A COMPARISON OF COMPLEX LATTICE REDUCTION ALGORITHMS FOR MIMO DETECTION

Luis G. Barbero, T. Ratnarajah and Colin Cowan

ECIT, Queen's University of Belfast
Queen's Island, Belfast, BT3 9DT, UK
e-mail: {l.barbero, t.ratnarajah, c.cowan}@ecit.qub.ac.uk

ABSTRACT

The performance and complexity of two complex lattice reduction (LR) algorithms used in multiple input-multiple output (MIMO) detection are compared in this paper. The Seysen's Algorithm (SA) has been previously proposed as a low-complexity alternative to the real version of the Lenstra-Lenstra-Lovász (LLL) algorithm while providing a better performance in LR-aided linear detectors. However, this paper shows that the SA has a higher complexity than the complex version of the LLL algorithm, due to its more computationally intensive preprocessing stage and its higher complexity per iteration. In addition, both the SA and the complex LLL algorithm provide practically the same performance when used in LR-aided successive interference cancellation (SIC) detectors.

Index Terms— lattice reduction, MIMO, LLL algorithm, Seysen's algorithm.

1. INTRODUCTION

In the context of uncoded detection of spatially multiplexed multiple input-multiple output (MIMO) systems, low-complexity linear and SIC detectors fail to achieve the diversity of the optimal maximum likelihood detector (MLD), resulting in a sub-optimal performance [1]. In order to overcome that problem, lattice reduction (LR)-aided detection has been recently proposed, where LR techniques are used to transform the MIMO channel into a *more orthogonal* equivalent MIMO channel [2]. Linear or SIC detectors can then be applied to this equivalent channel providing an improved performance [3]. In particular, it has been shown that LR-aided detectors achieve the same diversity as the MLD [4].

A number of LR methods exist in the literature with different levels of performance and complexity [5]. The optimal Korkine-Zolotareff (KZ) algorithm has the drawback of an exponential complexity, limiting its practical application [5]. As an alternative, the more popular Lenstra-Lenstra-Lovász (LLL) algorithm can approximate the performance of the KZ algorithm while having a polynomial complexity [6]. For that reason, the LLL algorithm has been considered, almost exclusively, for real and complex LR-aided detectors [7] -[9]. Recently, the Seysen's Algorithm (SA), originally proposed in [10], [11], has been presented as an alternative to the LLL algorithm for MIMO detection [12]. It results in a LR-aided linear detector with better performance than the LLL counterpart and requires less iterations per LR than the real LLL algorithm [12]. However, this paper shows that the SA has a higher complexity than the LLL algorithm, especially if the complex LLL is considered [9]. This is due to the overhead required in the SA and the higher complexity per iteration compared to the LLL algorithm. In addition, this paper shows that the improved performance of SA-aided linear detectors compared to LLL-aided ones disappears if successive interference cancellation (SIC) detectors are used.

This work was supported by the UK Engineering and Physical Sciences Research Council under grant number EP/C004132/1.

1.1. Lattice Reduction-Aided Detection

We consider a spatially-multiplexed MIMO system with M transmit and N receive antennas, denoted as $M \times N$. The vector of received symbols $\mathbf{y} \in \mathbb{C}^{N\times 1}$ can be modelled as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v}\,, \tag{1}$$

where $\mathbf{s} \in \mathbb{C}^{M\times 1}$ denotes the vector of transmitted symbols taken independently from a quadrature amplitude modulation (QAM) constellation $\mathcal{O}$ of P points with $\mathrm{E}[|s_i|^2] = 1/M$, for $1 \le i \le M$, and where $\mathbf{v} \in \mathbb{C}^{N\times 1}$ is the vector of independent complex Gaussian noise samples $v_i \sim \mathcal{CN}(0, \sigma^2)$, for $1 \le i \le M$. The channel matrix $\mathbf{H} \in \mathbb{C}^{N\times M}$ has independent elements $h_{j,i} \sim \mathcal{CN}(0, 1)$, for $1 \le j \le N$ and $1 \le i \le M$, representing a wireless propagation environment with uncorrelated Rayleigh fading. We assume that the channel is perfectly known at the receiver and that $N \ge M$.

