ON DECREASING THE COMPLEXITY OF LATTICE-REDUCTION-AIDED K-BEST MIMO DETECTORS.

Sandra Roger, Alberto Gonzalez, Vicenc Almenar, and Antonio M. Vidal

Institute of Telecommunications and Multimedia Applications (iTEAM)
Universidad Politecnica de Valencia
Camino de Vera s/n, 46022 Valencia, Spain
email: sanrova@iteam.upv.es

ABSTRACT

The application of Lattice Reduction techniques over the MIMO channel matrix is known to improve the performance of MIMO detectors. Several authors have proposed Lattice-Reduction-Aided K-Best detectors for improving the performance of conventional K-Best algorithms. In this paper, efficient ways of decreasing the computational complexity of previously proposed schemes are presented. The knowledge about how the Lattice Reduction stage affects the transmitted symbols is exploited in order to significantly decrease the complexity without performance loss.

1. INTRODUCTION

In the last years, Lattice Reduction (LR) techniques have been proposed to transform the MIMO system model into an equivalent one with a better-conditioned channel matrix [1]. These techniques have been shown to improve the detection performance when used previously to linear detectors [2][3]. Recently, some other authors have also proposed LR algorithms for working with K-Best algorithms [4], building Lattice-Reduction-Aided (LRA) K-Best detectors. The K-Best algorithm is a suboptimal MIMO detector based on tree search decoding [5]. LRA K-Best algorithms require an initial estimate of the solution at each level of the tree, generally obtained via Successive Interference Cancellation (SIC) schemes. In order to obtain the *K* best candidates at each level of the detection, these methods perform a search of *N* constellation points around the solution given by the SIC detector. In this work, we propose a low complexity preprocessing stage that allows limiting this search around the SIC solution without decreasing the detection performance. In addition, a dynamic distribution of both the *K* and *N* values is proposed in order to further decrease the number of explored candidates. Results show that these strategies can noticeably reduce the computational cost.

2. SYSTEM MODEL.

Let us consider a block fading MIMO system with n_T transmit antennas, n_R receive antennas ($n_R \geq n_T$) and a signal to noise ratio denoted by ρ. The baseband equivalent model for such MIMO system is given by

$$\mathbf{x_c} = \mathbf{H_c}\mathbf{s_c} + \mathbf{v_c}, \tag{1}$$

where $\mathbf{s_c}$ represents the baseband signal vector transmitted during each symbol period, which is composed by elements chosen from the same constellation $\mathbf{\Omega_C}$, such as QAM. Vector $\mathbf{x_c}$ in (1) denotes the received symbol vector and $\mathbf{v_c}$ is a complex white Gaussian noise vector. The Rayleigh fading channel matrix $\mathbf{H_c}$ is considered known at the receiver. This matrix is formed by $n_R \times n_T$ complex-valued elements, H_{ij}, which represent the complex fading gain from the *j*-th transmit antenna to the *i*-th receive antenna. For practical reasons, it is convenient to transform the $(n_R \times n_T)$-dimensional complex equation (1) into an equivalent $(2n_R \times 2n_T)$-dimensional real-valued representation (2) [5]:

$$\begin{bmatrix} \Re(\mathbf{x_c}) \\ \Im(\mathbf{x_c}) \end{bmatrix} = \begin{bmatrix} \Re(\mathbf{H_c}) & -\Im(\mathbf{H_c}) \\ \Im(\mathbf{H_c}) & \Re(\mathbf{H_c}) \end{bmatrix} \begin{bmatrix} \Re(\mathbf{s_c}) \\ \Im(\mathbf{s_c}) \end{bmatrix} + \begin{bmatrix} \Re(\mathbf{v_c}) \\ \Im(\mathbf{v_c}) \end{bmatrix} \tag{2}$$

The real-valued description of the system (2) will be considered throughout this work, denoted by $\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{v}$. Hence, components of $\mathbf{s}$ will belong to the real constellation $\mathbf{\Omega}$. For instance, a QAM constellation of 16 possible symbols (16-QAM) can be represented by the real constellation $\mathbf{\Omega} = \{-3, -1, +1, +3\}$.

