A Framework for Fixed Complexity Breadth-First MIMO Detection

(Invited Paper)

David L. Milliner*, Ernesto Zimmermann†, John R. Barry* and Gerhard Fettweis†

*School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332–0250
Email: {dlm,barry}@ece.gatech.edu

†Vodafone Chair Mobile Communications Systems
Technische Universität Dresden, D-01062 Dresden, Germany
Email: {zimmere,fettweis}@ifn.et.tu-dresden.de

***Abstract*—In this work we present a unifying framework to characterize different types of breadth-first tree search detectors for multiple antenna systems. All considered algorithms operate at fixed complexity and require only a single pass through the detection tree, making them very attractive for practical implementation. Existing approaches are placed into this framework and a performance-complexity analysis is performed for both hard-output and soft-output detection. The B-Chase and the parallel smart candidate adding algorithm emerge as the most attractive schemes for hard-output and soft-output detection, respectively.**

I. INTRODUCTION

Radio frequency spectrum is a scarce and thus valuable resource. Equipping the transmitter and receiver of a wireless system with multiple antennas allows for efficient use of this resource by increasing spectral efficiency. Unfortunately, this improvement is accompanied by potentially dramatic increases in detection complexity (exponential in the worst case). Numerous approaches have been proposed for solving the detection problem in such multiple-input multiple-output (MIMO) systems, for hard-output as well as soft-output detection. Many of these schemes reformulate the MIMO detection task into a tree search problem, in which leaf nodes maximizing a certain metric have to be found in a detection tree.

A class of tree search schemes which has received considerable attention, justified by their attractive performance-complexity tradeoff and amenability for practical implementation, are breadth-first algorithms. Breadth-first schemes search the detection tree layer-by-layer and are therefore easily constructed to have fixed complexity. However, one inherent drawback of these schemes is that their achievable performance is often limited by error propagation, particularly when the available computational resources are small.

In [1] the general "Chase detection" strategy was proposed for MIMO channels, reducing a variety of previously reported schemes to special cases of a larger framework. However, the framework focuses on the case of hard-output detection, leaving some aspects aside which are relevant for soft-output detection.

In this paper we present a general framework for fixed complexity breadth-first MIMO detection schemes through a vector parameterization approach which includes the preprocessing specification. It covers both hard-output *and* soft-output detection and is thus in some ways less restrictive, and in other ways more restricted, than the one from [1]. This framework enables a performance-complexity comparison of the most popular breadth-first schemes.

The remainder of this paper is organized as follows: after discussing the employed system model in Section II, Section III details fundamentals of MIMO detection. Section IV describes our framework for fixed complexity MIMO detection and the placement of existing approaches within this framework. In Section V we provide a performance-complexity comparison of existing fixed complexity breadth-first MIMO detection approaches in both hard-output and soft-output systems. Finally, conclusions are drawn in Section VI.

II. SYSTEM MODEL

Consider an $N_T \times N_R$ MIMO system based on a BICM transmit strategy as depicted in Fig. 1: the vector $\mathbf{u}$ of i.i.d. information bits is encoded and interleaved. The resulting coded bit stream is partitioned into blocks $\mathbf{c}$ of $N_T \cdot L$ bits and mapped onto a vector symbol $\mathbf{x} \in \mathcal{X}$ whose components are taken from some complex constellation $\mathcal{C}$ (e.g. Gray mapped 64-QAM). Here, L denotes the number of bits per complex symbol, resulting in $Q = |\mathcal{C}| = 2^L$ different constellation points. We consider transmission over a flat fading channel.

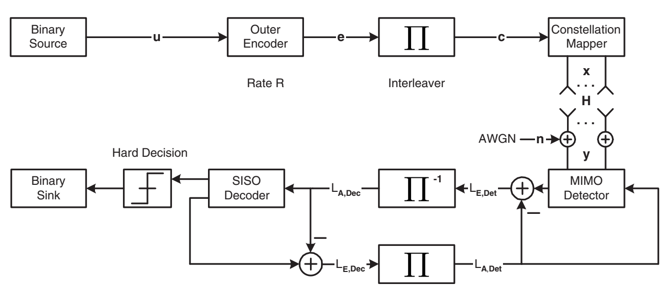

Figure 1. System model using a BICM transmit strategy.