The columns of the channel matrix $\mathbf{H}$ in (1), $\mathbf{h}_i$ for $1 \le i \le M$, can be seen as a generator basis of an M-dimensional complex lattice $\mathcal{L}(\mathbf{H}) \in \mathbb{C}^{N\times 1}$, where the lattice is defined as all complex integer combinations of the generator basis, i.e.

$$\mathcal{L}(\mathbf{H}) \triangleq \left\{ \mathbf{H}\mathbf{z} = \sum_{i=1}^{M} \mathbf{h}_i z_i \mid z_i \in \mathbb{CZ} \text{ for } 1 \le i \le M \right\}.$$

We concentrate our analysis on the complex lattice interpretation of the system, as opposed to the more common real one, since the SA is applied directly to the complex lattice and the complex LLL algorithm results in a lower complexity compared to the real LLL algorithm [8].

The main idea behind LR-aided detectors is to obtain a reduced (i.e. *more orthogonal*) generator basis $\tilde{\mathbf{H}}$ for the same lattice $\mathcal{L}$ in order to improve the performance of sub-optimal detectors [2]. Two matrices $\mathbf{H}$ and $\tilde{\mathbf{H}}$ generate the same lattice, $\mathcal{L}(\mathbf{H}) = \mathcal{L}(\tilde{\mathbf{H}})$, if they can be written as $\tilde{\mathbf{H}} = \mathbf{H}\mathbf{T}$, where $\mathbf{T} \in \mathbb{CZ}^{M\times M}$ is a unimodular matrix with determinant $\det(\mathbf{T}) = \pm 1$ [5]. The system model in (1) can then be rewritten as

$$\mathbf{y} = \tilde{\mathbf{H}}\mathbf{x} + \mathbf{v}\,, \tag{2}$$

where $\mathbf{x} = \mathbf{T}^{-1}\mathbf{s}$. Thus, sub-optimal detectors can be applied initially to (2) in order to obtain an estimate of $\mathbf{x}$, $\hat{\mathbf{x}}$, before calculating an estimate of the transmitted vector $\mathbf{s}$, $\hat{\mathbf{s}}$, using the relationship $\mathbf{s} = \mathbf{T}\mathbf{x}$. A detailed description of the operation of LR-aided detectors can be found in [7].

2. COMPLEX LATTICE REDUCTION ALGORITHMS

This Section briefly describes the main aspects of the LLL algorithm and the SA applied directly to the complex system defined in (1).

2.1. LLL Algorithm

The LLL algorithm transforms an input basis $\mathbf{H} = \mathbf{QR}$ into a LLL-reduced basis $\tilde{\mathbf{H}} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$ that satisfies

$$|\Re(\tilde{r}_{l,k})|\,,\ |\Im(\tilde{r}_{l,k})| \leq \frac{1}{2}|\tilde{r}_{l,l}| \quad \text{for} \quad 1 \leq l < k \leq M \qquad (3)$$

and

$$\delta|\tilde{r}_{k-1,k-1}|^2 \leq |\tilde{r}_{k,k}|^2 + |\tilde{r}_{k-1,k}|^2 \quad \text{for} \quad 2 \leq k \leq M\,, \qquad (4)$$

where $\tilde{r}_{l,k}$, for $1 \leq l, k \leq M$, are the elements of $\tilde{\mathbf{R}}$ and the parameter δ, assumed to be $\delta = 3/4$ as in [6], determines the speed of the algorithm and the *quality* of the reduced basis.

