = 4$	$N = 6$	$N = 8$	$N = 10$	$N = 12$	$N = 14$	$N = 16$
Percentage of no operation	67%	40%	33%	30%	29%	28%	28%	28%

```

INPUT:  $\mathbf{Q}, \mathbf{R}$ , such that  $\mathbf{H} = \mathbf{QR}$ 
OUTPUT:  $\bar{\mathbf{Q}}, \bar{\mathbf{R}}, \mathbf{T}$  such that  $\mathbf{HT} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$ 
(1) Initialize  $\bar{\mathbf{Q}} = \mathbf{Q}, \bar{\mathbf{R}} = \mathbf{R}, \mathbf{T} = \mathbf{I}_N$ 
(2)  $k = 2$ 
(3) while  $k \leq N$ 
(4)   for  $p = k - 1, \dots, 1$ 
(5)      $\mu = \lceil \bar{\mathbf{R}}(p, k) / \bar{\mathbf{R}}(p, p) \rceil$ 
(6)     if  $\mu \neq 0$ 
(7)        $\bar{\mathbf{R}}(1 : p, k) \leftarrow \bar{\mathbf{R}}(1 : p, k) - \mu \bar{\mathbf{R}}(1 : p, p)$ 
(8)        $\mathbf{T}(:, k) \leftarrow \mathbf{T}(:, k) - \mu \mathbf{T}(:, p)$ 
(9)     end
(10)   end
(11)   if  $\delta |\bar{\mathbf{R}}(k - 1, k - 1)|^2 > |\bar{\mathbf{R}}(k, k)|^2 + |\bar{\mathbf{R}}(k - 1, k)|^2$ 
(12)     do
(13)       swap columns  $k - 1$  and  $k$  in  $\bar{\mathbf{R}}$  and  $\mathbf{T}$ 
(14)       calculate Givens rotation matrix  $\Theta = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$ 
(15)        $a = \frac{\bar{\mathbf{R}}(k - 1, k - 1)}{\|\bar{\mathbf{R}}(k - 1 : k, k - 1)\|}$  and  $b = \frac{\bar{\mathbf{R}}(k, k - 1)}{\|\bar{\mathbf{R}}(k - 1 : k, k - 1)\|}$ 
(16)        $\bar{\mathbf{R}}(k - 1 : k, k - 1 : N) \leftarrow \Theta \bar{\mathbf{R}}(k - 1 : k, k - 1 : N)$ 
(17)        $\bar{\mathbf{Q}}(:, k - 1 : k) \leftarrow \bar{\mathbf{Q}}(:, k - 1 : k) \Theta^T$ 
(18)        $k \leftarrow k - 1$ 
(19)     while  $(\delta |\bar{\mathbf{R}}(k - 1, k - 1)|^2 > |\bar{\mathbf{R}}(k, k)|^2 + |\bar{\mathbf{R}}(k - 1, k)|^2 \& \& k \neq 1)$ 
(20)       if  $\lceil \bar{\mathbf{R}}(k, k + 1) / \bar{\mathbf{R}}(k, k) \rceil \neq 0$ 
(21)          $k \leftarrow k + 1$ 
(22)       else
(23)          $k \leftarrow k + 2$ 
(24)       end
(25)     else
(26)        $k \leftarrow k + 1$ 
(27)   end

```

ALGORITHM 2: The proposed loop reduction LLL algorithm using look-ahead check technique.

LLL reduction constraints are violated. Most important of all, redundant check operations occur very often when the index k decreases. Thus, the decrease of k is not always necessary because the size and LLL reductions have been checked in the last loop. We calculate the percentage of the redundant decreases of k and list them in the Table 1. We can see that the percentage of the redundant decrease of k achieves 67% for 2×2 lattice reduction and converge to 28% if the MIMO dimension is larger than 10×10 . Therefore, we propose a look-ahead check technique to modify index k and avoid unnecessary check operations in the original LLL algorithm.

4. Look-Ahead Check

The number of loops is often treated as a benchmark for computation complexity and latency in the literature on LRA MIMO detection [7]. In order to eliminate the redundant check operations in the LLL algorithm, we propose a look-ahead check technique by classifying the original loops to forward loop and back loop. And the loop reduction LLL algorithm is shown in Algorithm 2. The corresponding flow chart of the proposed algorithm and each loop is shown in Figure 1.

4.1. Back Loop. We define the back loop as the loop that only contains LLL reduction check and LLL violation processing as shown in Figure 1. We find that the size reduction constraint will not be violated after the k is decreased because the $\bar{\mathbf{R}}(k, k)$ has already been size reduced in the previous processing and is not changed in the column-swapping operation. And the givens rotation will only change the k and $k - 1$ row while the row value above $k - 2$ row remains size reduced. That means only LLL reduction check is required in the back loop. This LLL reduction check is named as *back-state* LLL reduction check to differentiate with the origin one. Because only the LLL reduction part needs to be executed in this back loop, we use a while loop in our algorithm to avoid the redundant size reduction operation. Nonetheless, there is still one case that the size reduction will process. If the division result exactly equals 0.5, the original LLL algorithm will do the size reduction operation in the back loop. But our algorithm will skip. This will produce a different lattice-reduced matrix at last. However, to do the size reduction or not to do in this case will both produce the matrix that obeys the LLL lattice reduction criterion. Although, we cannot prove the performance is the same mathematically, we will show that their performance is the same through the Monte-Carlo simulation in latter section. So the lattice reduction

FIGURE 1: The flow chart for loop-reduction LLL algorithm.