In the equivalent discrete-time base-band model, the received signal $\mathbf{y}$ is thus given by:

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

where $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ is the channel transfer matrix which is assumed to be perfectly known at the receiver. The entries of $\mathbf{H}$ are realizations of zero mean i.i.d. complex Gaussian random processes of unit variance (passive subchannels). The average transmit energy is normalized such that $\mathcal{E}\{\mathbf{x}\mathbf{x}^{\mathrm{H}}\} = E_s/N_T\ \mathbf{I}$. The vector $\mathbf{n} \in \mathbb{C}^{N_R \times 1}$ represents the receiver noise whose components are zero mean i.i.d. complex Gaussian random variables with variance $N_0/2$ per real dimension: $\mathcal{E}\{\mathbf{n}\mathbf{n}^{\mathrm{H}}\} = N_0\ \mathbf{I}$. The signal-to-noise ratio (SNR) at each receive antenna is hence given by $\mathrm{SNR} = E_s/N_0$.

III. Fundamentals

In the case of soft-output detection, the detector in Fig. 1 has to generate reliability information, i.e., determine the a posteriori probability for each of the code bits $c_{m,l}$ in $\mathbf{x}$, where $m \in \{1, \dots, N_T\}$ is the symbol index, and $l \in \{1, \dots, L\}$ the bit index in the m-th symbol. Since we are dealing with binary numbers, this information is conveniently expressed in the form of log-likelihood ratios (LLRs):

$$\begin{aligned} L(c_{m,l}|\mathbf{y}) &:= \ln \frac{P\left[c_{m,l} = +1|\mathbf{y}\right]}{P\left[c_{m,l} = -1|\mathbf{y}\right]} \qquad (2) \\ &\approx \max_{\hat{\mathbf{x}} \in \mathcal{X}_{m,l}^{+1}} \left\{ \frac{-\left\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\right\|^2}{N_0} + \sum_{i=1}^{N_T} \sum_{k=1}^{L} \ln P\left(c_{i,k} = \hat{c}_{i,k}\right) \right\} \\ &- \max_{\hat{\mathbf{x}} \in \mathcal{X}_{m,l}^{-1}} \left\{ \frac{-\left\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\right\|^2}{N_0} + \sum_{i=1}^{N_T} \sum_{k=1}^{L} \ln P\left(c_{i,k} = \hat{c}_{i,k}\right) \right\}, \end{aligned}$$

where the second line follows from the application of the max-log approximation. Here, $\mathcal{X}_{m,l}^{\pm 1}$ denotes the set of $2^{N_T \cdot L - 1}$ symbols $\hat{\mathbf{x}} \in \mathcal{X}$ for which $\hat{c}_{m,l} = \pm 1$, where $\hat{\mathbf{x}}$ denotes a certain hypothesis on the transmit sequence and $\hat{\mathbf{c}}$ the corresponding vector of code bits. Evaluating (2) by a brute-force approach (maxLogAPP detection) is well known to require an effort growing exponentially in the number of transmitted bits per vector symbol. However, only a few hypotheses in $\mathcal{X}_{m,l}^{\pm 1}$ actually maximize each of the respective terms in (2). Several close-to-optimal detection strategies therefore construct a subset list $\mathcal{L} \subset \mathcal{X}$ of size $|\mathcal{L}|$ from which the LLRs are determined. The list $\mathcal{L}$ should on the one hand include only a fraction of the elements from $\mathcal{X}$, to minimize complexity. On the other hand, it should be large enough such that the true detector LLRs can be approximated as closely as possible, to maximize performance. For hard-output detection the detector "only" needs to determine the most likely transmitted signal vector $\hat{\mathbf{x}}_{\mathrm{JML}}$ (and corresponding $\hat{\mathbf{c}}_{\mathrm{JML}}$).

A. Tree-Search MIMO Detection

Tree search based MIMO detection techniques construct $\mathcal{L}$ using a back-substitution approach. After an orthogonal-triangular decomposition of the channel matrix, e.g. $\mathbf{H} = \mathbf{Q}\mathbf{R}$, the LLRs can be determined using the per-antenna metric increments Λ_i:

$$L(c_{m,l}|\mathbf{y}) \approx \max_{\hat{\mathbf{x}} \in \mathcal{L} \cap \mathcal{X}_{m,l}^{+1}} \left\{ \sum_{i=1}^{N_T} \Lambda_i \right\} - \max_{\hat{\mathbf{x}} \in \mathcal{L} \cap \mathcal{X}_{m,l}^{-1}} \left\{ \dots \right\}$$

which are referred to as *branch metrics* and given by

$$\Lambda_i = -\frac{1}{N_0} \left| \tilde{y}_i - \sum_{j=i}^{N_T} r_{i,j} \hat{x}_j \right|^2 + \sum_{k=1}^{L} \ln P\left(c_{i,k} = \hat{c}_{i,k}\right) \qquad (3)$$