3. LATTICE-REDUCTION ALGORITHMS.

If the columns of the channel matrix $\mathbf{H}$ are considered the bases of a lattice, LR strategies such as the LLL algorithm [6] can be used to transform the channel matrix $\mathbf{H}$ into a new channel matrix $\widetilde{\mathbf{H}} = \mathbf{H}\mathbf{T}$ with less correlated columns [2], where $\mathbf{T}$ is a unimodular matrix ($\det(\mathbf{T}) = \pm 1$) with integer entries and called the transformation matrix. Considering the transformation matrix $\mathbf{T}$ into the system model (2), the received signal vector $\mathbf{x}$ can be rewritten as

$$\mathbf{x} = \mathbf{H}\mathbf{T}\mathbf{T}^{-1}\mathbf{s} + \mathbf{v} = \widetilde{\mathbf{H}}\mathbf{z} + \mathbf{v}, \tag{3}$$

where the symbols to be detected are now $\mathbf{z} = \mathbf{T}^{-1}\mathbf{s}$. Moreover, the application of Lattice Reduction techniques before the detection requires the constellation symbols being a set of continuous integers, which will be called $\widetilde{\mathbf{\Omega}}$. Therefore, it is necessary to introduce a $2n_T \times 1$ displacement vector $\mathbf{d} = [1, \ldots, 1]^T$ to shift and scale the original constellation symbols in the following way

$$\widetilde{\mathbf{x}} = \frac{(\mathbf{x} + \mathbf{H}\mathbf{d})}{2} = \mathbf{H}\left[\frac{(\mathbf{s} + \mathbf{d})}{2}\right] + \frac{\mathbf{v}}{2} = \widetilde{\mathbf{H}}\widetilde{\mathbf{z}} + \frac{\mathbf{v}}{2}, \tag{4}$$

where $\widetilde{\mathbf{s}} = (\mathbf{s} + \mathbf{d})/2$ and $\widetilde{\mathbf{z}} = \mathbf{T}^{-1}\widetilde{\mathbf{s}}$. After this transformation, the signal $\widetilde{\mathbf{x}}$ is used for carrying out the detection and the detected vector is then transformed back to the original format $\mathbf{s} = 2\mathbf{T}\widetilde{\mathbf{z}} - \mathbf{d}$.

4. DETECTION IN LATTICE REDUCED MIMO SYSTEMS

Given the received signal $\mathbf{x}$ and before applying any LR technique, the detection problem consists in determining the transmitted vector $\hat{\mathbf{s}}$ with the highest a posteriori probability. This is typically carried out in practice by solving the following least squares problem

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \Omega^{2n_T}} \|\mathbf{x} - \mathbf{H}\mathbf{s}\|^2. \quad (5)$$

The Maximum-Likelihood (ML) algorithm performs an exhaustive search over the total $2n_T$-dimensional lattice points $\mathbf{s}$. However, such an implementation is cumbersome for real systems and in practice the detection is performed via alternative detectors, such as tree search methods [7]. These methods obtain the solution of (5) by performing a search for each of the components of vector $\hat{\mathbf{s}}$, taking into account that they belong to a finite and a priori known alphabet.