TABLE 2: Average loop of lattice-reduction Algorithms for different antenna numbers.

N	2	4	6	8	10	12
Original LLL algorithm	4.72	16.99	29.73	39.25	46.11	51.56
LR-LLL algorithm	4.6 (97.4%)	15.98 (94%)	27.8 (93.5%)	36.7 (93.5%)	43.53 (94.4%)	48.77 (94.5%)

performance will not suffer any degradation. Using the look-ahead check technique, we can more precisely determine the next k value at the end of each loop.

4.2. Forward Loop. Forward loop is just like the original loop defined in the previous section except once the LLL reduction is violated, it will enter the back loop. If *back-state* LLL reduction constraint is not violated, we enter the *stay-state* size reduction check to predict the next index k . Notice that if the *stay-state* size reduction constraint ($\mu \neq 0$) is not violated, the LLL reduction constraint must not be violated because the $\bar{\mathbf{R}}(k, k)$ has already been LLL reduced in the previous processing and all values remain unchanged. Therefore, we can also perform the *stay-state* size reduction check ahead. If the *stay-state* size-reduction constraint is not violated, we can simply increase the index k by 2 to skip a redundant LLL reduction and enter the forward loop.

5. Simulation Results

To verify the proposed LLL algorithm, we simulate the LLL-aided MIMO detections based on the MIMO system described in Section 2, and we employ sorted QR decomposition in all MIMO detectors. The LLL-reduction parameter δ equals 0.75, as suggested in [11]. Table 2 shows the average loop numbers of the original and the proposed LLL

algorithms for different antenna numbers. Forward loop and back loop are all counted as a loop in our algorithm. The proposed LLL algorithm can reduce the average number of loops to 93%~94% of the original LLL algorithm. The BER versus SNR curve is shown for 4×4 and 8×8 MIMO systems in Figures 2 and 3, respectively. The performance is exactly the same for our algorithm and the original LLL algorithm.

We also analyze the computational complexity and latency of our algorithm. The results for 4×4 and 8×8 MIMO systems are listed in Tables 3 and 4. The computation is divided to four operations such as addition, multiplication, division, and givens rotation. Our algorithm is lower in total computational complexity and especially in the division which tends to cost more time for computation. The lower ratio is just like the loop-reduced ration. But only average computational complexity cannot clearly show the advantage of our algorithm. Since the original LLL algorithm contains lots of redundant checks operation which are unable to process in parallel, it will result in long average processing time to complete the lattice reduction operation. We try to simulate the latency by parallelizing all the possible operations. The latency counts are as follows: the line (5) to line (9) in our algorithm is counted as one division, one addition and one multiplication. The LLL reduction check operation contains four multiplications and two additions. The column swap operation is counted as no operation delay. The givens rotation counts one at each back loop. And the *stay-state* size

TABLE 3: Average computational complexity and latency of lattice-reduction Algorithms for 4×4 MIMO detection.

Algorithm	Addition	Multiplication	Division	Givens Rotation	Total
LLL	204.6/49.4	238.5/83.4	64.45/64.45	67.34/10.43	575/208.4
LR-LLL	202.79/47.5	234.81/79.5	46.43 (72%)/46.43	67.83/10.43	551.88 (96%)/184 (88.2%)

TABLE 4: Average computational complexity and latency of lattice-reduction Algorithms for 8×8 MIMO detection.

Algorithm	Addition	Multiplication	Division	Givens Rotation	Total
LLL	1463.3/141.6	1541.8/220	339.6/339.6	285.8/24.34	3630/725.6
LR-LLL	1462.5/137.21	1527.4/211.1	218.12 (64%)/218.12	287.46/24.49	3504.5 (96%)/591 (81.4%)

FIGURE 2: BER versus SNR of the 4×4 QR-SIC MIMO detectors using different lattice-reduction algorithms. FIGURE 3: BER versus SNR of the 8×8 QR-SIC MIMO detectors using different lattice-reduction algorithms.

reduction is counted as a division operation. The latency is shown in the Table after the dashed line. The saving is about 22%~29% and grows with antenna number.

6. Hardware Architecture

6.1. Top Structure. In this paper, we proposed a very intuitive structure for our LR-LLL algorithm in Figure 4. The center controller counts the index k by the LLL violation results. And it will send control signals to choose the specific matrix element to the input of the combinational circuit. We can also call size reduction part as size reduction loop and LLL reduction part as LLL reduction loop, respectively. The update circuit for the remaining $\mathbf{R}$ matrix, $\mathbf{T}$ matrix, and $\mathbf{Q}$ matrix are omitted for simplification. In this architecture, CORDIC circuit has two pipelined stages. So it required one cycle for size reduction loop and four cycles for a LLL reduction loop. The traditional LLL algorithm always processes the forward loop which contains the execution of the whole circuit. While using LR-LLL algorithm, some

TABLE 5: Average Cycle of Lattice Reduction Algorithms for different antenna numbers.