With the initialization $\tilde{\mathbf{Q}} = \mathbf{Q}$, $\tilde{\mathbf{R}} = \mathbf{R}$ and $\mathbf{T} = \mathbf{I}$, the algorithm performs a series of iterations starting from $k = 2$ until $k = M$ and (3) and (4) are satisfied. The output of the algorithm is then given by the updated $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{R}}$ and $\mathbf{T}$. The following two steps are performed in each iteration:

1. A number of basis reduction operations, as detailed in [9], are performed for $1 \leq l < k$ if (3) is not satisfied.
2. Columns $\tilde{\mathbf{r}}_k$ and $\tilde{\mathbf{r}}_{k-1}$ are swapped (with Givens rotations), as detailed in [9], if (4) is not satisfied and $k \leftarrow \max(k-1, 2)$. If (4) is satisfied, no swapping occurs and $k \leftarrow k+1$. In both cases, the algorithm returns to step 1.

Given that the complexity of the LLL algorithm depends greatly on the number of column exchanges, that can be reduced by using a sorted version of the QR decomposition (SQR) that iteratively minimizes the diagonal elements of $\mathbf{R}$ [7]. This version of the LLL algorithm is denoted as sorted-LLL (SLLL) in this paper.

2.2. Seysen's Algorithm

The SA consists of obtaining a SA-reduced basis $\tilde{\mathbf{H}}$ by simultaneously reducing the original basis $\mathbf{H}$ and a basis $\mathbf{H}^{\mathrm{d}}$ of the dual lattice $\mathcal{L}^{\mathrm{d}}$, where $\mathbf{H}^{\mathrm{d}} = \mathbf{H}(\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}$ [11]. The reduced basis $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{H}}^{\mathrm{d}}$ satisfy [1]

$$\lambda_{i,j} \triangleq \lfloor \alpha_{i,j} \rceil = \left\lfloor \frac{1}{2}\left(\frac{\tilde{g}^{\mathrm{d}}_{j,i}}{\tilde{g}^{\mathrm{d}}_{i,i}} - \frac{\tilde{g}_{j,i}}{\tilde{g}_{j,j}} \right) \right\rceil = 0 \text{ for } 1 \leq i \neq j \leq M, \quad (5)$$

where $\lfloor \cdot \rceil$ denotes rounding to the next integer and $\tilde{g}_{i,j}$ and $\tilde{g}^{\mathrm{d}}_{i,j}$, for $1 \leq i, j \leq M$, are the elements of $\tilde{\mathbf{G}} = \tilde{\mathbf{H}}^{\mathrm{H}}\tilde{\mathbf{H}}$ and $\tilde{\mathbf{G}}^{\mathrm{d}} = \tilde{\mathbf{H}}^{\mathrm{dH}}\tilde{\mathbf{H}}^{\mathrm{d}}$, respectively. When (5) is satisfied, the SA has found a local minimum of the Seysen's orthogonality measure, defined in (2.2) in [11].

With the initialization $\tilde{\mathbf{H}} = \mathbf{H}$, $\tilde{\mathbf{H}}^{\mathrm{d}} = \mathbf{H}^{\mathrm{d}}$ and $\mathbf{T} = \mathbf{I}$, the SA performs a series of iterations until (5) is satisfied. In each iteration, taking into account that $1 \leq i \neq j \leq M$, the following steps are performed:

1. Initially, the values $\lambda_{i,j}$ are calculated. The SA terminates if $\lambda_{i,j} = 0$ for all i, j, giving the updated $\tilde{\mathbf{H}}$, $\tilde{\mathbf{H}}^{\mathrm{d}}$ and $\mathbf{T}$ as output.
2. An index pair (k, l) is selected according to [12]

$$(k, l) = \arg\max_{(i,j)} \ 2\,\tilde{g}_{j,j}\,\tilde{g}^{\mathrm{d}}_{i,i}\,(2\Re(\lambda^*_{i,j}\alpha_{i,j}) - |\lambda_{i,j}|^2). \quad (6)$$