However, when a previous LR has been performed to the channel matrix, the unknown vector to be detected becomes $\mathbf{z} = \mathbf{T}^{-1}\mathbf{s}$ instead of $\mathbf{s}$. Unfortunately, the set of $\mathbf{z}$ values is not predetermined in advance, since it not only depends on the constellation used but also on the matrix $\mathbf{T}^{-1}$ (i.e. it depends on the current channel realization). A straightforward way of determining all the possible $\mathbf{z}$ values could be calculating all the possible $\mathbf{s} \in \Omega^{2n_T}$ and afterwards transforming them with $\mathbf{T}^{-1}$, in order to have the points of the transformed lattice available. For instance, the set of possible values for symbols belonging to a QPSK constellation expressed in its real form is known to be $\mathbf{\Omega} = \{-1, 1\}$. Once the shift operation is performed over it, it results in $\widetilde{\mathbf{\Omega}} = \{0, 1\}$. Considering the example for the 2×2 transformation matrix used in [3], that is $\mathbf{T} = [1, -1; 0, 1]$ and its inverse matrix is $\mathbf{T}^{-1} = [1, 1; 0, 1]$, an exhaustive calculation of the possible vectors in the transformed lattice gives the correspondence shown in Table 1.

Table 1: Correspondence between $\widetilde{\mathbf{s}}$ vectors and $\widetilde{\mathbf{z}}$ vectors for a QPSK constellation and a value of $\mathbf{T} = [1, -1; 0, 1]$.

Possible $\widetilde{\mathbf{s}}$	Associated $\widetilde{\mathbf{z}}$
$[0,0]^T$	$[0,0]^T$
$[0,1]^T$	$[1,1]^T$
$[1,0]^T$	$[2,1]^T$
$[1,1]^T$	$[2,0]^T$

According to the values shown in Table 1, it can be noted that the elements of $\widetilde{\mathbf{z}}$ do not belong to $\widetilde{\mathbf{\Omega}}$ in all cases. Moreover, each of the components of vector $\mathbf{z}$ may expand a different range of possible values, which will depend on the $\mathbf{T}^{-1}$ matrix. It is known that conventional tree search detectors would need this set of transformed candidates for performing the detection. Unfortunately, this calculation requires similar complexity than ML-detection, so it is not feasible in practice. In order to avoid such a complex scheme, authors in [4] proposed a detection algorithm that calculates a first estimate of the solution of (5), generally via SIC detectors, and then the candidate solutions that lie on the neighborhood of this point are explored. For each component of the detected signal vector, only the K best candidates are stored and considered for the rest of components to detect. For this reason, this group of detectors are commonly known as K-Best detectors.

The steps proposed in [4] to carry out the detection are:

1) Perform a QR factorization of the lattice-reduced channel matrix ($\widetilde{\mathbf{H}} = \widetilde{\mathbf{Q}}\widetilde{\mathbf{R}}$) and multiply (4) by $\widetilde{\mathbf{Q}}^T$. The system becomes:

$$\widetilde{\mathbf{x}}' = \widetilde{\mathbf{R}}\widetilde{\mathbf{z}} + \frac{\widetilde{\mathbf{Q}}^T\mathbf{v}}{2}. \quad (6)$$

2) Calculate symbol estimate at layer $l = 2n_T$ using a SIC procedure:

$$\hat{\mathrm{z}}_{2n_T} = \left\lceil \frac{\widetilde{\mathrm{x}}'_{2n_T}}{\widetilde{\mathrm{R}}_{2n_T,2n_T}} \right\rfloor, \quad (7)$$

where $\lceil \cdot \rfloor$ rounds to the nearest integer.

3) Pick $N > K$ integer values around $\hat{\mathrm{z}}_{2n_T}$ at the $2n_T$ layer and calculate their Euclidean distances to $\widetilde{\mathrm{x}}'_{2n_T}$.

4) Select the K candidates with the lowest Euclidean distances and store them.

5) Decrease $l = l - 1$. For each of the K best paths $\widetilde{\mathrm{z}}^i_{l+1:2n_T}$ that were stored in level $l+1$, generate the symbol estimate at layer l using the SIC procedure:

$$\hat{\mathrm{z}}^i_l = \left\lceil \frac{\widetilde{\mathrm{x}}'_l - \widetilde{\mathrm{R}}_{l,l+1:2n_T}\widetilde{\mathrm{z}}^i_{l+1:2n_T}}{\widetilde{\mathrm{R}}_{l,l}} \right\rfloor, \quad (8)$$

where $i \in 1, \ldots, K$.