with $\tilde{\mathbf{y}} = \mathbf{Q}^{\mathrm{H}}\mathbf{y}$. The detector starts in layer $i = N_T$ and works its way up until layer $i = 1$ is reached. For each branch in the tree, Q different choices are possible for the signal estimate x_i. The detection process can hence be interpreted as a search for leaf nodes in a tree structure. Different types of tree search based detectors can be implemented by using the *path metrics* $\sum_{j=i}^{N_T} \Lambda_j$ (with i as the current layer index) to control which tree nodes are added to the working stack and in which order.

A major problem for all "conventional" tree search schemes are missing counter-hypotheses: whenever $\mathcal{L} \cap \mathcal{X}_{m,l}^{\pm 1} = \emptyset$, the magnitude of the LLR for the corresponding bit cannot be determined from the entries of $\mathcal{L}$. One solution to this problem is to use a technique called smart candidate adding [2]–[6]. This strategy is based on the recognition that the LLRs as defined by (2) can also be written as the difference between the metric of the *MAP estimate* $\hat{\mathbf{x}}^{\mathrm{MAP}}$ (i.e., the hypothesis maximizing the a posteriori probability) and the metric of the best counter-hypothesis for each bit:

$$\begin{aligned} L(c_{m,l}|\mathbf{y}) \approx \hat{c}_{m,l}^{\mathrm{MAP}} \Bigg(&\sum_{i=1}^{N_T} \Lambda_i \left(\hat{\mathbf{x}}^{\mathrm{MAP}}\right) \quad \dots \\ &- \max_{\hat{\mathbf{x}} \in \mathcal{X}_{m,l}^{-\mathrm{MAP}}} \left\{ \sum_{i=1}^{N_T} \Lambda_i \left(\hat{\mathbf{x}}\right) \right\} \Bigg), \end{aligned} \qquad (4)$$

where $\hat{\mathbf{c}}^{\mathrm{MAP}}$ is the bit pattern of the MAP estimate and $\mathcal{X}_{m,l}^{-\mathrm{MAP}}$ the set of potential counter-hypotheses, for which $\hat{c}_{m,l} = -\hat{c}_{m,l}^{\mathrm{MAP}}$. The maxLogAPP detection problem may hence be solved by first finding the MAP estimate and then performing $N_T \cdot L$ searches which cover only a subset of the transmitter signal set, as proposed for the sphere detector in [2] and extended to general tree search algorithms in [5].

A lower complexity (and slightly suboptimal) alternative to [2] is the parallel smart candidate adding algorithm [6], which requires only a single pass through the detection tree and ensures that each node in the detection tree is visited only once. Specifically, in the PSCA algorithm counter-hypotheses are found *concurrently* with the MAP estimate as the breadth-first search proceeds through the detection tree, rather than through supplemental searches. This is similar to the "parallel sphere detector" approach taken in [7].

B. Ordering

It is widely recognized [8]–[10] that the order in which the signal components are detected has a significant impact on the tree search complexity and the quality of the detector output.

For decision feedback schemes (as a trivial special case) the Bell Labs Layered Space-Time (BLAST) ordering [11] has been developed to maximize performance. In general, it is attractive for tree search schemes to incorporate the ordering into the matrix decomposition step. Here, the sorted QR decomposition (SQRD) [12] is an attractive choice, as the layer ordering then becomes part of the Gram-Schmidt orthogonalization procedure of the QR decomposition, at negligible additional overhead. Thus, the decomposition effectively operates on a channel matrix whose columns are shuffled by a permutation matrix $\mathbf{P}$, such that $\mathbf{HP} = \mathbf{QR}$.

Note that the BLAST ordering is roughly twice as complex as the SQRD, but with improved performance as a result of its better ordering criterion [13]. Other useful orderings which we will discuss in more detail in Section IV-C include the one employed by the parallel detector (PD) [14], the fixed-complexity sphere decoder (FSD) [10], [15], and B-Chase preprocessing [1], [9], [13], [16].

