On Maximum Likelihood Detection and the Search for the Closest Lattice Point

Mohamed Oussama Damen*, Hesham El Gamal†, and Giuseppe Caire‡

Abstract— **Maximum likelihood decoding algorithms for Gaussian MIMO linear channels are considered. Linearity over the field of real numbers facilitates the design of maximum likelihood decoders using number theoretic tools for searching the** *closest lattice point.* **These decoders are collectively referred to as** *sphere decoders* **in the literature. In this paper, a fresh look at this class of decoding algorithms is taken. In particular, two novel algorithms are developed. The first algorithm is inspired by the Pohst enumeration strategy and is shown to offer a significant reduction in complexity compared to the Viterbo-Boutros sphere decoder. The connection between the proposed algorithm and the stack sequential decoding algorithm is then established. This connection is utilized to construct the second algorithm which can also be viewed as an application of the Schnorr-Euchner strategy to maximum likelihood decoding. Aided with a detailed study of pre-processing algorithms, a variant of the second algorithm is developed and shown to offer significant reductions in the computational complexity compared to all previously proposed sphere decoders with a** *near* **maximum likelihood detection performance. This claim is supported by intuitive arguments and simulation results in many relevant scenarios.**

Index Terms– **Decision feedback equalization (DFE), fading channels, lattices, lattice reduction, maximum-likelihood (ML) detection, minimum mean-square error (MMSE), multi-input multi-output (MIMO) systems, Pohst enumeration, sequential decoding, space-time constellations, stack algorithms.**

I. Introduction

In several communication problems, the received signal is given by a linear combination of the data symbols corrupted by additive noise, where linearity is defined over the field of real numbers. The input-output relation describing such channels can be put in the form of the real multi-input multi-output (MIMO) linear model

$$\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{z} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{y}, \mathbf{z} \in \mathbb{R}^n$ denote the channel input, output and noise signals, and $\mathbf{B} \in \mathbb{R}^{n \times m}$ is a matrix representing the channel linear mapping. Typically, the noise components $z_j, j = 1, \ldots, n$, are independent, identically distributed (i.i.d.) zero-mean Gaussian random variables with a common variance, and the information signal $\mathbf{x}$ is uniformly distributed over a discrete and finite set $\mathcal{C} \subset \mathbb{R}^m$, representing the transmitter *codebook*. Under such conditions and assuming $\mathbf{B}$ perfectly known at the receiver, the optimal detector $g : \mathbf{y} \mapsto \widehat{\mathbf{x}} \in \mathcal{C}$ that minimizes the average error probability $P(e) \stackrel{\Delta}{=} P(\widehat{\mathbf{x}} \neq \mathbf{x})$ is the maximum-likelihood (ML) detector given by

$$\widehat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{C}} \left| \mathbf{y} - \mathbf{B}\mathbf{x} \right|^2. \tag{2}$$

For the sake of simplicity, we assume that $\mathcal{C} = \mathcal{X}^m$, where $\mathcal{X}$ is a PAM signal set [1] of size Q, i.e.,

$$\mathcal{X} = \{u = 2q - Q + 1 \ : \ q \in \mathbb{Z}_Q\} \tag{3}$$

with $\mathbb{Z}_Q \stackrel{\Delta}{=} \{0, 1, \ldots, Q-1\}$. More general signal sets will be briefly considered in Section V.

Under the assumption (3), by applying a suitable translation and scaling of the received signal vector, (2) takes on the *normalized* form

$$\widehat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{Z}_Q^m} \left| \mathbf{y} - \mathbf{B}\mathbf{x} \right|^2 \tag{4}$$

where the components of the noise $\mathbf{z}$ have a common variance equal to 1.

In this paper we consider a class of algorithms, generally known as sphere decoders [2], [3], [4], [5], [6], [8], that compute or approximate (4) with a polynomial expected complexity within a wide range of system parameters. Throughout the paper, the terms "decoding" and "detection" are used interchangeably to refer to the same procedure. Before proceeding further, we review some applications that provide the main motivation for the rest of the work.

1. **Linear-dispersion Encoded QAM over a Frequency-flat MIMO Channel:**
Consider an M-transmit, N-receive antennas system with frequency-flat quasi-static fading [9], [10]. The baseband complex received signal after matched filtering and symbol rate sampling is given by

$$\mathbf{r}_t = \sqrt{\frac{\mathrm{SNR}}{M}} \mathbf{H}\mathbf{s}_t + \boldsymbol{\nu}_t, \quad t = 1, \ldots, T \tag{5}$$

where $\mathbf{H}$ is the $N \times M$ channel matrix whose (i,j)-th element $h_{i,j}$ is the complex fading gain from transmit antenna j to receive antenna i and $\boldsymbol{\nu}_t \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{I})$, with $\mathbf{I}$ denoting the identity matrix of appropriate dimension, is a sequence of independent proper Gaussian noise vectors with i.i.d. components. Assuming $E[\mathbf{s}_t \mathbf{s}_t^H] = \mathbf{I}$ and $E[|h_{i,j}|^2] = 1$, the factor SNR in (5) denotes the signal-to-noise ratio per receiving antenna defined as the ratio of the total transmit energy per channel use divided by the per-component noise variance.

This work was initiated when M. O. Damen was visiting the Ohio-State University during the summer of 2002. The work of H. El Gamal was funded in part by the National Science Foundation under Grant CCR 0118859.

* M. O. Damen is the Department of Electrical and Computer Engineering, University of Alberta, ECERF, Edmonton, AB, T6G 2V4, Canada. (e-mail: damen@ece.ualberta.ca).

† H. El Gamal is with the Department of Electrical Engineering, the Ohio State University, Columbus, OH. (e-mail: helgamal@ee.eng.ohio-state.edu).

‡ G. Caire is with the Mobile Communications Group, Institut EURECOM, B.P. 193, 06904 Sophia-Antipolis, France. (e-mail: giuseppe.caire@eurecom.fr)

Let $\mathcal{U}$ denote a squared QAM signal set with Q^2 signal points [1]. Assume that the transmitter maps blocks $\mathbf{u} \in \mathcal{U}^L$ with independent and uniformly distributed entries (information symbols) onto $M \times T$ transmit arrays of the form

$$\mathbf{S}(\mathbf{u}) = \sum_{\ell=1}^{L} (\mathbf{F}_\ell u_\ell + \mathbf{G}_\ell u_\ell^*) \tag{6}$$

where $\{\mathbf{F}_\ell, \mathbf{G}_\ell \in \mathbb{C}^{M\times T} : \ell = 1, \ldots, L\}$ are the linear dispersion code *generators* [11], [12], [13]. Then, the array $\mathbf{S}(\mathbf{u}) = [\mathbf{s}_1(\mathbf{u}), \ldots, \mathbf{s}_T(\mathbf{u})]$ is transmitted column by column in T channel uses. Due to the linearity of the channel (5) and of the encoder (6), one can easily see that there exists a matrix $\mathbf{B} \in \mathbb{R}^{2NT\times 2L}$ such that the ML detection can be written in the form (4). In general, matrix $\mathbf{B}$ depends on the physical channel matrix $\mathbf{H}$ and on the code generators $\{\mathbf{F}_\ell, \mathbf{G}_\ell\}$ (see [11] for an explicit expression of the matrix $\mathbf{B}$ in terms of $\mathbf{H}$ and the code generators). A particularly simple case corresponds to the transmission of uncoded QAM symbols. In this case, $L = M$, $T = 1$, $\mathbf{S}(\mathbf{u}) = \mathbf{u}$ and $\mathbf{B}$ is given explicitly by

$$\mathbf{B} = \sqrt{\frac{12\ \mathrm{SNR}}{M(Q^2-1)}} \begin{bmatrix} \mathrm{Re}\{\mathbf{H}\} & -\mathrm{Im}\{\mathbf{H}\} \\ \mathrm{Im}\{\mathbf{H}\} & \mathrm{Re}\{\mathbf{H}\} \end{bmatrix}. \tag{7}$$

$\Diamond$

2. **Block Transmission over Time-selective or Frequency-selective Fading:**
In [14], linearly block-coded transmission for time-selective fading was considered. The complex baseband channel is given by $\mathbf{r} = \mathbf{HMu} + \boldsymbol{\nu}$, where now $\mathbf{H} = \mathrm{diag}(h_1, \ldots, h_N)$, $\{h_i\}$ is the sequence of scalar fading coefficients, and $\mathbf{M}$ is a suitable rotation matrix. Similarly, for frequency-selective slow fading we can work in the frequency domain to obtain linearly precoded OFDM schemes (see for example [15] and references therein) in the same form as above, where $\mathbf{H} = \mathrm{diag}(H_1, \ldots, H_N)$ and H_i denotes the fading channel frequency response at the i-th OFDM subcarrier. $\Diamond$

3. **Multiple-access Gaussian Waveform Channel (CDMA):**
The canonical baseband complex model for a K-user synchronous CDMA system is given by [16]

$$\mathbf{y}_t = \mathbf{HWu}_t + \boldsymbol{\nu}_t \tag{8}$$

where the columns of $\mathbf{H} \in \mathbb{C}^{N\times K}$ represent the (discrete-time) users' signature waveforms, $\mathbf{W}$ is a diagonal $K \times K$ matrix of amplitudes and $\mathbf{u}_t$ is a $K \times 1$ vector containing the modulation symbols transmitted by the users in the t-th symbol interval. In most CDMA systems the user symbols take on values in the 4-QAM signal set. Again, after suitable translation and scaling, the ML joint detection of the K user symbols can be put in the form (4) [17], [18]. $\Diamond$

It is well-known that the minimization (4) for arbitrary $\mathbf{B}$ and $\mathbf{y}$ is NP-hard (see [3], [6], [19], [20]). Nevertheless, it has been shown recently that in many relevant cases, for a certain range of system parameters such as SNR, m, n and Q, the *average* complexity of some algorithms implementing or approximating ML detection is polynomial in m (roughly, $O(m^3)$). The reason of this behavior is that, in (1), the received point $\mathbf{y}$ is not arbitrary, but it is obtained by perturbing the transmitted point $\mathbf{Bx}$ by the additive noise $\mathbf{z}$. Therefore, it can be expected that as the SNR increases the average complexity decreases, for fixed m, n and Q.

This fact has been shown by theoretical analysis in [21] in the case of uncoded QAM over a $N \times M$ frequency flat MIMO channel with $N \geq M$, $\mathbf{H}$ having i.i.d. entries $\sim \mathcal{N}_{\mathbb{C}}(0, 1)$, and the *basic* sphere decoder, known as Pohst enumeration (see details later). However, the same fact has been observed in more general cases by computer simulations in several recent works (e.g., [22], [23], [24]). While the exact average complexity analysis of the basic sphere decoder for general linear-dispersion codes and, *a fortiori*, for improved sphere decoding algorithms as those proposed in this work, appear to be intractable, finding improvements on the existing algorithms and illuminating the trade-offs and relationships between the different approaches is relevant on its own.