6) Pick $N > K$ integer values around $\hat{\mathrm{z}}^i_l$ at the l layer and calculate their Euclidean distances to $\hat{\mathrm{x}}^i_l$.

7) Select the K candidates with the lowest Euclidean distances and store them.

8) If the iteration arrives at level 1 of the tree, stop the algorithm and select the best path as $\hat{\mathbf{z}}$. Transform it into $\hat{\mathbf{s}} = 2\mathbf{T}\hat{\mathbf{z}} - \mathbf{d}$, quantize the value of $\hat{\mathbf{s}}$ if it is outside the initial lattice and give the result as an output. Otherwise go to step 5).

5. REDUCTION OF THE NUMBER OF CANDIDATES

As said above, the set of possible $\mathbf{s}$ values comes affected by the inverse of the transformation matrix $\mathbf{T}^{-1}$, resulting in a new vector to be detected called $\mathbf{z}$. It is known that calculating all possible $\mathbf{z}$ values requires a computational cost similar to exhaustive search detection. However, if the minimum and maximum integer value of each of the components of vector $\mathbf{z}$ are known (in what follows called as *boundaries of the transformed lattice*), they can be used to discard those candidate solutions at each level of the tree that are not within these minimum and maximum values, i.e. the candidates that certainly lead to non-valid solutions.

In this work, a novel technique for calculating the boundaries of the transformed lattice is proposed and will be next detailed. As a first approach, let matrix $\mathbf{T}^{-1}$ be available for each channel realization. It can be easily seen that the maximum $\mathbf{z}$ at levels $l \in \{1, \ldots, 2n_T\}$ ($\mathrm{z}_{\max}{}^{(l)}$) appears when the positive entries of row l of $\mathbf{T}^{-1}$ are multiplied by the maximum value of $\widetilde{\mathbf{\Omega}}$ ($\widetilde{\Omega}_{\max}$) and the negative entries multiplied by the minimum value of $\widetilde{\mathbf{\Omega}}$ ($\widetilde{\Omega}_{\min}$). A similar procedure can be followed for calculating the minimum values of $\mathbf{z}$

$(\mathrm{z}_{\min}{}^{(l)})$, since they correspond to the addition of the negative entries of row l of $\mathbf{T}^{-1}$ multiplied by the maximum value of $\widetilde{\Omega}$ ($\widetilde{\Omega}_{\max}$) and the positive entries of the same row multiplied by the minimum value of $\widetilde{\Omega}$ ($\widetilde{\Omega}_{\min}$). The resulting equations can be expressed as

$$\mathrm{z}_{\max}{}^{(l)} = \widetilde{\Omega}_{\max} \sum_{j \in P^{(l)}} \mathrm{T}_{l,j}^{-1} + \widetilde{\Omega}_{\min} \sum_{j \in N^{(l)}} \mathrm{T}_{l,j}^{-1}, \qquad (9)$$

$$\mathrm{z}_{\min}{}^{(l)} = \widetilde{\Omega}_{\min} \sum_{j \in P^{(l)}} \mathrm{T}_{l,j}^{-1} + \widetilde{\Omega}_{\max} \sum_{j \in N^{(l)}} \mathrm{T}_{l,j}^{-1}, \qquad (10)$$

where $P^{(l)}$ stands for the set of j indices where $\mathrm{T}_{l,j}^{-1} > 0$ and $N^{(l)}$ stands for the set of j indices where $\mathrm{T}_{l,j}^{-1} < 0$.