3. The k-th columns of $\tilde{\mathbf{H}}$ and $\mathbf{T}$ and the l-th column of $\tilde{\mathbf{H}}^{\mathrm{d}}$ are updated using

$$\tilde{\mathbf{h}}_k = \tilde{\mathbf{h}}_k + \lambda_{k,l}\tilde{\mathbf{h}}_l\,;\ \mathbf{t}_k = \mathbf{t}_k + \lambda_{k,l}\mathbf{t}_l\,;\ \tilde{\mathbf{h}}^{\mathrm{d}}_l = \tilde{\mathbf{h}}^{\mathrm{d}}_l - \lambda^*_{k,l}\tilde{\mathbf{h}}^{\mathrm{d}}_k\,.$$

4. According to step 2, the corresponding values of $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{G}}^{\mathrm{d}}$ are updated as detailed in [12]. The algorithm returns to step 1 where only the values $\lambda_{i,j}$ corresponding to updated elements of $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{G}}^{\mathrm{d}}$ need to be calculated.

The SA has been shown to provide an improved performance compared to the LLL algorithm when applied to LR-aided linear detectors [12]. Although it has been claimed to have lower complexity than the LLL algorithm, only the number of basis updates in the SA (steps 3 and 4 above) and the number of column exchanges in the LLL algorithm have been compared [12]. Instead, the next two Sections look at the total number of operations of both algorithms, indicating that the SA has a higher complexity than the LLL algorithm.

3. COMPUTATIONAL COMPLEXITY

This Section looks at the *computational* complexity in number of real operations of the two LR algorithms in order to obtain some insight into their relative complexity [2]. The complexity of both algorithms can be divided between the complexity of the preprocessing stage (i.e. input generation) and the complexity of the algorithm stage. Although both outputs are different, it can be assumed, for simplicity, that the use of $\tilde{\mathbf{Q}}/\tilde{\mathbf{R}}$ or $\tilde{\mathbf{H}}^{\mathrm{d}}$ results in the same complexity for the MIMO detectors.

3.1. LLL Algorithm

Firstly, the preprocessing stage performs a QR decomposition of $\mathbf{H}$, or a SQR decomposition if the SLLL is used [7]. Considering the modified Gram-Schmidt (MGS) algorithm [13], the number of operations required by the QR and the SQR decomposition are $N_{\mathrm{QR}} = 8NM^2 - 2NM - M^2 + M$ and $N_{\mathrm{SQR}} = 10NM^2 - 4NM - 1.5M^2 + 1.5M$, respectively. The SQR decomposition results in a slight increase in complexity that can be reduced, at the expense of a minor performance degradation, if a different sorting is used [8]. Secondly, the complexity of each iteration of the algorithm stage can be divided between the complexity of the two steps defined in Section 2.1:

1. The number of operations required for the basis reductions of step 1 depends on the number of times that (3) is not satisfied [9]. However, an upper bound, not tight, can be found if we consider $k = M$ and (3) not satisfied for all $1 \leq l < M$. Thus, the number of operations of step 1 can be upper bounded by $N_{\mathrm{LLL}-1} \leq 18M - 10$.
2. The number of operations required for the column exchange of step 2 depends on the index k [9]. Again, an upper bound can be found considering $k = 2$, yielding $N_{\mathrm{LLL}-2} \leq 28(N + M) + 16$.

[1] $\tilde{\mathbf{H}}$ is used to denote both an SA-reduced and an LLL-reduced basis to simplify the notation.

[2] We consider that a complex product requires 4 real products and 2 real additions; a complex addition requires 2 real additions; a complex by complex division requires 6 real products, 3 real additions and 2 real divisions; and a complex by real division requires 2 real divisions.

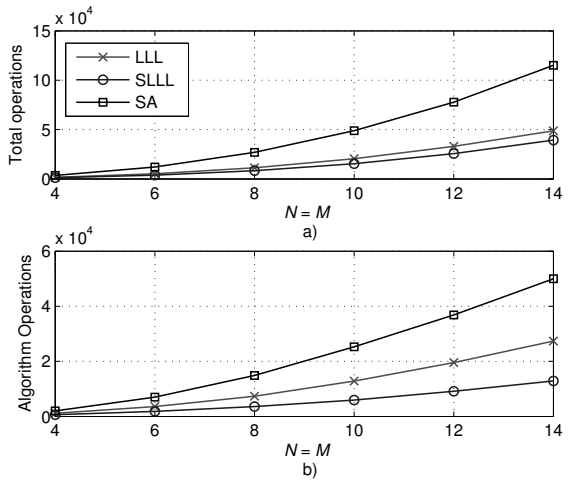

Fig. 1. a) Average total number of operations and b) average number of operations of the algorithm stage of the LLL, SLLL and SA as a function of the number of antennas $N = M$.