Developing efficient sphere decoders to solve or approximate (4) has recently gained renewed attention mainly because of their applications to multiple-antenna systems [8]. This interest is due to the significant performance gain achieved by sphere decoders compared to other sub-optimal detection schemes [24], [25], [26], and to their average polynomial complexity (experimentally demonstrated) at medium to high SNR. Moreover, the ML detector can be easily augmented to provide symbol-by-symbol soft output, in the form of approximated posterior marginal probabilities $p(x_i|\mathbf{y})$. The resulting soft-output detector forms the core of some iterative decoders based on Belief-Propagation (see for example [27], [28], [29], [30], [31]).

The main contribution of this paper is a fresh look at the class of sphere decoding algorithms. We start by reviewing the basic Pohst and Schnorr-Euchner enumeration strategies for infinite lattices in Section II. In Section III we propose two variants of the basic algorithms in order to take into account finite PAM signal sets. We also show that the proposed algorithms can be interpreted as a chain of sequential decoding stages, where each stage is based on a special case of the *stack algorithm* [32]. This observation makes precise the intuition given in [27] that sphere decoding and sequential decoding are "similar algorithms", and provides a path for cross-fertilization between the rich bodies of work on sequential and lattice decoding. In Section IV, we discuss some heuristics for pre-processing aimed at transforming the problem so that the resulting algorithm has lower average complexity while still being ML (or near ML). In Section V we discuss some extensions of the proposed algorithms to handle general lattice codes and non-invertible channel matrices. Finally, the performance of the proposed algorithms and the impact of various pre-processing methods are investigated through computer simulations in a number of relevant examples in Section VI. This simulation study, along with some intuitive arguments, lead to the conclusion that one of the proposed algorithms uniformly outperforms all known sphere decoders in terms of receiver complexity, while offering near-ML detection performance (i.e., the performance is within a very small fraction of a dB from ML in all considered scenarios).

II. The Pohst and Schnorr-Euchner Enumerations

For $\mathbf{B} \in \mathbb{R}^{n\times m}$ and $\mathbf{y} \in \mathbb{R}^n$, consider the minimization

$$\widehat{\mathbf{x}} = \arg \min_{\mathbf{x}\in\mathbb{Z}^m} |\mathbf{y} - \mathbf{B}\mathbf{x}|^2. \tag{9}$$

This is analogous to (4) but $\mathbb{Z}_Q$ is replaced by $\mathbb{Z}$, the (infinite) ring of integer numbers. The set $\Lambda = \{\mathbf{B}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^m\}$ is an m-dimensional lattice in $\mathbb{R}^n$ [33]. The search in (9) for the *closest lattice point* to a given point $\mathbf{y}$ has been widely investigated in lattice theory. In general, the optimal search algorithm should exploit the structure of the lattice. For general lattices, that do not exhibit any particular structure, the problem was shown to be NP-hard. In [2], however, Pohst proposed an efficient strategy for enumerating all the lattice points within a sphere with a certain radius. Although its worst-case complexity is exponential in m, this strategy has been widely used ever since in *closest lattice point* search problems due to its efficiency in many useful scenarios (see [6] for a comprehensive review of related works).

The Pohst enumeration strategy was first introduced in digital communications by Viterbo and Biglieri [4]. In [5], Viterbo and Boutros applied it to the ML detection of multi-dimensional constellations transmitted over single antenna fading channels, and gave a flowchart of a specific implementation. More recently, Agrell *et al.* [6] proposed the use of the Schnorr-Euchner refinement [7] of the Pohst enumeration in the *closest lattice point* search. They further concluded, based on numerical results, that the Schnorr-Euchner enumeration is more efficient than the Viterbo-Boutros (VB) implementation.

The Pohst enumeration is briefly outlined as follows. Assume that $n \geq m$ and $\text{rank}(\mathbf{B}) = m$. Let C_0 be the squared radius of an n-dimensional sphere $\mathcal{S}(\mathbf{y}, \sqrt{C_0})$ centered at $\mathbf{y}$. We wish to produce a list of all points of $\Lambda \cap \mathcal{S}(\mathbf{y}, \sqrt{C_0})$. By performing the Gram-Schmidt orthonormalization of the columns of $\mathbf{B}$ (equivalently, by applying QR decomposition on $\mathbf{B}$), one writes

$$\mathbf{B} = [\mathbf{Q}, \mathbf{Q}'] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{10}$$

where $\mathbf{R}$ is an $m\times m$ upper triangular matrix with positive diagonal elements, $\mathbf{0}$ is an $(n-m)\times m$ zero matrix and $\mathbf{Q}$ (resp., $\mathbf{Q}'$) is an $n \times m$ (resp., $n \times (n-m)$) unitary matrix. The condition $\mathbf{B}\mathbf{x} \in \mathcal{S}(\mathbf{y}, \sqrt{C_0})$ can be written as

$$\begin{aligned} |\mathbf{y} - \mathbf{B}\mathbf{x}|^2 &\leq C_0 \\ \left|[\mathbf{Q}, \mathbf{Q}']^T\mathbf{y} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\mathbf{x}\right|^2 &\leq C_0 \\ \left|\mathbf{Q}^T\mathbf{y} - \mathbf{R}\mathbf{x}\right|^2 &\leq C_0 - \left|(\mathbf{Q}')^T\mathbf{y}\right|^2 \\ |\mathbf{y}' - \mathbf{R}\mathbf{x}|^2 &\leq C_0' \end{aligned} \tag{11}$$

where $\mathbf{y}' \triangleq \mathbf{Q}^T\mathbf{y}$ and $C_0' \triangleq C_0 - \left|(\mathbf{Q}')^T\mathbf{y}\right|^2$. Due to the upper triangular form of $\mathbf{R}$, the last inequality implies the set of conditions

$$\sum_{j=i}^{m} \left| y_j' - \sum_{\ell=j}^{m} r_{j,\ell}x_\ell \right|^2 \leq C_0', \quad i = 1, \ldots, m. \tag{12}$$

By considering the above conditions in the order from m to 1 (akin to back-substitution in the solution of a linear upper triangular system), we obtain the set of admissible values of each symbol x_i for given values of symbols $x_{i+1}, \ldots, x_m$. More explicitly, let $\mathbf{x}_\ell^m \triangleq (x_\ell, x_{\ell+1}, \ldots, x_m)^T$ denote the last $m-\ell+1$ components of the vector $\mathbf{x}$. For a fixed $\mathbf{x}_{i+1}^m$, the component x_i can take on values in the range of integers $\mathcal{I}_i(\mathbf{x}_{i+1}^m) = [A_i(\mathbf{x}_{i+1}^m), B_i(\mathbf{x}_{i+1}^m)]$ where

$$\begin{aligned} &A_i(\mathbf{x}_{i+1}^m) = \\ &\left\lceil \frac{1}{r_{i,i}} \left(y_i' - \sum_{j=i+1}^{m} r_{i,j}x_j - \sqrt{C_0' - \sum_{j=i+1}^{m} \left| y_j' - \sum_{\ell=j}^{m} r_{j,\ell}x_\ell \right|^2} \right) \right\rceil, \\ &B_i(\mathbf{x}_{i+1}^m) = \\ &\left\lfloor \frac{1}{r_{i,i}} \left(y_i' - \sum_{j=i+1}^{m} r_{i,j}x_j + \sqrt{C_0' - \sum_{j=i+1}^{m} \left| y_j' - \sum_{\ell=j}^{m} r_{j,\ell}x_\ell \right|^2} \right) \right\rfloor. \end{aligned} \tag{13}$$

If $\sum_{j=i+1}^{m} \left| y_j' - \sum_{\ell=j}^{m} r_{j,\ell}x_\ell \right|^2 > C_0'$ or if $A_i(\mathbf{x}_{i+1}^m) > B_i(\mathbf{x}_{i+1}^m)$, then $\mathcal{I}_i(\mathbf{x}_{i+1}^m) = \emptyset$ (the empty set). In this case, there is no value of x_i satisfying the inequalities (12) and the points corresponding to this choice of $\mathbf{x}_{i+1}^m$ do not belong to the sphere $\mathcal{S}(\mathbf{y}, \sqrt{C_0})$.

The Pohst enumeration consists of spanning at each level i the admissible *interval* $\mathcal{I}_i(\mathbf{x}_{i+1}^m)$, starting from level $i = m$ and climbing "up" to level $i = m-1, m-2, \ldots, 1$. At each level, the interval $\mathcal{I}_i(\mathbf{x}_{i+1}^m)$ is determined by the current values of the variables at lower levels (corresponding to higher indices). If $\mathcal{I}_1(\mathbf{x}_2^m)$ is non-empty, the vectors $\mathbf{x} = (x_1, (\mathbf{x}_2^m)^T)^T$, for all $x_1 \in \mathcal{I}_1(\mathbf{x}_2^m)$, yield lattice points $\mathbf{B}\mathbf{x} \in \mathcal{S}(\mathbf{y}, \sqrt{C_0})$. The squared Euclidean distances between such points and $\mathbf{y}$ are given by $d^2(\mathbf{y}, \mathbf{B}\mathbf{x}) = \sum_{j=1}^{m} \left| y_j' - \sum_{\ell=j}^{m} r_{j,\ell}x_\ell \right|^2$. The algorithm outputs the point $\widehat{\mathbf{x}}$ for which this distance is minimum. If, after spanning the interval $\mathcal{I}_m$ corresponding to x_m (ground level), no point in the sphere is found, the sphere is declared empty and the search fails. In this case, the search squared radius C_0 must be increased and the search is restarted with the new squared radius.