Once the boundaries $\mathrm{z}_{\min}{}^{(l)}$ and $\mathrm{z}_{\max}{}^{(l)}$ have been calculated, they can be stored in the following boundary vector

$$\boldsymbol{\phi}^{(l)} = \left[\phi_1^{(l)}, \phi_2^{(l)}\right]^{\mathrm{T}} = \left[\mathrm{z}_{\min}{}^{(l)}, \mathrm{z}_{\max}{}^{(l)}\right]^{\mathrm{T}}. \qquad (11)$$

Finally, a $2 \times 2n_T$ matrix $\boldsymbol{\Phi}$ will store all these boundary vectors, which will be necessary for the proposed LRA K-Best schemes:

$$\boldsymbol{\Phi} = \left[\boldsymbol{\phi}^{(1)}, \boldsymbol{\phi}^{(2)}, \ldots, \boldsymbol{\phi}^{(2n_T)}\right]. \qquad (12)$$

It can be seen that if matrix $\mathbf{T}^{-1}$ has already been computed, the proposed boundary calculation according to Eqs. (9) and (10) becomes very simple. Another important fact is that this calculation is only performed once per channel realization.

6. PROPOSED LRA K-BEST SCHEMES

Once the boundaries of the transformed lattice are available, they can be used to limit the search for the solution at each level of the decoding tree of the LRA K-Best detector of [4]. In this work, we propose two different approaches aimed at reducing the number of explored candidate solutions. The first approach discards the candidates that fall outside the boundaries, and thus, it is hereafter called LRA K-Best with Candidate Limitation (LRA-CL). The second strategy leads to two different Dynamic LRA K-Best schemes, which are based on choosing the values for K and/or N dynamically.

6.1 LRA K-Best with Candidate Limitation

Considering that the boundaries of the transformed lattice have been calculated as presented in Section 5, the only steps of the original algorithm that have to be modified in order to discard erroneous solutions are steps 3) and 6). Step 6) would be finally described as follows:

6) Pick $N > K$ integer values around $\hat{z}_l^i$ at the l layer following a zig-zag strategy around the SIC solution. In case of reaching the boundaries, follow the same direction and reduce N to the value of $N = \lceil (N/2) \rceil$, where $\lceil \cdot \rceil$ denotes rounding to the higher integer. If the other boundary is also reached, stop exploring candidates. Finally calculate the Euclidean distances of the candidates to $\hat{\mathrm{x}}_l^i$.

Fig. 1 represents an example of the candidate selection following a zig-zag strategy around the SIC solution. It will be considered for this case a value of $N = 8$ and a solution from the SIC detector equal to -2, labelled in the figure as SIC. The first row of numbers surrounded by circles represent the order in which each candidate is explored when there is no candidate limitation. However, when the candidate limitation is applied, if a value for the lower boundary of $\mathrm{z}_{\min} = -3$ is supposed, the value of N is reduced to $N = 4$ and the selected candidates points now range from -3 to 0. Therefore, the proposed candidate limitation can reduce the number of candidates without decreasing the detection performance, as will be discussed below.

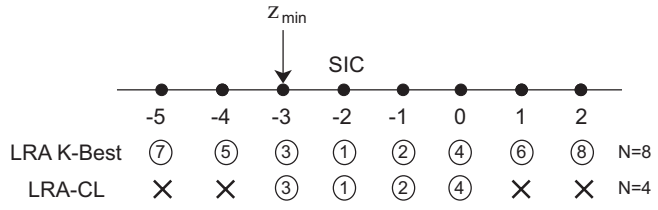

Figure 1: Representation of the order in which candidate points are explored in the conventional LRA K-Best scheme and in the LRA-CL scheme.

The modifications over step 3) can be straightforwardly predicted.

6.2 Dynamic LRA K-Best schemes

In this subsection, a dynamic selection of the parameters of the LRA K-Best detectors is proposed. It must be noted that in this work, the aim is to decrease the complexity of LRA K-Best detectors, which is subject to the two parameters K and N, with $N > K$. Parameter K is responsible for the number of stored paths per level and N for the number of candidate solutions that are explored before choosing the K best ones, as introduced in Section 4.