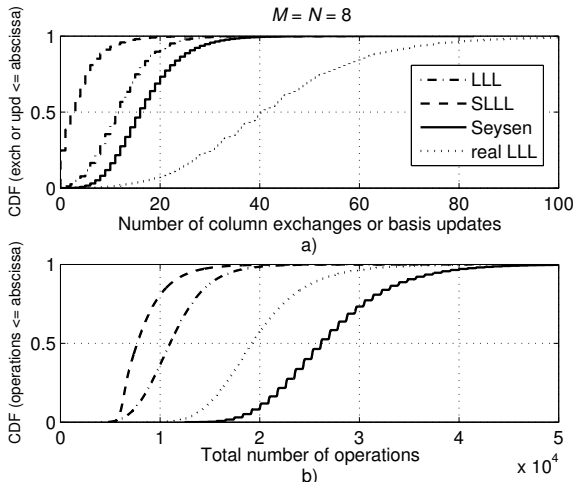

Fig. 2. a) CDFs of the number of column exchanges in LLL-based algorithms and the number of basis updates in the SA and b) CDFs of the total number of operations of the LLL, SLLL, SA and real LLL in an 8×8 system.

Looking at the two upper bounds, it can be seen that the complexity of the algorithm stage is linear with M as indicated in [12]. However, the overall complexity depends on the number of iterations performed, which cannot be easily characterized given that k can be either incremented or decremented in step 2. Thus, we resort to simulations to evaluate the overall complexity in Section 4.

3.2. Seysen's Algorithm

Firstly, the preprocessing stage in this case needs to calculate the basis $\mathbf{H}^{\mathrm{d}}$ and the initial values of $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{G}}^{\mathrm{d}}$, therefore, two matrix multiplications and a matrix inversion are required. Assuming a matrix inversion through Gaussian elimination [13], the number of operations of the preprocessing stage is given by $N_{\mathrm{SA}-0} = 16NM^2 + 8M^3 - 2NM - M^2 - 10M + 6$. Thus, the SA has a higher preprocessing complexity than the LLL algorithm. Secondly, the complexity of each iteration of the algorithm stage can be divided between the complexity of the four steps defined in Section 2.2:

1. The number of operations required in step 1 is $N_{\mathrm{SA}-1} = 8(M^2 - M)$ in the first iteration and down to $N_{\mathrm{SA}-1} = 32(M-1)$ in the following iterations, given that only the $\lambda_{i,j}$ corresponding to updated elements of $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{G}}^{\mathrm{d}}$ need to be calculated [12].
2. Equivalently, and without considering the $\max$ search, the number of operations required in step 2 is $N_{\mathrm{SA}-2} \leq 10(M^2 - M)$ in the first iteration and down to $N_{\mathrm{SA}-2} \leq 40(M-1)$ in the following iterations, since only the terms in (6) where $\lambda_{i,j} \neq 0$ need to be calculated [12].
3. The number of operations required in step 3 is $N_{\mathrm{SA}-3} = 16N + 8M$.
4. Finally, the number of operations required in step 4 is $N_{\mathrm{SA}-4} = 24M - 18$ [12].

The complexity per iteration is also linear with M, except in the first iteration. However, looking at one *full* iteration, the number of operations in the LLL algorithm is $N_{\mathrm{LLL-it}} \leq 28N + 46M + 6$ while the number of operations in the SA is, at best, $16N + 64M - 50 \leq N_{\mathrm{SA-it}} \leq 16N + 104M - 90$. Thus, unless $N \gg M$, only in the case where the number of column exchanges in the LLL is considerably larger than the number of basis updates in the SA, can the SA have a lower algorithm complexity than the LLL algorithm. Next Section shows that this is not the case in practice, limiting, in principle, the relevance of the SA as a low-complexity alternative to the LLL algorithm.