The Pohst enumeration is based on the so-called *natural spanning* of the intervals $\mathcal{I}_i(\mathbf{x}_{i+1}^m)$ at each level i, i.e., x_i takes on values in the order $A_i(\mathbf{x}_{i+1}^m), A_i(\mathbf{x}_{i+1}^m)+1, \ldots, B_i(\mathbf{x}_{i+1}^m)$. Schnorr-Euchner enumeration is a variation of the Pohst strategy where the intervals are spanned in a zig-zag order, starting from the mid-point

$$S_i(\mathbf{x}_{i+1}^m) = \left\lfloor \frac{1}{r_{i,i}} \left(y_i' - \sum_{j=i+1}^{m} r_{i,j}x_j \right) \right\rceil \tag{14}$$

(where $\lfloor \cdot \rceil$ denotes rounding to the closest integer). Hence, the Schnorr-Euchner enumeration will produce at each level i the (ordered) sequence of values

$$x_i \in \{S_i(\mathbf{x}_{i+1}^m), S_i(\mathbf{x}_{i+1}^m)+1, S_i(\mathbf{x}_{i+1}^m)-1, S_i(\mathbf{x}_{i+1}^m)+2, S_i(\mathbf{x}_{i+1}^m)-2, \ldots\} \cap \mathcal{I}_i(\mathbf{x}_{i+1}^m)$$

if $y_i' - \sum_{j=i+1}^{m} r_{i,j}x_j - r_{i,i}S_i(\mathbf{x}_{i+1}^m) \geq 0$, or the (ordered) sequence of values

$$x_i \in \{S_i(\mathbf{x}_{i+1}^m), S_i(\mathbf{x}_{i+1}^m)-1, S_i(\mathbf{x}_{i+1}^m)+1, S_i(\mathbf{x}_{i+1}^m)-2, S_i(\mathbf{x}_{i+1}^m)+2, \ldots\} \cap \mathcal{I}_i(\mathbf{x}_{i+1}^m)$$

if $y_i' - \sum_{j=i+1}^{m} r_{i,j}x_j - r_{i,i}S_i(\mathbf{x}_{i+1}^m) < 0$. Similar to the Pohst enumeration, when a given value of x_i results in a point segment $\mathbf{x}_i^m$ outside the sphere, the next value of x_{i+1} (at level $i+1$) is produced.

Note that with the Schnorr-Euchner enumeration one can set $C_0 = \infty$. Obviously, in this way the event of declaring an empty sphere never occurs. It is also easy to see that the first point found with $C_0 = \infty$ corresponds to the Babai point [6]. In the communication theory parlance this point is referred to as the *nulling and canceling* [25], or *zero-forcing decision-feedback equalization* (ZF-DFE) point. Explicitly, this is given by the back-substitution with integer quantization (slicing)

$$\begin{aligned} x_i^{\mathrm{zf-dfe}} &= S_i(x_{i+1}^{\mathrm{zf-dfe}}, \ldots, x_m^{\mathrm{zf-dfe}}) \\ &= \left\lfloor \frac{1}{r_{i,i}} \left(y_i' - \sum_{j=i+1}^{m} r_{i,j} x_j^{\mathrm{zf-dfe}} \right) \right\rceil \end{aligned} \quad (15)$$

for $i = m, m-1, \ldots, 1$. As we shall see in Section VI, the drawback of setting $C_0 = \infty$ is that the distance $d^2(\mathbf{y}, \mathbf{B}\mathbf{x}^{\mathrm{zf-dfe}})$ might be quite large, therefore the algorithm will eventually span a large number of points before finding the ML solution.

III. ML Detection of Finite Alphabet Constellations

Probably the most immediate application of the Pohst enumeration to solve (4) consists of the following steps:

1. Fix C_0 according to some criterion (see the discussion in Section VI).
2. Apply the Pohst enumeration with the interval boundaries modified as

$$\begin{aligned} A_i(\mathbf{x}_{i+1}^m) &= \max\{0, \\ &\left\lceil \frac{1}{r_{i,i}} \left(y_i' - \sum_{j=i+1}^{m} r_{i,j}x_j - \sqrt{C_0' - \sum_{j=i+1}^{m} \left| y_j' - \sum_{\ell=j}^{m} r_{j,\ell}x_\ell \right|^2} \right) \right\rceil \Bigg\} \\ B_i(\mathbf{x}_{i+1}^m) &= \min\{Q-1, \\ &\left\lfloor \frac{1}{r_{i,i}} \left(y_i' - \sum_{j=i+1}^{m} r_{i,j}x_j + \sqrt{C_0' - \sum_{j=i+1}^{m} \left| y_j' - \sum_{\ell=j}^{m} r_{j,\ell}x_\ell \right|^2} \right) \right\rfloor \Bigg\} \end{aligned} \quad (16)$$

and obtain the list of all vectors $\mathbf{x} \in \mathbb{Z}_Q^m$ such that $\mathbf{B}\mathbf{x} \in \mathcal{S}(\mathbf{y}, \sqrt{C_0})$.
3. If the list is non-empty, output the point achieving minimum distance (i.e., the ML decision). Otherwise, increase C_0 and search again.

The average complexity of this simple algorithm has been given in closed form in [21] for the special case where $\mathbf{B}$ is a random matrix with i.i.d. entries $\sim \mathcal{N}(0,1)$. As anticipated in the introduction, this corresponds to the transmission of uncoded QAM over a MIMO frequency-flat independent Rayleigh fading channel.

The above sphere decoder can be improved in several ways. The VB implementation is essentially the same algorithm given above, but C_0 is changed adaptively along the search: as soon as a vector $\mathbf{x} \in \mathbb{Z}_Q^m$ is found such that $\mathbf{B}\mathbf{x} \in \mathcal{S}(\mathbf{y}, \sqrt{C_0})$, then C_0 is updated as

$$C_0 \leftarrow d^2(\mathbf{y}, \mathbf{B}\mathbf{x})$$

and the search is restarted in the new sphere with the smaller radius. The drawback of this approach is that the VB algorithm may re-span values of x_i for some levels i, $1 < i \leq m$, that have already been spanned in the previous sphere.

Next, we give the details of two new sphere decoding algorithms. The first, namely Algorithm I, is similar to the VB algorithm, but avoids re-spanning already spanned point segments. The second, namely Algorithm II, can be seen as a modification of the Schnorr-Euchner enumeration in order to take into account the finite signal set boundary. Interestingly, both algorithms are *functionally equivalent* to a chain of *stack sequential decoders* [32], where the stack content and path metric of each decoder depend on the outcome of the previous decoder.

Algorithm I (Input C_0', $\mathbf{y}'$, $\mathbf{R}$. Output $\widehat{\mathbf{x}}$):

Step 1. (Initialization) Set $i := m$, $T_m := 0$, $\xi_m := 0$ $d_c = C_0'$ (current sphere squared radius).

Step 2. (Bounds on x_i) If $d_c < T_i$ go to 4. Else, $A_i(\mathbf{x}_{i+1}^m) := \max\left\{0, \left\lceil \frac{y_i' - \xi_i - \sqrt{d_c - T_i}}{r_{i,i}} \right\rceil \right\}$, $B_i(\mathbf{x}_{i+1}^m) := \min\left\{Q-1, \left\lfloor \frac{y_i' - \xi_i + \sqrt{d_c - T_i}}{r_{i,i}} \right\rfloor \right\}$, and set $x_i := A_i(\mathbf{x}_{i+1}^m) - 1$.

Step 3. (Natural spanning of the interval $\mathcal{I}_i(\mathbf{x}_{i+1}^m)$) $x_i := x_i + 1$. If $x_i \leq B_i(\mathbf{x}_{i+1}^m)$ go to 5, else go to 4.

Step 4. (Increase i: move one level down) If $i = m$ terminate, else set $i := i+1$ and go to 3.

Step 5. (Decrease i: move one level up) If $i > 1$, then $\Big\{$ let $\xi_{i-1} := \sum_{j=i}^{m} r_{i-1,j}x_j$, $T_{i-1} := T_i + |y'_i - \xi_i - r_{i,i}x_i|^2$, let $i := i-1$ and go to 2 $\Big\}$.

Step 6. (A valid point is found) Compute $\widehat{d} := T_1 + |y'_1 - \xi_1 - r_{1,1}x_1|^2$. If $\widehat{d} < d_c$ $\Big\{$ let $d_c := \widehat{d}$, save $\widehat{\mathbf{x}} := \mathbf{x}$, and update the upper boundaries $B_\ell(\mathbf{x}_{\ell+1}^m) := \min\left\{Q-1, \left\lfloor \frac{y'_\ell - \xi_\ell + \sqrt{d_c - T_\ell}}{r_{\ell,\ell}} \right\rfloor\right\}$, for all $\ell = 1, \ldots, m$ $\Big\}$. Go to 3.

□

As in the VB implementation, if no valid point is found, C'_0 is increased and the algorithm is restarted. Note that the variable $\xi_i, i = m, \ldots, 1$, is the decision feedback of a ZF-DFE when the decisions on the symbols from $i+1$ to m are the current values of $(x_{i+1}, \ldots, x_m)$.

As it will become clear in the following, it is useful to visualize sphere decoders for finite PAM signals as a *bounded search* in a tree. In fact, thanks to the upper triangular form of $\mathbf{R}$, the symbol vectors $\mathbf{x} \in \mathbb{Z}_Q^m$ can be represented as paths in a tree of depth m, where the possible values of symbol x_i at level i correspond to branches at depth $m-i+1$. For example, Fig. 1 shows the tree for $Q = 2$ and $m = 4$, corresponding to the case of uncoded 4-QAM transmission over a multiple antenna channel with $M = 2$ and $N \geq 2$. Each branch at depth $m-i+1$ is labeled by the branch metric defined by

$$w_i(\mathbf{x}_i^m) \triangleq \left| y'_i - \sum_{j=i}^{m} r_{i,j}x_j \right|^2 \tag{17}$$

where $\mathbf{x}_i^m \triangleq (x_i, x_{i+1}, \ldots, x_m)^T$ are the symbols labeling the path connecting the branch with the root. The path metric for path $\mathbf{x}_i^m$ is defined as

$$\mathcal{M}(\mathbf{x}_i^m) \triangleq \sum_{j=i}^{m} w_j(\mathbf{x}_j^m) \tag{18}$$

and coincides with the term T_{i-1} in Algorithm I.

Generally speaking, a sphere decoding algorithm explores the tree of all possible symbol sequences and uses the path metric in order to discard paths corresponding to points outside the search sphere.

The main advantage of Algorithm I over the VB algorithm is that once we find a lattice point, we just update all the upper bounds of the intervals without restarting. In other words, partial paths in the tree that have already been examined will not be reconsidered after reducing the sphere radius. We can prove the following

Proposition 1: For a given $\mathbf{R}, \mathbf{y}'$ and C'_0, the number of tree nodes visited by Algorithm I is upper bounded by the number of tree nodes visited by the original Pohst enumeration and by the VB algorithm.