As it can be seen from the proposed boundary calculation, vector $\phi^{(l)}$ allows to determine the number of valid candidate solutions at each level l of the decoding tree. It seems reasonable that levels with higher number of candidate solutions should not discard as many paths as levels with lower number of candidates. Therefore, the computational complexity of the algorithm can be at a first approach decreased either by having both a different K_l and N_l value at each level (later called as Dynamic-K scheme), or keeping the number of stored paths K unaltered and only assigning different N_l values per level (Dynamic-N scheme). The first approach leads to both a variable number of explored paths $K_{l-1}N_l$ and a variable number of stored paths K_l at each level, whereas the second approach explores KN_l paths and stores K paths (which is a fixed value).

Following the above mentioned strategies, we propose two different dynamic algorithms. In the Dynamic-K (Dyn-K) algorithm there is a non-linear distribution of the K_l values at each level of the tree, by using a generalized logistic function [8] rounded to the higher integer by the operator $\lceil \cdot \rceil$. The resulting K_l values are calculated according to the number of candidates at level l and also to the K value of a same performance fixed K-Best algorithm, as follows:

$$K_l = \left\lceil 1 + \frac{(K-1)}{(1 + 0.5\exp(-(L_l - M)))^2} \right\rceil, \qquad (13)$$

where L_l stands for the number of candidate points at level l:

$$L_l = (\phi_2^{(l)} - \phi_1^{(l)}) + 1 \tag{14}$$

and M denotes the central point of the range of values expanded by $\mathbf{L} = [L_1, \ldots, L_l, \ldots, L_{2n_T}]$

$$M = (\max_{\forall l}(\mathbf{L}) - \min_{\forall l}(\mathbf{L}))/2. \tag{15}$$

Let this distribution of K_l values be clarified by means of an example. Considering a 4×4 complex MIMO system (8×8 in its real-valued representation), working with a 16-QAM constellation, 8 detection levels are necessary to obtain the whole detected vector. At each of these levels, the number of candidate points (L_l) for a given channel realization can be calculated by using (14) and previously related equations. A set of possible values of L_l is given as an example in Table 2 for each level l. The next step for the calculation of the K_l values is the calculation of M, which can be easily performed using (15) as $M = (8-1)/2 = 3.5$.

Once the values of L_l and M are available, a fixed K-Best algorithm to be compared has to be chosen, for instance 5-Best ($K = 5$). Inserting L_l, M and K into (13), the values shown in the third column of Table 2, labelled as K_l, are achieved. It can be easily checked that the average K value is now 3.62, which is obviously decreasing the K value of the fixed 5-Best for this particular case.

Table 2: Example of assigned K_l values in a 8×8 real-valued MIMO system, associated to the set of number of candidates L_l for each of the 8 detection levels.

Level (l)	Candidates (L_l)	Assigned K (K_l)
1	3	3
2	2	2
3	4	4
4	6	5
5	3	3
6	3	3
7	1	1
8	8	5

Considering the smallest value for vector N, N_l can be calculated at each level as $N_l = K_l + 1$.

In the Dyn-K scheme, the fact of working with different K_l values affects steps 4) and 7) of the above described LRA-CL scheme. These steps should be modified in order to operate with the K_l best paths instead of with the K best paths. In the same way, steps 3) and 6) of the above described LRA-CL scheme should be modified in order to operate with the N_l integer values around the SIC solution instead of with the N values. This strategy has been shown to reduce the average number of stored paths (K) and the average number of expanded nodes of the LRA-CL algorithm, without decreasing the performance, as will be presented in the next section. Therefore, this first dynamic approach saves average power consumption in practical implementations.

In the second dynamic scheme, called Dynamic-N algorithm (Dyn-N), the value of K remains constant and the N_l values are now calculated as follows:

$$N_l = \left\lceil 1 + \frac{(K-1)}{(1 + 0.5\exp(-(L_l - M)))^2} \right\rceil. \tag{16}$$

In this case, only steps 3) and 6) of the above described LRA-CL scheme should be modified in order to operate with the N_l integer values around the SIC solution instead of with the N values. This strategy has been shown to reduce the average number of explored paths and to keep the average number of stored paths fixed, since K is the same for all levels.