4. SIMULATION RESULTS

The complexity and performance of both LR algorithms is studied in this Section through Monte-Carlo simulations. Fig. 1-a) shows the average overall number of real operations of the algorithms as a function of the number of antennas $M = N$. Furthermore, given that the higher complexity of the preprocessing stage for the SA can mask the relative complexity of the algorithm stages, Fig. 1-b) shows the average complexity of the algorithm stage. It can be seen how the LLL and SLLL algorithms have a significantly lower average complexity as the number of antennas increase. Even though the SLLL algorithm has an increased preprocessing complexity compared to the LLL algorithm, that effect is compensated for by the reduction in complexity in the SLLL algorithm stage.

Fig. 2-a) shows the cumulative distribution functions (CDFs) of the number of column exchanges in the LLL-based algorithms and the number of basis updates in the SA in an 8×8 system. As in [12], it can be observed that the SA has a lower number of basis updates compared to the number of column exchanges of the real LLL algorithm. However, that is not the case when the complex LLL or SLLL are considered. Fig. 2-b) shows the CDFs of the total complexity of the algorithms. The only qualitative difference compared to Fig. 2-a) is the fact that the real LLL algorithm also has a lower complexity than the SA Thus, the complex SLLL algorithm presents the lowest complexity of the algorithms under study while the SA has the highest complexity. It should be noted that this conclusion is based solely on the evaluation of the number of real operations. Given the algorithmic differences between the LLL algorithm and the SA, a hardware implementation would be required to establish their exact relative complexities, since both LR algorithms are likely to result in

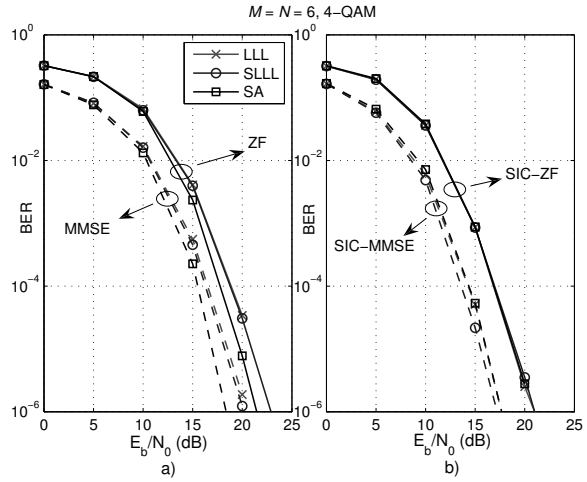

Fig. 3. BER performance of LR-aided a) linear detectors and b) SIC detectors as a function of the SNR per bit in a 6×6 system with 4-QAM.

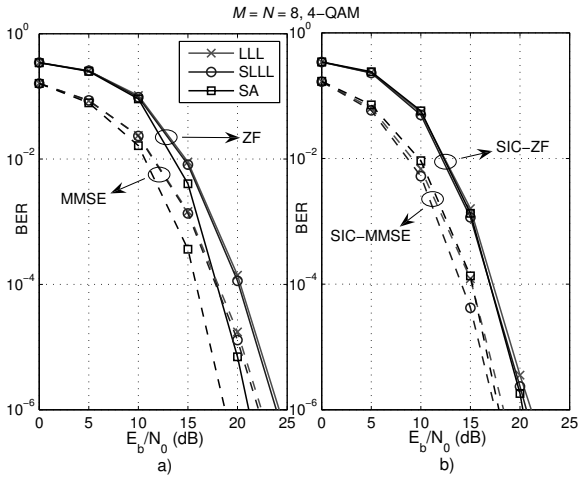

Fig. 4. BER performance of LR-aided a) linear detectors and b) SIC detectors as a function of the SNR per bit in a 8×8 system with 4-QAM.

very different hardware architectures.