Proof: We give the proof for the VB implementation; the proof is clear for the original Pohst algorithm. To this end, it suffices to show that the lower bounds $A_i(\mathbf{x}_{i+1}^m)$ when updating d_c by $\widehat{d}$ and restarting from Step 1 are smaller than the current values of the components of $\widehat{\mathbf{x}}$ (Step 6 in Algorithm I). But this is clear since the current point is inside the sphere of squared radius $\widehat{d}$. □

Sequential decoders comprise a set of efficient and powerful decoding techniques able to perform close to ML decoding, without suffering the complexity of exact ML decoding, for coding rates not too close to the channel capacity [32], [34]. Next, we interpret Algorithm I as a chain of sequential decoders. To facilitate this interpretation, we make use of the *stack sequential decoding algorithm*, briefly summarized as follows. Consider a tree of depth m, where each branch at level $m-i+1$ is labeled by $x_i \in \mathbb{Z}_Q$ and is associated with a weight $w_i(\mathbf{x}_i^m)$ which depends, in general, on both i and the path $\mathbf{x}_i^m$ connecting the branch with the root. The path metric $\mathcal{M}(\mathbf{x}_i^m)$ associated with the path $\mathbf{x}_i^m$ is given by (18). The stack algorithm is defined by a sorting rule used in conjunction with the above path metric. At the beginning, the stack contains only the root of the tree with an associated metric equal to zero. At each step, the algorithm sorts the stack according to the sorting rule and expands the path at the top of the stack, say $\mathbf{x}_i^m$, by generating the Q extensions $\{\mathbf{x}_{i-1}^m : x_{i-1} \in \mathbb{Z}_Q\}$. Then, for each extension, the algorithm computes the associated path metric as $\mathcal{M}(\mathbf{x}_{i-1}^m) = \mathcal{M}(\mathbf{x}_i^m) + w_{i-1}(\mathbf{x}_{i-1}^m)$, and substitutes the original path with all its extensions. The algorithm stops when one path in the stack is fully extended (i.e., it reaches depth m).

Now, consider the branch metric (17) and the following stack sorting rule (denoted by the ordering relation $\succeq$):

Sorting rule I. If $\mathcal{I}_{i-1}(\mathbf{x}_i^m) = \emptyset$, the path $\mathbf{x}_i^m$ cannot be extended and should be eliminated from the stack. Consider two paths $\mathbf{x}_i^m$ and $\mathbf{u}_j^m$ such that both $\mathcal{I}_{i-1}(\mathbf{x}_i^m)$ and $\mathcal{I}_{j-1}(\mathbf{u}_j^m)$ are non-empty. Then, $\mathbf{x}_i^m \succeq \mathbf{u}_j^m$ if either $i < j$ or $i = j$ and $x_i \leq u_j$. □

In words, rule I says that path $\mathbf{x}_i^m$ has priority higher than path $\mathbf{u}_j^m$ if it can be extended without violating the sphere (condition $\mathcal{I}_{i-1}(\mathbf{x}_i^m) \neq \emptyset$) and either it has depth larger than $\mathbf{u}_j^m$ or, if both paths have the same depth, it has a lexicographic priority (the condition $x_i \leq u_j$ for $i = j$). Lexicographic ordering corresponds to the natural spanning of the Pohst enumeration.

Suppose that, by applying the stack algorithm for the first time with the above metric and sorting rule, we obtain one point in the sphere, say $\widehat{\mathbf{x}}_1$. Then, we can apply the stack algorithm again, by letting $C_0 = d^2(\mathbf{y}, \mathbf{B}\widehat{\mathbf{x}}_1)$ and keeping in the stack all the paths generated in the first round. This will eventually produce a second point $\widehat{\mathbf{x}}_2$ or declare an empty stack. If a second point is found, we repeat the stack algorithm a third time, by letting $C_0 = d^2(\mathbf{y}, \mathbf{B}\widehat{\mathbf{x}}_2)$ and keeping in the stack all the partial paths after the second round. We continue in this way until the stack is empty, i.e., no more paths in the stack can be extended. It is immediate to see that this sequence of concatenated stack sequential decoding steps is *functionally equivalent* to Algorithm I, in the sense that both algorithms produce the same set of can-

didate ML vectors in the same order.

However, we hasten to say that Algorithm I, as given above, is much more efficient (complexity-wise) than the equivalent sequential decoding formulation, since thanks to the lattice structure of the signal set and the choice of this particular sorting rule, the path at the top of the stack can be predicted at each step without explicitly maintaining and sorting a stack.

Our second sphere decoding algorithm can now be obtained as a sequence of concatenated stack sequential decoding steps, with the following *enhanced* sorting rule:

Sorting rule II. If $\mathcal{I}_{i-1}(\mathbf{x}_i^m) = \emptyset$, the path $\mathbf{x}_i^m$ cannot be extended and should be eliminated from the stack. Consider two paths $\mathbf{x}_i^m$ and $\mathbf{u}_j^m$ such that both $\mathcal{I}_{i-1}(\mathbf{x}_i^m)$ and $\mathcal{I}_{j-1}(\mathbf{u}_j^m)$ are non-empty. Then, $\mathbf{x}_i^m \succeq \mathbf{u}_j^m$ if either $i < j$ or $i = j$ and $\mathcal{M}(\mathbf{x}_i^m) \leq \mathcal{M}(\mathbf{u}_j^m)$. □

Rule II is identical to rule I, except for the fact that equal length paths are sorted according to their accumulated path metric instead of lexicographic ordering. If the stack algorithm based on rule II is applied with $C_0 = \infty$, the resulting $\widehat{\mathbf{x}}$ coincides with the ZF-DFE point given by (14), with the additional constraint that slicing is forced to give a value in $\mathbb{Z}_Q$ for each component (notice the analogy with the Schnorr-Euchner enumeration). Algorithm II can be concisely formulated as follows

Algorithm II (Input C_0', $\mathbf{y}'$, $\mathbf{R}$. Output $\widehat{\mathbf{x}}$):

Step 1. (Initialization) Put the root of the tree in the stack, with the associated metric equal to zero, let $d_c = C_0'$ and $k = 1$.

Step 2. (k-th Stack sequential decoding stage): if the stack is empty, terminate, else expand the path at the top of the stack and order the stack with rule II. If the top path has depth m go to 3, else repeat step 2.

Step 3. (A valid point is found) Let $\mathbf{x}$ denote the depth-m path found. Then, remove it from the stack, let $d_c := \mathcal{M}(\mathbf{x})$, save $\widehat{\mathbf{x}} := \mathbf{x}$, let $k := k + 1$ and go to 2.

□

As for Algorithm I, if no valid point is found, C_0' is increased and the algorithm is restarted.

Again, thanks to the particular structure of the problem and the appropriate choice of the sorting rule, Algorithm II can be implemented in a much more efficient way that does not require maintaining and sorting a stack explicitly. This is given as follows

Algorithm II, smart implementation (Input C_0', $\mathbf{y}'$, $\mathbf{R}$. Output $\widehat{\mathbf{x}}$):

Step 1. (Initialization) Set $i := m$, $T_m := 0$, $\xi_m := 0$, and $d_c := C_0'$ (current sphere squared radius).

Step 2. (DFE on x_i) Set $x_i := \lfloor (y_i' - \xi_i)/r_{i,i} \rceil$ and $\Delta_i := \mathrm{sign}(y_i' - \xi_i - r_{i,i}x_i)$.

Step 3. (Main step) If $d_c < T_i + |y_i' - \xi_i - r_{i,i}x_i|^2$, then go to 4 (i.e., we are outside the sphere).

Else if $x_i \notin [0, Q-1]$ go to 6 (i.e., we are inside the sphere but outside the signal set boundaries).

Else (i.e., we are inside the sphere and signal set boundaries) if $i > 1$, then $\Big\{$ let $\xi_{i-1} := \sum_{j=i}^{m} r_{i-1,j}x_j$, $T_{i-1} := T_i + |y_i' - \xi_i - r_{i,i}x_i|^2$, $i := i - 1$, and go to 2 $\Big\}$.

Else ($i = 1$) go to 5.

Step 4. If $i = m$, terminate, else set $i := i + 1$ and go to 6.

Step 5. (A valid point is found) Let $d_c := T_1 + |y_1' - \xi_1 - r_{1,1}x_1|^2$, save $\widehat{\mathbf{x}} := \mathbf{x}$. Then, let $i := i + 1$ and go to 6.

Step 6. (Schnorr-Euchner enumeration of level i) Let $x_i := x_i + \Delta_i$, $\Delta_i := -\Delta_i - \mathrm{sign}(\Delta_i)$, and go to 3.

□

The main difference with Algorithm I is that given the values of $x_{i+1}, \ldots, x_m$, taking the ZF-DFE on x_i avoids re-testing other nodes at level i in case we fall outside the sphere. Notice also that in the implementation of the algorithm above, the branch metric $|y_i' - \xi_i - r_{i,i}x_i|^2$ needs to be computed only once in steps 3 and 5 (even if it appears twice in step 3 and once in step 5).

By setting $d_c = \infty$, one ensures that the first point found by the algorithm is the ZF-DFE (or the Babai) point (15). This choice, however, may result in some inefficiency if the distance between the ZF-DFE point and the received signal (referred to as the *Babai distance*) is very large. This inefficiency becomes especially significant at very large dimensions, as the algorithm zig-zags its way in the tree from the ZF-DFE point to the ML point. By setting d_c to a finite value C_0', one *informs* the algorithm that the ML solution lies in a sphere of a squared radius C_0', which allows it to retrace its path in the tree if the squared Babai distance is larger than C_0'.

IV. Pre-processing and Ordering

The complexity of sphere decoders depends critically on the pre-processing stage, the ordering in which the components of $\mathbf{x}$ are considered, and the initial choice of C_0. The standard pre-processing and ordering (implicitly assumed in the formulation of Algorithms I and II) consists of the QR decomposition of the channel matrix $\mathbf{B}$ and the natural *back-substitution* component ordering, given by $x_m, x_{m-1}, \ldots, x_1$. However, different pre-processing/ordering approaches may yield a lower expected complexity.

1. **Columns Ordering According to the Euclidean Norm:** In [3], Fincke and Pohst proposed two modifications of the pre-processing stage to further reduce the complexity of the Pohst enumeration for infinite lattices.

(a) Apply the LLL reduction algorithm [35] on the upper triangular matrix $\mathbf{R}$. The LLL algorithm finds a unimodular[1] matrix $\mathbf{U}$ such that $\mathbf{G} = \mathbf{R}\mathbf{U}$ and $\mathbf{G}$ has reduced lattice vectors and almost orthogonal columns.

(b) Order the columns of $\mathbf{G}$ according to their Euclidean norms in a non-decreasing order. Clearly, $\mathbf{G}\mathbf{\Pi}$ where $\mathbf{\Pi}$ is the column ordering permutation matrix, is also a generator matrix for the lattice Λ generated by $\mathbf{R}$.

[1] A square matrix $\mathbf{U}$ is said to be unimodular if it has integer components and its inverse has also integer components or, equivalently, if it has integer components and a determinant equal to ± 1.

This approach is useful for the infinite lattice problem. In fact, (9) can be replaced by

$$\widehat{\mathbf{u}} = \arg \min_{\mathbf{u}\in\mathbb{Z}^m} |\mathbf{y}' - \mathbf{G\Pi u}|^2 \tag{19}$$

where $\mathbf{\Pi}$ is the column ordering permutation matrix. The Pohst, or Schnorr-Euchner enumeration strategies can be applied to solve (19). The point of Λ closets to $\mathbf{y}'$ is given by $\mathbf{G\Pi}\widehat{\mathbf{u}}$.