7. RESULTS AND COMPLEXITY DISCUSSION

The computational complexity comparison has been performed for a 4×4 MIMO system working with 16-QAM and 64-QAM constellations. For the sake of simplicity, the simulations were carried out using the real representation of the MIMO system and an uncoded scheme, although the proposed technique can be easily adapted to a complex system. The LLL algorithm [6] was employed for the lattice-reduction operation.

For a $2n_T \times 2n_T$ MIMO system, it can be easily seen that the number of explored paths at each run of the original LRA K-Best algorithm equals $N + (2n_T - 1)NK$, since every level needs to explore NK candidates except the first level, where only N are explored. In the case considered for our simulations, the number of levels of the decoding tree is 8, and thus, the number of explored paths at each run of the algorithm equals $N + 7NK$. From now on, it will be considered that $N = K + 1$, which leads to a number of explored candidates of $7K^2 + 8K + 1$.

Tables 3 and 4 show the average number of expanded nodes in the LRA K-Best algorithm proposed in [4], compared to the average number of expanded candidates of the LRA-CL. In Table 3 it can be seen that for the 16-QAM case, simulations were run with the LRA 3-Best and LRA 5-Best detectors. For the LRA 3-Best case, the LRA-CL scheme decreases the number of explored candidates in a 19%, whereas for the LRA 5-Best case, the complexity is reduced in a 31.5%. In Table 4 the simulation was performed with a 64-QAM constellation and with the LRA 5-Best and LRA 10-Best detectors. It can be seen that in the 5-Best scheme, the number of candidates is decreased in a 14.4% and in the LRA 10-Best case the reduction is 27% of the candidates. Therefore, it can be concluded that the higher the value of K is, the higher the percentage of reduction can be achieved.

Table 3: Reduction of average number of expanded candidates for the LRA-CL detector (16-QAM case).

LRA 3-Best			LRA 5-Best		
[4]	LRA-CL	Reduction (%)	[4]	LRA-CL	Reduction (%)
88	71	19	216	148	31.5

Table 4: Reduction of average number of expanded candidates for the LRA-CL detector (64-QAM case).

LRA 5-Best			LRA 10-Best		
[4]	LRA-CL	Reduction (%)	[4]	LRA-CL	Reduction (%)
216	185	14.4	781	570	27

Tables 5 and 6 show the average number of expanded candidates for the LRA-CL algorithm and for the two proposed dynamic versions of this method, labelled as Dyn-K

and Dyn-N. It can be seen that dynamic schemes can further reduce the complexity.

Table 5: Average number of expanded candidates for the LRA-CL, Dyn-K and Dyn-N detectors (16-QAM case).

LRA 3-Best			LRA 5-Best		
LRA-CL	Dyn-K	Dyn-N	LRA-CL	Dyn-K	Dyn-N
71	66.2	54.2	148	131.8	124.3

Table 6: Average number of expanded candidates for the LRA-CL, Dyn-K and Dyn-N detectors (64-QAM case).

LRA 5-Best			LRA 10-Best		
LRA-CL	Dyn-K	Dyn-N	LRA-CL	Dyn-K	Dyn-N
185	155	143	570	473	486.7

Table 7 shows the comparison between the average K values for some fixed K-Best detectors and the average K values for the Dynamic-K detection scheme that achieves the same performance. Also, the relative reduction of K in % obtained with the dynamic scheme is presented. It can be observed that the Dyn-K detector reduces the average K value and, thus, the average number of stored paths of the algorithm.

Table 7: Reduction of the average K values with a Dyn-K detector.