N	4	8	12
Original LLL algorithm	109	451	869
LR-LLL algorithm	89 (81.6%)	320 (70.9%)	469 (53.9%)

forward loops will replace by back loops. The average cycle counts for the LLL algorithm, and our LR-LLL algorithm is listed in Table 5. We can find out that as the antenna number grows, the reduction of average cycle grows. And the FPGA results are shown in Table 6. In [5], the complex-valued LLL lattice reduction algorithm is proved to have lower computational complexity than real-valued system. This is mainly due to the double sized of the real number system comparing to complex number. So the hardware or cycle counts may be larger than the previous two complex number works.

6.2. Other Blocks. The divr block executes the divide and round operation which can be easily designed by long

FIGURE 4: Hardware architecture for LR-LLL algorithm.

TABLE 6: FPGA implementation results and comparison.

	Virtex-4 [8]	Vertex-4 [9]	Virtex-4 [10]	Virtex-4 [This work]
LRA algorithm	RS-LLL	CLLL (modified)	CT-LLL	LR-LLL
System model	Complex	Complex	Real	Real
Slices	4805	3640	11330	4170
Antenna number	4	4	4	4
Frequency	79 MHz	140 MHz	8.7 MHz	53 MHz
Ave. cycles/matrix	14	96	5	89
Throughput (M mat/s)	5.6	1.45	1.74	0.59

FIGURE 5: Division and round circuit.

division architecture. In Figure 5, we show a four-stage long-division architecture for five bits output divr circuit. The size reduction update circuit is composed of multiplication and addition circuit. Instead of calculating the square norm to do the LLL reduction comparison, we choose the CORDIC vector mode circuit to calculate the square root of the norm which may also be the output if the LLL check violates. The square root of δ is set to 0.875 to approximate the square root of 0.7. CORDIC rotation mode is used to do the Givens rotation of the algorithm. The output of the comparison circuit is the LLL reduction violation check results which will control the center controller and also enable the update circuits. The LLL reduction update block contains multiple CORDIC rotation circuits to do the givens

rotation of remaining row element of $\mathbf{R}$ and $\mathbf{Q}$ matrix. It also contains a swap circuit for $\mathbf{T}$ matrix.

7. Conclusion

In this paper, we propose a look-ahead check technique to eliminate unnecessary check operation in the LLL algorithm. The proposed algorithm not only reduces the average number of loops in the LLL algorithm but also reduces the computation complexity and latency of LLL algorithm. We also proposed a very intuitive architecture to estimate the clock cycle saving of our algorithm. The saving is dramatically increased while the antenna number grows. Therefore, we believe that the proposed loop reduction LLL algorithm benefits the lattice-reduction-aided MIMO detection.

References

- [1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.
- [2] M. Shabany and P. G. Gulak, "The application of lattice-reduction to the K-Best algorithm for near-optimal MIMO detection," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '08)*, pp. 316–319, May 2008.

- [3] H. Yao and G. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 1, pp. 424–428, November 2002.
- [4] W. Zhang, X. Ma, B. Gestner, and D. V. Anderson, "Designing low-complexity equalizers for wireless systems," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 56–62, 2009.
- [5] Y. H. Gan, C. Ling, and W. H. Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2701–2710, 2009.
- [6] P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, "VLSI implementation of a high-speed iterative sorted MMSE QR decomposition," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 1421–1424, May 2007.
- [7] M. Sandell, A. Lillie, D. McNamara, V. Ponnampalam, and D. Milford, "Complexity study of lattice reduction for MIMO detection," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 1088–1092, March 2007.
- [8] L. Bruderer, C. Studer, M. Wenk, D. Seethaler, and A. Burg, "VLSI implementation of a low-complexity LLL lattice reduction algorithm for MIMO detection," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '10)*, pp. 3745–3748, May 2010.
- [9] B. Gestner, Z. Wei, X. Ma, and D. V. Anderson, "Lattice reduction for MIMO detection: from theoretical analysis to hardware realization," *IEEE Transactions on Circuits and Systems—Part I: Regular Papers*, vol. 58, no. 4, pp. 813–826, 2011.
- [10] C. F. Liao and Y. H. Huang, "Power-saving 4x4 lattice-reduction processor for MIMO detection with redundancy checking," *IEEE Transactions on Circuits and Systems—Part II: Express Briefs*, vol. 58, no. 2, pp. 95–99, 2011.
- [11] A. K. Lenstra, H. W. Lenstra, and L. Lovasz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.
- [12] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "MMSE-based lattice-reduction for near-ML detection of MIMO systems," in *Proceedings of the ITG Workshop on Smart Antennas*, pp. 106–113, May 2004.