The utility of this approach for ML detection of finite signal sets is questionable. The reason is that, while $\mathbf{u} = \mathbf{U}^{-1}\mathbf{x}$ has integer components, the image of the hypercube $\mathbb{Z}_Q^m$ under $\mathbf{U}^{-1}$ is not necessarily a hypercube any longer. Therefore, controlling the range of $\mathbf{u}$ is, in general, a very complicated problem. Without the PAM boundary control, there is no guarantee that the output of the algorithm will belong to the transmitted signal set. Therefore, to ensure ML detection, the search may be repeated several times, after excluding the undesired lattice points, which increases the complexity. Overall, our experiments indicate that this modification does not offer complexity reduction in the ML detection problem.

Nevertheless, a useful heuristic consists of ordering the columns of $\mathbf{B}$ according to their Euclidean norms, in a non-decreasing order. This can be explained as follows. Consider the ordering of the components of $\mathbf{x}$ given by the permutation π, i.e., we process the components of $\mathbf{x}$ in the order $x_{\pi(m)}, x_{\pi(m-1)}, \ldots, x_{\pi(1)}$. The recursive upper and lower bounds in the Pohst enumeration (or, equivalently, in Algorithm I) imply that the span of $x_{\pi(i)}$ depends on the spans of $x_{\pi(i+1)}, \ldots, x_{\pi(m)}$ in the same way as the decision on $x_{\pi(i)}$ depends on the decisions on all the previous components in standard decision-feedback equalization. By choosing the permutation π such that $\mathbf{B}' = \mathbf{B\Pi}$ has columns with increasing Euclidean norms (here $\mathbf{\Pi}$ denotes the column permutation matrix corresponding to π), the span of $x_{\pi(m)}$ is reduced with a high probability, so that the expected complexity is reduced. We note that this column ordering is not reported in previous papers on the subject [5], [8], [36], [22], [6]. We have found that this column ordering also decreases the average complexity of Algorithm II.

2. **V-BLAST ZF-DFE Pre-processing and Ordering:**

As a second pre-processing and ordering approach, we propose the V-BLAST *optimal* detection ordering given in [25]. The goal of this ordering is to find the permutation matrix $\mathbf{\Pi}$ such that the QR decomposition of $\mathbf{B}' = \mathbf{B\Pi}$ has the property that $\min_{1\le i\le m} r_{i,i}$ is maximized over all column permutations. The column ordering algorithm is recursive and yields the optimal permutation π in m steps. Let $\mathcal{A}_k$ denote the set of columns indices for the not yet chosen columns. Then, for $k = m, m-1, \ldots, 1$ the algorithm chooses $\pi(k)$ such that

$$\pi(k) = \arg \max_{j\in\mathcal{A}_k} \left\{ \mathbf{b}_j^T \left[\mathbf{I} - \mathbf{B}_{k,j}(\mathbf{B}_{k,j}^T\mathbf{B}_{k,j})^{-1}\mathbf{B}_{k,j}^T \right] \mathbf{b}_j \right\}$$

where $\mathbf{B}_{k,j}$ is the $n \times (k-1)$ matrix formed by the columns $\mathbf{b}_i$ with $i \in \mathcal{A}_k - \{j\}$. The column ordering (equivalently, the ordering of the components of $\mathbf{x}$) is given by $\pi(m), \pi(m-1), \ldots, \pi(1)$. There are two heuristic arguments supporting this pre-processing and ordering approach: 1) in the expressions of the boundaries (16), we see that a large $r_{i,i}$ corresponds to a small interval $\mathcal{I}_i(\mathbf{x}_{i+1}^m)$, therefore, by maximizing the minimum $r_{i,i}$ we attempt to reduce the range of each component in the Pohst enumeration (Algorithm I); 2) the Schnorr-Euchner enumeration with infinite squared radius C_0 yields the ZF-DFE solution as the first point, and hence, the complexity of Algorithm II depends on how close the ZF-DFE point is to the ML point. It has been shown that the V-BLAST detection ordering improves the error probability with ZF-DFE [25]. Therefore, one can argue that the V-BLAST ordering provides a better quality ZF-DFE point, i.e., closer on the average to the ML point.

3. **V-BLAST MMSE-DFE Pre-processing and Ordering:**

In order to further enhance the *quality* of the first point found by Algorithm II, we consider minimum mean-square error (MMSE) instead of ZF filtering. In this MMSE-DFE pre-processing stage, we first translate the observation $\mathbf{y}$ by subtracting the mean signal vector $\beta\sum_{i=1}^m \mathbf{b}_i$, where $\beta = (Q-1)/2$, and consider the variables x_i taking values in the zero-mean signal set $\mathcal{X}_Q = \mathbb{Z}_Q - \beta$. Assume that the components $x_{\pi(m)}, \ldots, x_{\pi(k+1)}$ are known, then the unbiased MMSE estimate of the component $x_{\pi(k)}$ is given by

$$\widetilde{x}_{\pi(k)} = \mathbf{f}_{\pi(k)}^T \left(\mathbf{y} - \sum_{\ell=k+1}^{m} \mathbf{b}_{\pi(\ell)} x_{\pi(\ell)} \right)$$

where

$$\mathbf{f}_{\pi(k)} = \frac{\left[\mathbf{I} + \alpha\sum_{\ell=1}^{k} \mathbf{b}_{\pi(\ell)}\mathbf{b}_{\pi(\ell)}^T\right]^{-1} \mathbf{b}_{\pi(k)}}{\mathbf{b}_{\pi(k)}^T \left[\mathbf{I} + \alpha\sum_{\ell=1}^{k} \mathbf{b}_{\pi(\ell)}\mathbf{b}_{\pi(\ell)}^T\right]^{-1} \mathbf{b}_{\pi(k)}}$$

is the unbiased MMSE filter, and $\alpha = (Q-1)^2/12$ is the variance of the random variables x_i, uniformly distributed over $\mathcal{X}_Q$. The signal to interference plus noise ratio (SINR) for the detection of $x_{\pi(k)}$ from the observation $\widetilde{x}_{\pi(k)}$ is given by

$$\mu_k = \alpha\mathbf{b}_{\pi(k)}^T \left[\mathbf{I} + \alpha\sum_{\ell=1}^{k-1} \mathbf{b}_{\pi(\ell)}\mathbf{b}_{\pi(\ell)}^T \right]^{-1} \mathbf{b}_{\pi(k)}. \tag{20}$$

A very efficient square-root algorithm for computing the permutation π that maximizes $\min_{1\le k\le m} \mu_k$ and, at the same time, providing the corresponding MMSE-DFE filter vectors $\mathbf{f}_{\pi(k)}$ is given in [37]. Letting $\mathbf{F} \triangleq [\mathbf{f}_{\pi(1)}, \mathbf{f}_{\pi(2)}, \ldots, \mathbf{f}_{\pi(m)}]$ denote the MMSE-DFE filter bank for the optimal permutation π and

$$\mathbf{M} \triangleq \text{diag}(\sqrt{\mu_{\pi(1)}}, \ldots, \sqrt{\mu_{\pi(m)}}),$$

the proposed pre-processing and ordering approach now consists of

(a) Computing π, $\mathbf{F}$ and $\mathbf{M}$ by using the method of [37].

(b) Computing the new observation $\boldsymbol{\rho} = \mathbf{MF}^T\mathbf{y}$.

(c) Applying either Algorithm I or Algorithm II, to the new upper triangular (non-equivalent) channel model given by

$$\boldsymbol{\rho} = \mathbf{Gu} + \boldsymbol{\nu} \tag{21}$$

where $\mathbf{G}$ is the upper triangular part of the matrix $\mathbf{MF}^T\mathbf{B\Pi}$ and $\mathbf{\Pi}$ is the column ordering permutation matrix corresponding to the permutation π.

In (21), the noise $\boldsymbol{\nu}$ is non-Gaussian and correlated. However, following the standard approach in equalization, we shall treat it as Gaussian with i.i.d. components $\sim \mathcal{N}(0,1)$. Hence, Algorithms I and II can be modified to find

$$\widehat{\mathbf{u}} = \arg \min_{\mathbf{u}\in\mathcal{X}_Q^m} |\boldsymbol{\rho} - \mathbf{G}\mathbf{u}|^2 \tag{22}$$

by simply replacing $\mathbf{R}$ by $\mathbf{G}$ and $\mathbf{y}'$ by $\boldsymbol{\rho}$.

It is important to notice that, unlike V-BLAST ZF-DFE preprocessing, this approach will not yield ML detection, even if the sphere decoder is initialized by $C_0 = \infty$. However, as shown in the numerical results, the loss in performance, compared to ML detection, is very marginal and the reduction in complexity, compared to other sphere decoders, is often significant. It is also interesting to notice that Algorithm II with the MMSE-DFE pre-processing stage is akin to the standard approach of reduced state sequence estimation by delayed decision feedback (see [38], [39], [40] and references therein).

A remark on the complexity:

The impact of the above pre-processing approaches on the average complexity depends very much on how often pre-processing is performed. In fact, pre-processing depends only on the channel matrix $\mathbf{B}$. If the channel is used repeatedly and $\mathbf{B}$ remains fixed for a long time (a large number of channel uses), the complexity of pre-processing is negligible with respect to the complexity of the sphere decoder search. On the contrary, if $\mathbf{B}$ changes arbitrarily at each channel use, then the pre-processing stage may have a considerable impact on the average complexity.

V. Generalizations and Extensions

A. ML Decoding of More General Codebooks

Although we described Algorithms I and II for symmetric PAM signal sets, they can be easily applied to any hyper-rectangular codebook $\mathcal{C}$

$$\mathcal{C}=\left\{\mathbf{x}\in\mathbb{Z}^m \;:\; \mathbf{x}\in[\mu_{1,\min},\mu_{1,\max}]\times\cdots\times[\mu_{m,\min},\mu_{m,\max}]\right\}$$

by making the boundaries control dependent on the index $i = 1,\ldots,m$. In order to ensure finding the ML solution in Algorithms I and II, one should incorporate the signal set boundary control in the search algorithm. Alternatively, one can ignore the boundaries and run the algorithm to search for the *closest lattice point*, which is then projected (or quantized) by imposing the constellation boundaries only at the end of the algorithm. Such strategy is known in the literature as lattice decoding (as opposed to ML detection or minimum distance decoding [6], [33]). Interestingly, beyond being suboptimal, lattice decoding also results in an increased average complexity when using the original Pohst enumeration [2], [3], the VB implementation [5], or Algorithm I. This is because controlling the boundaries of the intervals inside the search can reduce the range of each variable x_i in the enumeration by excluding many unnecessary points outside the constellation boundaries. On the other hand, this is not always the case for Algorithm II.