Fig. 3 and Fig. 4 show the bit error rate (BER) performance of LR-aided detectors in a 6×6 and an 8×8 system, respectively, with 4-QAM as a function of the SNR per bit $E_b/N_0 = \log_2^{-1}(P)/\sigma^2$. It can be seen how the SA provides an improved performance when zero forcing (ZF) or minimum mean square error (MMSE) detectors are used, i.e. Fig. 3-a) and Fig. 4-a). This effect, more noticeable as the number of antennas increases, is due to the *better quality* of the reduced basis provided by the SA compared to the LLL algorithms [12]. When SIC detectors are used in Fig. 3-b) and Fig. 4-b), the three methods provide practically the same BER performance, similar to what has been previously observed between KZ-aided and LLL-aided SIC detectors [14]. The SLLL-aided SIC detector gives a slightly better performance in the MMSE case, due to the column sorting within the SQR decomposition.

5. CONCLUSION AND FUTURE WORK

This paper has compared the performance and complexity of the SA and the complex LLL algorithm when applied to MIMO detection. Although it has been previously shown that the SA provides a *more orthogonal* lattice basis compared to the LLL algorithm, it does so at the expense of a higher complexity, especially compared to the SLLL. When looking at the performance of both LR-aided detectors, the *more orthogonal* basis of the SA results in an improved performance compared to the LLL algorithm if linear detectors are used. However, both algorithms yield practically the same performance if SIC detectors are used.

Further work includes implementing both LR algorithms in hardware and assessing whether LR algorithms might not be able to improve the performance of SIC detectors beyond a certain limit, independently of the specific LR algorithm.

6. ACKNOWLEDGEMENT

The authors would like to thank Dominik Seethaler for many insightful discussions.

7. REFERENCES

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, New York, NY, USA, 2005.

[2] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE GLOBECOM '02*, Taipei, Taiwan, Nov. 2002.

[3] C. Windpassinger and R. F. H. Fischer, "Low-complexity near-maximum-likelihood detection and precoding for MIMO systems using lattice reduction," in *Proc. IEEE ITW '03*, Paris, France, Apr. 2003.

[4] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "LLL lattice-basis reduction achieves the maximum diversity in MIMO systems," in *Proc. IEEE ISIT '05*, Adelaide, Australia, Sept. 2005.

[5] 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.

[6] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, Dec. 1982.

[7] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE-based lattice-reduction for near-ML detection of MIMO systems," in *Proc. ITG WSA '04*, Munich, Germany, Mar. 2004.

[8] Y. H. Gan and W. H. Mow, "Complex lattice reduction algorithms for low-complexity MIMO detection," in *Proc. IEEE GLOBECOM '05*, St. Louis, MO, USA, Nov. 2005.

[9] M. Sandell, A. Lillie, D. McNamara, V. Ponnampalam, and D. Milford, "Complexity study of lattice reduction for MIMO detection," in *Proc. IEEE WCNC '07*, Las Vegas, NV, USA, Mar. 2007.

[10] M. Seysen, "Simultaneous reduction of a lattice basis and its reciprocal basis," *Combinatorica*, vol. 13, no. 3, pp. 363–376, Sept. 1993.

[11] B. A. LaMacchia, *Basis Reduction Algorithms and Subset Sum Problems*, MIT MSc Thesis, May 1991.

[12] D. Seethaler, G. Matz, and F. Hlawatsch, "Low-complexity MIMO data detection using Seysen's lattice reduction algorithm," in *Proc. IEEE ICASSP '07*, Honolulu, HI, USA, Apr. 2007.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press Ltd., London, UK, 1996.

[14] C. Windpassinger and R. F. H. Fischer, "Optimum and sub-optimum lattice-reduction-aided detection and precoding for MIMO communications," in *Proc. CWIT '03*, Waterloo, ON, Canada, May 2003.