16-QAM			64-QAM		
Fixed	Dyn-K	Reduction (%)	Fixed	Dyn-K	Reduction (%)
3	2.8	6.7	5	4.3	14
5	4.4	12	10	8.1	19

Finally, Fig. 2 shows the performance of two of the proposed LRA K-Best schemes, concretely the LRA-CL and Dyn-K with $K = 3$, on a 4×4 MIMO system using 16-QAM, compared to conventional 3-Best, LRA 3-Best and ML detectors. It can be seen that the BER curves of the proposed lower complexity detectors overlap the BER curve of the LRA 3-Best detector proposed in [4], and all of them improve the performance of conventional 3-Best detector for $\rho \geq 20dB$. Therefore, the proposed schemes can decrease the average complexity without modifying the performance at all. It has been checked that the BER curve for the Dyn-N algorithm also overlaps the curves of LRA K-Best, LRA-CL and Dyn-K.

On the other hand, for values of $\rho < 20dB$, the conventional 3-Best detector slightly outperforms the LRA 3-Best schemes. As it can be observed in [3], the ZF-SIC detector also decreases its performance for low SNR regimes when it is combined with lattice reduction. Since the LRA K-Best method uses as starting point the ZF-SIC solution, it seems reasonable to find a similar performance decrease in the proposed LRA K-Best schemes.

8. CONCLUSION

Throughout this paper, several schemes for decreasing the computational complexity of already existing LRA K-Best detectors have been proposed. The first one has been called as LRA K-Best scheme with Candidate Limitation (LRA-CL). This approach calculates the boundaries of the transformed lattice in order to discard in the LRA K-Best scheme those candidates that are for sure outside the transformed lattice. The LRA-CL has been shown to decrease the average computational complexity. In addition, two schemes with a dynamic distribution of the parameters K and/or N have been proposed. The dynamic distribution of the parameters is based on the number of possible candidates existing in the transformed lattice. The computational complexity can be further decreased by means of these last schemes without performance loss.

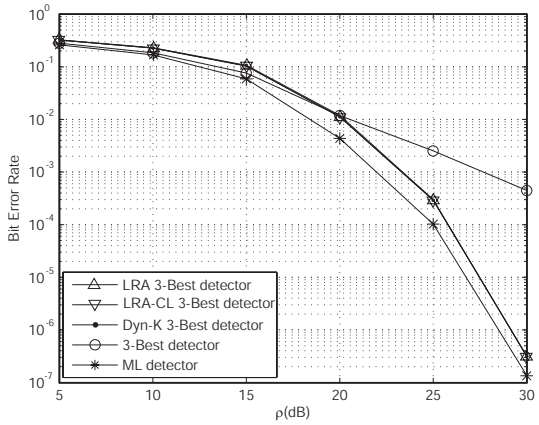

Figure 2: BER curves of the proposed LRA K-Best schemes (LRA-CL and Dyn-K) on a 4×4 MIMO system using 16-QAM, compared to conventional 3-Best, LRA 3-Best and ML detectors.

REFERENCES

[1] C. Windpassinger and R. F. H. Fischer, "Low-complexity near-maximum-likelihood detection and precoding for MIMO systems using lattice reduction," in *IEEE Inf. Theory Workshop*, March 2003, pp. 345–348.

[2] H. Yao and G. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *IEEE GLOBECOM*, vol. 1, 2002, pp. 424–428.

[3] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "Near-Maximum-Likelihood Detection of MIMO Systems using MMSE-Based Lattice Reduction," in *IEEE International Conference on Communications*, Paris, France, June 2004.

[4] X.-F. Qi and K. Holt, "A Lattice-Reduction-Aided Soft Demapper for High-Rate Coded MIMO-OFDM Systems," *IEEE Signal Processing Letters*, vol. 14, no. 5, pp. 305 – 308, May 2007.

[5] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-Best Sphere Decoding for MIMO Detection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, March 2006.

[6] A. Lenstra, H. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515 – 534, 1982.

[7] C. Studer, A. Burg, and W. Fichtner, "A unification of ML-optimal tree-search decoders," in *Fortieth Asilomar Conference on Signals, Systems and Computers, 2006 (ACSSC '06)*.

[8] F. J. Richards, "A flexible growth function for empirical use," *J. Exp. Bot.*, vol. 10, pp. 290–300, 1959.