Algorithms I and II can be also extended to handle general lattice codes $\mathcal{C}$ (not necessarily hyper-rectangles carved from the integer cubic lattice). Consider a lattice $\Lambda \subset \mathbb{R}^m$ with generator matrix $\mathbf{G}$, i.e., $\Lambda = \{\mathbf{G}\mathbf{x} \;:\; \mathbf{x}\in\mathbb{Z}^m\}$. A lattice code (or constellation) $\mathcal{C}$ is the set of lattice points $\mathbf{c}\in\Lambda\cap\mathcal{R}$, where $\mathcal{R}$ is some shaping region [41], [42]. Typically, the shaping region $\mathcal{R}$ is chosen to be the m-dimensional sphere of given squared radius R^2. In order to handle this case, we first restrict $\mathbf{x}$ to belong to the hyper-rectangular region $\mathcal{X} = [\mu_{1,\min},\mu_{1,\max}]\times\cdots\times[\mu_{m,\min},\mu_{m,\max}]$. The difference with respect to the previous case is that now there exist points $\mathbf{x}\in\mathcal{X}$ such that $\mathbf{G}\mathbf{x}\notin\mathcal{C}$, i.e., $|\mathbf{G}\mathbf{x}|^2 > R^2$, therefore, the *closest lattice point* found by the sphere decoder may not be a valid code word. To overcome this difficulty we modify Algorithms I (or equivalently Algorithm II) as follows. The algorithm now takes R^2 and $\mathbf{G}$ as additional inputs. When a lattice point is found, one tests whether it is a valid point by testing $|\mathbf{G}\mathbf{x}|^2 \le R^2$: if yes, then the algorithm saves $\widehat{\mathbf{x}} := \mathbf{x}$, updates the boundaries and goes on to examine the next point. Otherwise, it moves on to the next point without updating the boundaries and without saving the point found. At the end of the search, if the algorithm has found a valid point $\widehat{\mathbf{x}}$ (i.e., such that $\mathbf{G}\widehat{\mathbf{x}}\in\mathcal{C}$), then this is the ML point, otherwise, one increases the initial squared sphere radius C_0 and restarts the search.

B. More Sources than Sensors

The proposed algorithms can be generalized to handle the case $m > n$ by following the approach of [36]. The main idea is to partition $\mathbf{x}$ into two vectors $\mathbf{x}^1$ with n elements and $\mathbf{x}^2$ with $p \triangleq m-n$ elements. This partitioning induces a similar partition on the matrix $\mathbf{B}$, i.e., $\mathbf{B} = [\mathbf{B}_1,\,\mathbf{B}_2]$. Assuming that $\mathrm{rank}(\mathbf{B}_1) = n$, the QR decomposition applied to $\mathbf{B}_1$ (with possibly ordered columns) yields the equivalent model

$$\begin{aligned}\mathbf{y}' &\triangleq \mathbf{Q}^T\mathbf{y} = \mathbf{R}_1\mathbf{x}^1 + \mathbf{R}_2\mathbf{x}^2 + \mathbf{Q}^T\mathbf{z} \\ &= \begin{pmatrix} r_{1,1}^{(1)} & r_{1,2}^{(1)} & \cdots & r_{1,n}^{(1)} \\ 0 & r_{2,2}^{(1)} & \cdots & r_{2,n}^{(1)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{n,n}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\ &+ \begin{pmatrix} r_{1,1}^{(2)} & r_{1,2}^{(2)} & \cdots & r_{1,p}^{(2)} \\ r_{2,1}^{(2)} & r_{2,2}^{(2)} & \cdots & r_{2,p}^{(2)} \\ \vdots & \ddots & \ddots & \vdots \\ r_{n,1}^{(2)} & r_{n,2}^{(2)} & \cdots & r_{n,p}^{(2)} \end{pmatrix} \begin{pmatrix} x_{n+1} \\ x_{n+2} \\ \vdots \\ x_m \end{pmatrix} + \mathbf{Q}^T\mathbf{z}\end{aligned} \tag{23}$$

where the diagonal elements $\{r_{i,i}^{(1)}\}$ are positive, $\mathbf{Q}^T\mathbf{z}$ has the same statistics of $\mathbf{z}$, and $\mathbf{R}_2 = \mathbf{Q}^T\mathbf{B}_2$. Note that (23) is an under-determined system (more unknowns than equations) but it may still admit a unique solution since $\mathbf{x}$ is constrained to belong to a finite and discrete set.

In order to apply Algorithms I and II (in general, any sphere decoder) to the case of (23), we fix the components of $\mathbf{x}^2$ and

use our algorithms to solve

$$\widehat{\mathbf{x}}^1 = \arg \min_{\mathbf{x}^1 \in \mathcal{X}} \left| \mathbf{y}' - \mathbf{R}_2 \mathbf{x}^2 - \mathbf{R}_1 \mathbf{x}^1 \right|^2 \tag{24}$$

where $\mathcal{X}$ is an appropriate n-dimensional hyper-rectangular region (e.g., in the case of PAM signals we have $\mathcal{X} = \mathbb{Z}_Q^n$). This can be repeated for every choice of $\mathbf{x}^2$, and the ML point is eventually found.

We observe that this approach has an exponential complexity in $m - n$ independent of the signal-to-noise ratio (SNR), which seems to be inherent to the problem itself. Nonetheless, the idea of generalizing the sphere decoder to $m > n$ has an additional subtle advantage: once a valid point is found with distance d_c, all values of $\mathbf{x}^2$ such that $|\mathbf{y}' - \mathbf{R}_2 \mathbf{x}^2|^2 > d_c$ can be discarded from the beginning. Their exclusion from the search costs only the computation of the squared distance $|\mathbf{y}' - \mathbf{R}_2 \mathbf{x}^2|^2$ instead of computing the distance of the whole point.

C. Complex and Algebraic Pohst Enumeration

In [28], a complex sphere decoder is proposed for ML detection in MIMO channels without passing to the real representation as in [8]. In this approach, the QR decomposition is performed on the complex channel matrix $\mathbf{H}$ and the Pohst strategy is applied by making successive bounds on the complex components x_i, where the search is done over a disc in $\mathbb{C}$ of a given radius. This is made possible because one can still *count* the complex integers (Gaussian integers) in $\mathbb{C}$. The complex sphere decoder is especially useful when using PSK signal sets, which lie on a circle, since the intersection between the disk and the PSK circle can be easily characterized [28]. This complex sphere decoder can be considered as a special case of the generalization of the Pohst strategy for lattices over number fields [43], [44]. In fact, one can perform the Pohst enumeration over any set of algebraic integers:

$$\{x_i = u_{i1} + u_{i2}\theta + \ldots + u_{iq}\theta^{q-1}, \\ u_{i,j} \in \mathbb{Z}, i = 1, \ldots, m, j = 1, \ldots, q\}$$

where θ is algebraic of degree $\geq q$ over $\mathbb{Q}$ [43], [44]. Also in this case we make successive lower and upper bounds on $|x_i - \xi_i|$, $i = m, \ldots, 1$, with the same setup as in the original Pohst strategy described in Section II [43], [44].

In order to choose a point x_i in the allowed "interval" (i.e., satisfying the boundaries) the coefficients $u_{i1}, \ldots, u_{iq}$ of the representation of x_i in the basis $\{1, \theta, \ldots, \theta^{q-1}\}$ must be determined. These can be found again by applying the same Pohst strategy [43], [44]. Because of the latter enumeration for each $i = m, \ldots, 1$, the algorithm complexity depends on the product mq. We see now that, although the complex ($q = 2, \theta = \sqrt{-1}$) Pohst strategy does not expand the lattice dimension, it has complexity comparable with the real algorithm of double dimension because the complex enumeration costs twice as much as the real one, and complex operations cost at least twice as the real ones [43]. Nevertheless, in some cases it may be useful to keep the original problem, when the lattice generator matrix $\mathbf{H}$ has some desirable properties (e.g., it is upper triangular or band-limited as in ISI or multi-path complex channels [18]) or the signal set where the components of $\mathbf{x}$ take on values has some nice feature (e.g., it is a PSK signal sets). It is easy to see that Algorithms I and II extend naturally to these scenarios.

VI. Simulation Results

In this section we compare the proposed algorithms with the basic Pohst enumeration [2], [3] (see also the beginning of Section III) and the VB sphere decoder [5]. Our experimental setup corresponds to the transmission of multi-dimensional square QAM constellations over a multi-antenna flat Rayleigh fading channel. We consider M transmit and N receive antennas and assume that the $N \times M$ channel matrix $\mathbf{H}$ remains fixed during $T = 100$ symbols and then changes randomly. Unless stated otherwise, we perform ZF or MMSE-DFE V-BLAST optimal ordering of the columns of matrix $\mathbf{B}$. Following in the footsteps of [3], [22], we use the number of flops as a measure for complexity and we plot the average *complexity exponent* defined as [21], [22]

$$\log_m(\text{average number of flops}).$$

We only count the flops of the search algorithm without accounting for the cost of the pre-processing stage. In all the simulations, at least 10000 channel realizations are generated. We implemented all the algorithms in floating point C (for previously reported algorithms, we followed closely the published flowcharts), and the programs are invoked inside MATLAB 6 by using mex files.

We follow an ad-hoc method to initialize the decoders by starting with a small C_0, generally determined by trial and error depending on the system parameters [4], [8], [6], and increasing it gradually by steps of C_0 until a point is found. In general, the proper initial choice of C_0 is critical in order to minimize the complexity of the decoders; a too small C_0 may result in an empty sphere, whereas a too large C_0 may result in too many points to be enumerated. We expect the natural spanning algorithms (i.e., original Pohst, VB implementation, and Algorithm I) to be more sensitive to a large radius initialization than Algorithm II. This is because the first point found by Algorithm II is the ZF-DFE (or MMSE-DFE) solution, and the algorithm is not sensitive to increasing C_0 beyond the corresponding squared Babai distance.

Fig. 2 shows the average complexity of Algorithm I as a function of C_0 when decoding a 4×4 MIMO system with 64-QAM symbols (corresponding to a spectral efficiency of 24 bit/s/Hz) at different SNR's. It is seen that the "optimal" value of C_0 (i.e., the value achieving minimum complexity) decreases as a function of SNR. For medium to large SNR's (i.e., 12 to 22 dB in this scenario) the "optimal" values of C_0 are small, and the average complexity of Algorithm I increases considerably by increasing C_0.

Algorithm II is sensitive to the initial squared radius C_0 at larger dimensions, as demonstrated in Fig. 3 where we compare the average complexity of Algorithm II with ZF-DFE and MMSE-DFE pre-processing over a 16×16 MIMO system with a 16-QAM constellation at an SNR of 20 dB. We observe that MMSE-DFE pre-processing makes Algorithm II more robust to the initial C_0 than ZF-DFE pre-processing. This matches the in-

tuition that with MMSE-DFE the Babai distance is better, with high probability, than with ZF-DFE. The optimal choice of C_0 for both algorithms is typically larger than that of Algorithm I. As the dimension increases, even Algorithm II with MMSE-DFE pre-processing becomes quite sensitive to the initial radius choice, as shown in Fig. 4 for a 20×20 MIMO system with a 16-QAM constellation at different SNR's.

In the rest of this section, we report the average complexity of all the algorithms when initialized by the corresponding "optimal" ad-hoc values for C_0.

Fig. 5 compares the average complexity of the four algorithms for $M = N = 4$ and a 64-QAM constellation at different SNR's, where both ZF-DEF and MMSE-DFE pre-processing stages are used with Algorithm II. Notice that, under the same initialization conditions, Algorithm I is the most efficient among the natural spanning algorithms as predicted by Proposition 1, and the VB implementation is the least efficient. Interestingly, the basic Pohst enumeration is more efficient than the VB implementation in certain scenarios, since one does not repeat previously spanned point-segments when counting all the points inside the sphere. This observation is only true for small dimensions and with small C_0; for large dimensions and/or large C_0, the VB implementation is more efficient than the basic Pohst enumeration. Algorithm II, with ZF-DFE or MMSE-DFE, is more efficient than the natural spanning algorithms, all under optimized initializations. Algorithm II with MMSE-DFE is the most efficient algorithm. It is about 70 times faster than Algorithm I at small SNR's, and about 2 times faster at large SNR's. Remarkably, this significant complexity reduction costs only a very marginal loss in performance as shown in Fig. 6. For the sake of comparison, in this figure we also show the performance of the MMSE linear detector, i.e., a linear MMSE filter followed by symbol-by-symbol hard decisions [16]. The advantage of ML (or near ML) with respect to linear detection methods is evident.

The gain in complexity reduction offered by Algorithm II, with MMSE-DFE, further increases as the lattice dimension increases as reported in Fig. 7, where Algorithms I and II are compared over an $M \times M$ MIMO system with a 4-QAM constellation at SNR's of 10 and 20 dB. For example, Algorithm II with MMSE-DFE is about 40 times faster than Algorithm I for $M = 64$ at an SNR of 20 dB. Again this complexity reduction comes at almost no cost in performance.

In all the previous figures, ZF-DFE or MMSE-DFE V-BLAST optimal ordering of the generator matrix columns was adopted. In Fig. 8 we show the effectiveness of V-BLAST ordering by comparing it with the natural ordering (the columns of $\mathbf{B}$ are processed in the natural order) and the ordering of the columns according to increasing Euclidean norm for an $M \times M$ MIMO system with a 4-QAM constellation at an SNR of 20 dB. In this figure, we plot the average number of flops instead of the average complexity exponents in order to better visualize the complexity reduction factor. We observe that, as m increases, the advantage of the V-BLAST column ordering becomes more significant. For example, at $m = 128$, the V-BLAST ordering gives a complexity reduction factor of about 13 times over the column ordering based on the Euclidean norm. This complexity reduction can be attributed to the improved *quality* of the first point found by Algorithm II.

In summary, driven by extensive numerical evidence[2] we conclude that Algorithm II, with V-BLAST MMSE-DFE preprocessing and ordering, offers a very attractive implementation of the sphere decoder with finite alphabet constellations. This algorithm yields almost ML performance at a polynomial (often between $O(m^2)$ and $O(m^3)$) average complexity in the problem dimension over a wide range of system parameters, and consistently outperforms all previously known sphere decoders.

VII. Concluding Remarks

In this paper we have investigated reduced complexity methods for ML detection of multi-dimensional constellations and lattice codes, based on the Pohst enumeration strategy [2]. Two efficient algorithms were proposed. Algorithm I is directly inspired by the Pohst enumeration strategy and was shown to be more efficient than the Viterbo-Boutros sphere decoder. Algorithm II is inspired by the Schnorr-Euchner enumeration strategy and is more robust than the Pohst-based algorithms with respect to the initial choice of the sphere radius. Both algorithms have been shown to be functionally equivalent to a concatenated sequence of the stack sequential decoding algorithm with appropriate path metrics and stack sorting rules.

By combining Algorithm II with an efficient pre-processing stage, we obtained a near ML decoding algorithm that uniformly outperforms all known sphere decoders in terms of receiver complexity. Furthermore, we have discussed some generalizations of the proposed algorithms to the under-determined case ($n < m$) and to more general codebooks, such as lattice codes and complex signal sets.

We would like to conclude by pointing out that the connection between sphere decoding and sequential decoding established in this paper may have wider implications than those exploited in our work. One would expect the cross fertilization between the two areas to yield more efficient decoding algorithms, and to allow for a better understanding of their fundamental limits.

References

[1] J. Proakis, *Digital communications, 4th Ed.*, New York: McGraw-Hill, 2000.

[2] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications," in *ACM SIGSAM*, Vol. 15, pp. 37–44, Bull, 1981.

[3] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, Vol. 44, pp. 463–471, Apr. 1985.

[4] E. Viterbo and E. Biglieri, "A universal lattice decoder," in *GRETSI 14-ème Colloque*, Juan-les-Pins, Sept. 1993.

[5] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channel," *IEEE Trans. Inform. Theory*, Vol. 45, pp. 1639–1642, July 1999.

[6] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, Vol. 48, pp. 2201-2214, Aug. 2002.

[7] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Mathematical Programming*, Vol. 66, pp. 181-191, 1994.

[8] M. O. Damen, A. Chkeif, and J.-C. Belfi ore, "Lattice codes decoder for space-time codes," *IEEE Commun. Lett.*, Vol. 4, pp. 161–163, May 2000.

[2]The results reported here are, for the sake of space limitation, a small subset of our simulations.

[9] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. on Telecommun. ETT*, Vol. 10, No. 6, pp. 585–596, November 1999.
[10] G. Foschini and M. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless personal communications*, Vol. 6, No. 3, pp. 311–335, March 1998.
[11] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inform. Theory,* Vol. 48, No. 7, pp. 1804 -1824, Jully 2002.
[12] H. El Gamal and M. O. Damen, "Universal space-time coding," *IEEE Trans. Inform. Theory*, Vol. 49, No. 5, pp.1097-1119, May 2003.
[13] M. O. Damen, H. El Gamal, and N. C. Beaulieu, "Linear threaded algebraic space-time constellations," to appear in *IEEE Trans. Inform. Theory* 2003.
[14] J. Boutros and E. Viterbo, "Signal space diversity: a power- and bandwidth-efficient diversity technique for the Rayleigh fading channel," *IEEE Trans. Inform. Theory,* Vol. 44, No. 4, pp. 1453 -1467, July 1998.
[15] M. Debbah, P. Loubaton, and M. De Courville, "Linear precoded OFDM transmissions with MMSE equalization: facts and results," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, FL. May, 2002.
[16] S. Verdu, *Multiuser Detection*, Cambridge, UK: Cambridge University Press, 1998.
[17] L. Brunel and J. Boutros, "Euclidean space lattice decoding for joint detection in CDMA systems," in *Proc. IEEE Inform. Theory Workshop*, Kruger National Park, South Africa, June 1999.
[18] M. O. Damen, K. A. Meraim, and A. Safavi, "On CDMA with space-time codes over multi-path fading channels," in *IEEE Trans. on Wireless Commun.*, Vol. 2, pp. 11-19, Jan. 2003.
[19] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," in *Proc. 30-th Annual ACM Symp. Theory of Computing*, Dallas, TX., May, 1998.
[20] D. Marcciancio, "The shortest vector in a lattice is hard to approximate to within some constant," in *Proc. 39-th Annual Symp. Found. Computer Science*, Palo Alto, CA., Nov. 1998.
[21] H. Vikalo and B. Hassibi, "The expected complexity of sphere decoding, Part I: Theory, Part II: Applications," submitted to *IEEE Trans. on Signal Processing*, 2002.
[22] M. O. Damen, K. Abed-Meraim, and M. S. Lemdani, "Further results on the sphere decoder," in *Proc. IEEE International Symposium on Inform. Theory*, p. 333, Washington, D.C., June 2001.
[23] G. Rekaya and J.-C. Belfiore, "Complexity of ML lattice decoders for the decoding of linear full rate space-time codes," submitted to *IEEE Trans. Wireless Commun.*, Dec. 2002.
[24] C. Windpassinger and R. Fischer, "Low-complexity near maximum likelihood detection and precoding for MIMO systems using lattice reduction," in *Proc. IEEE Inform. Theory Workshop*, pp. 345-348, Paris, France, April 2003.
[25] J. Foschini, G. Golden, R. Valenzuela, and P. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commun.*, Vol. 17, No. 11, pp. 1841–1852, Nov. 1999.
[26] M. O. Damen, K. Abed-Meraim, and S. Burykh, "Iterative QR detection for BLAST," *Wireless Personal Communications*, Vol. 19, pp. 179–192, Dec. 2001.
[27] S. Baro, J. Hagenauer and M. Witzke, "Iterative detection of MIMO transmission using a list sequential (LISS) detector," in *Proc. IEEE International Conference on Commun.*, Anchorage, Alaska, May 2003.
[28] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Commun.*, Vol. 51, pp. 389-399, March 2003.
[29] J. Boutros, N. Gresset, L. Brunel and M. Fossorier, "Soft-input soft-output lattice sphere decoder for linear channels," submitted to *IEEE Global Communications Conference*, March 2003.
[30] H. Vikalo and B. Hassibi, "Iterative decoding for MIMO channels via modified sphere decoder," submitted to *IEEE Transactions on Wireless Communications*, 2002.
[31] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for multi-antenna modulation and detection," submitted to *IEEE Trans. on Commun.*, June 2002.
[32] A. Viterbi and J. Omura, *Principles of Digital Communication and Coding*, New York: McGraw-Hill, 1979.
[33] J.H. Conway and N.J. Sloane, *Sphere Packing, Lattices and Groups, 3rd ed.* New York: Springer-Verlag 1998.
[34] R. Gallager, *Information Theory and Reliable Communication*, New York: Wiley, 1968.
[35] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, Vol. 261, pp. 515–534, 1982.
[36] M. O. Damen, K. Abed-Meraim, and J.-C. Belfiore, "Generalized sphere decoder for asymmetrical space-time communication architecture," *Electron. Lett.*, Vol. 36, p. 166, Jan. 2000.
[37] B. Hassibi, "An efficient square-root algorithm for BLAST," submitted to *IEEE Trans. on Signal Processing*, 1999. Available on line: "http://mars.bell-labs.com".
[38] S. Shamai and R. Laroia, "The intersymbol interference channel: lower bounds on capacity and channel precoding loss," *IEEE Trans. Inform. Theory*, Vol. 42, pp. 1388–1404, Sept. 1996.
[39] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. on Commun.*, Vol. 37, pp. 428–436, May 1989.
[40] G. Caire and G. Colavolpe, "On space-time coding for quasi-static multiple-antenna channels," *IEEE Trans. Inform. Theory*, Vol. 49, pp.1400-1416 June 2003.
[41] G. D. Forney, Jr., "Coset codes-part I: Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, Vol. 34, pp. 1123-1151, Sept. 1988.
[42] G. D. Forney, Jr., "Coset codes-part II: Binary lattice and related codes," *IEEE Trans. Inform. Theory*, Vol. 34, pp. 1151-1186, Sept. 1988.
[43] C. Fieker, A. Jurk, M. Pohst, "On solving relative norm equations in algebraic number fields," Math. Comp. Vol. 66, pp. 399-410, 1997.
[44] C. Fieker and M. Pohst, "On lattices over number fields," manuscript.

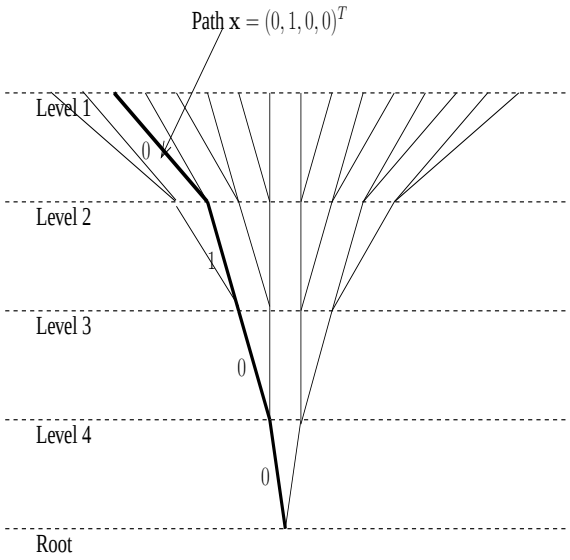

Fig. 1. Tree representation of the paths searched by the sphere decoder in the case $m = 4$ and $Q = 2$. A particular path is evidenced as an example.

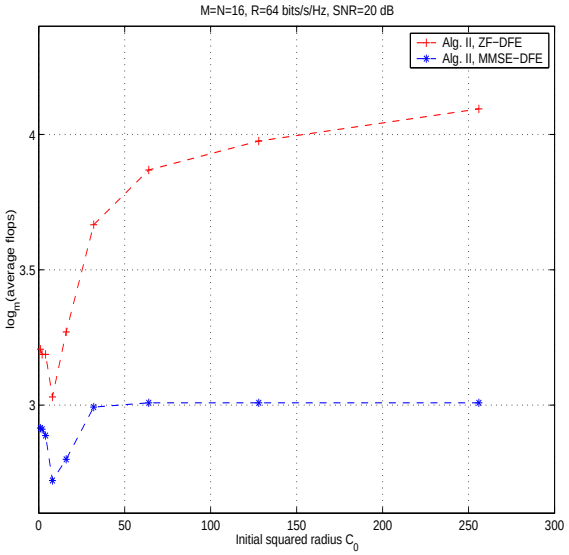

Fig. 3. Initialization: the average complexity of Algorithm II (ZF and MMSE-DFE) as a function of the initial squared radius in an uncoded 16×16 system with a 16-QAM constellation.

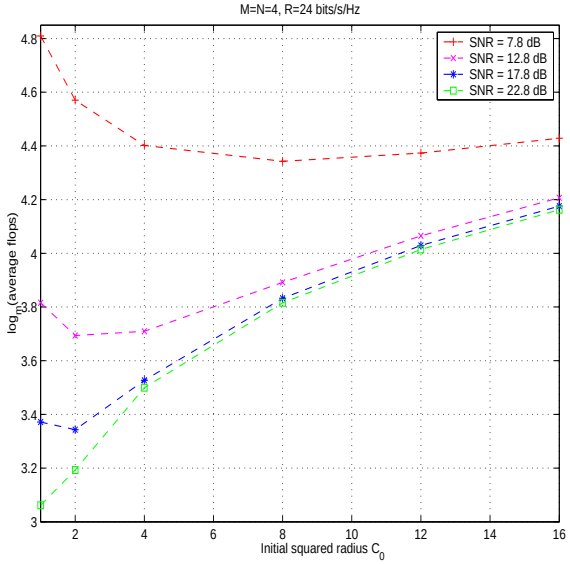

Fig. 2. Initialization: the average complexity of Algorithm I as a function of the initial squared radius in an uncoded 4×4 system with a 64-QAM constellation.

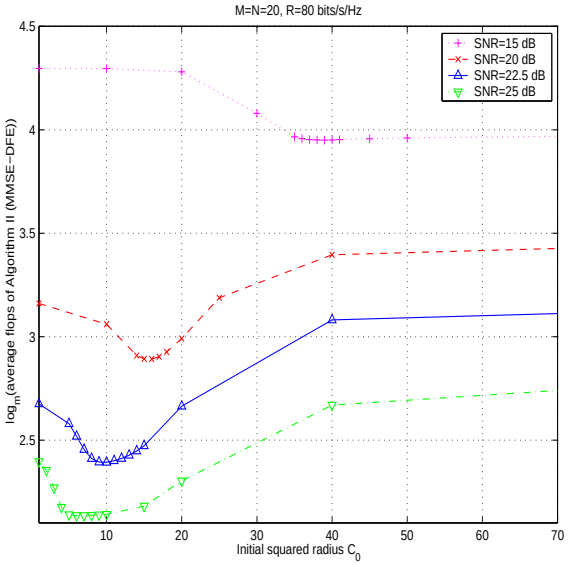

Fig. 4. Initialization: the average complexity of Algorithm II (MMSE-DFE) as a function of the initial squared radius in an uncoded 20×20 system with a 16-QAM constellation.

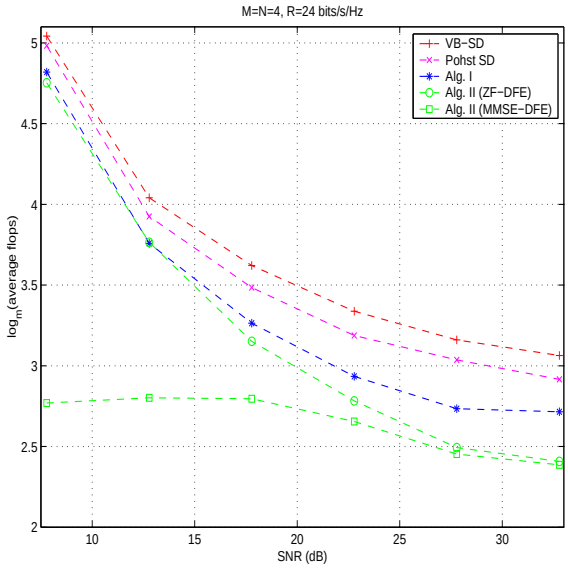

Fig. 5. The average complexity of different sphere decoders in an uncoded 4×4 system with a 64-QAM constellation.

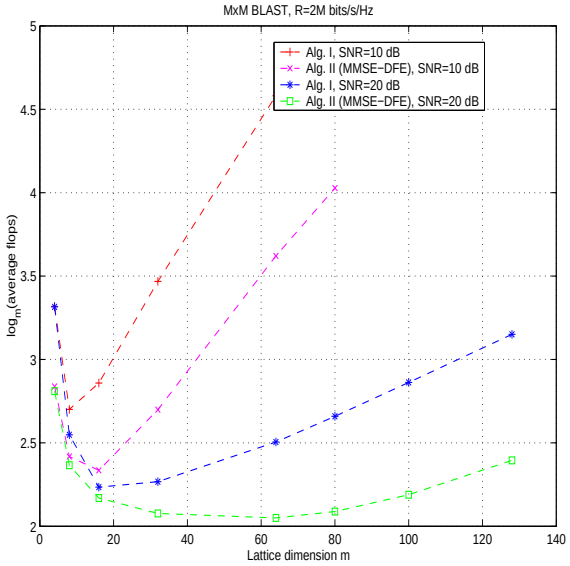

Fig. 7. The complexity of Algorithm II (MMSE-DFE) and Algorithm I in an $M \times M$ uncoded system ($m = 2M$) with a 4-QAM constellation.

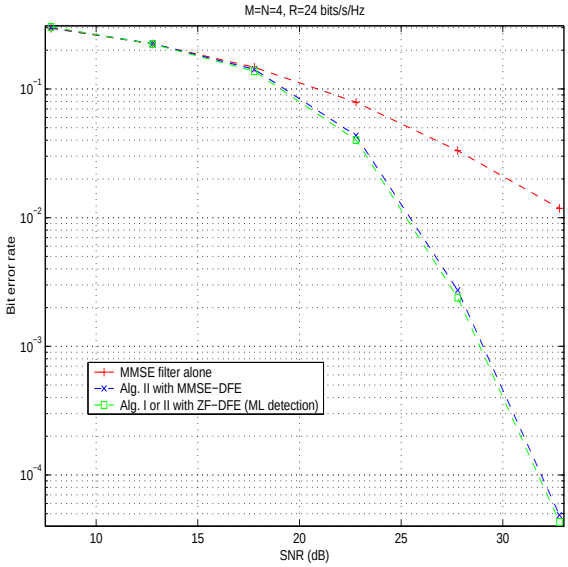

Fig. 6. The performance of Algorithm II (MMSE-DFE), the MMSE fi ltering, and ML detection (Algorithm I or II with ZF) in an uncoded 4×4 system with a 64-QAM constellation.

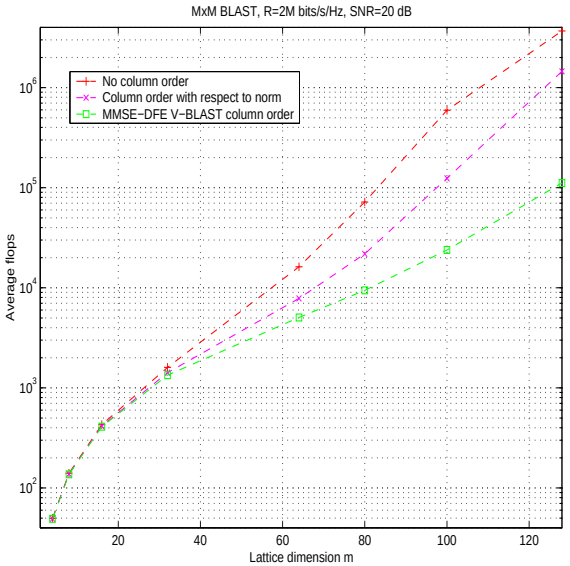

Fig. 8. The effect of column ordering on Algorithm II in an $M \times M$ uncoded system ($m = 2M$) with a 4-QAM constellation.