IV. Fixed Complexity Framework

A. Classical M Algorithm and Special Cases

In this work, we will focus on breadth-first search algorithms, with the classical M algorithm [17] as the most famous example (often also referred to as QRD-M [18] algorithm). As any other breadth first scheme, it traverses the tree layer-by-layer; in this case by rejecting all but M nodes at a given layer before advancing to the next one [17]. Specifically, the b best children (with largest branch metrics) are extended from each of the M retained nodes, and of the bM contenders that result, only the M best are retained. In the case of hard-output detection, the candidate at the final detection layer is considered to be the hard-output decision. The performance of the M algorithm in hard-output systems approaches that of the joint maximum likelihood (JML) detector when M is large, but falls off significantly as M is decreased [19]. A common technique for soft-output versions of the M algorithm is to let $\mathcal{L}$ comprise the M best candidates at the final detection layer.

Given this description, we observe that the entire M algorithm can be parameterized by just two scalar values, M and b. Many detection algorithms are special cases of this scheme with specific parameterizations and preprocessing. The simplest example is the decision feedback detector, for which $M = b = 1$. With $b = Q$ and $M = \infty$[1], the algorithm turns into the maximum complexity, brute-force, APP approach which enumerates all possible transmit vectors. For $b = Q$, and arbitrary positive M, the M algorithm is sometimes referred to as the K-best approach [19].

B. Fixed-Complexity Framework

We can generalize the M algorithm by allowing the parameters M and b to be represented as $1 \times N_T$ vectors $\mathbf{M} = [M_1\ M_2\ \ldots\ M_{N_T}]$ and $\mathbf{b} = [b_1\ b_2\ \ldots\ b_{N_T}]$, respectively, where M_i represents the number of nodes retained at the ith layer of the detection tree and b_i represents the number of children extended from each parent at the ith tree layer[2].

This generalization can be extended further to include the parallel smart candidate adding schemes [6] by introducing a boolean vector $\mathbf{S} = [S_1\ S_2\ \ldots\ S_{N_T}]$, where S_i determines whether or not to perform smart candidate adding at the ith layer in the detection tree. This allows for more flexibility in the design of SCA algorithms than in [6].

With the introduction of $\mathbf{M}$, $\mathbf{b}$ and $\mathbf{S}$ vectors, as well as the specification of the preprocessing algorithm, we now have a framework which allows for the characterization of a large class of fixed complexity, single-pass, breadth-first MIMO detectors. While this framework is simple to describe, it enables a myriad of possibilities and brings to light many new design considerations. An appropriate configuration is crucial to achieve a desirable performance-complexity tradeoff.

Before proceeding it is important to assess the complexity of different algorithms within the established framework. A good measure of complexity for tree search schemes is the number of branch metric computations performed[3] μ. The total number of branch metric computations μ is a function of the number of nodes retained for a given layer in the detection tree ζ_i:

$$\zeta_i = \begin{cases} 1 & i = 0 \\ \min\left(\zeta_{i-1} b_i + S_i \kappa_i, M_i\right) & i > 0 \end{cases}, \tag{5}$$

where i is the detection layer, and where $\phi = 1$ for the complex-valued and $\phi = 2$ for the real-valued system model. κ_i denotes the number of branch metrics calculated as part of the smart candidate adding at layer i and is given by:

$$\kappa_i = \begin{cases} \max\left(L - 2(\sqrt{b_i} - 1), 0\right) & \phi = 1 \\ \max\left(\frac{L}{2} - (b_i - 1), 0\right) & \phi = 2 \end{cases}, \tag{6}$$

where for the complex-valued system model it is assumed that b_i is the square of an integer. The total number of branch metric computations for schemes in our framework becomes:

$$\mu = \sum_{i=1}^{\phi N_T} \left(\zeta_{i-1} b_i + S_i \kappa_i\right). \tag{7}$$

The final list size is $|\mathcal{L}| = \zeta_{\phi N_T}$.

C. Placement

We will now place existing fixed-complexity breadth-first detectors into the framework just presented and discuss the design considerations that accompany each detection scheme, as well as relationships amongst them. While the list of detection algorithms in this subsection does not claim to be complete, it does provide insight into many of the most common and effective fixed complexity breadth-first schemes.

Decision-Feedback: The simplest scheme to be captured by the presented framework is the decision-feedback (DFE), or

[1] An $\mathbf{M}$ value of ∞ implies that no pruning of the detection occurs.

[2] The root node of the detection tree corresponds to $i = 0$.

[3] Obviously, the *overall* complexity also includes a preprocessing step which, however, is only needed when the channel changes. Its impact on complexity thus depends on the channel coherence time/bandwidth. Furthermore, the complexity of different preprocessing schemes is often very similar.

successive interference cancelation (SIC), approach. After removing ("cancelling") the signal contribution of previous layers, this scheme will recursively determine the single best candidate at the currently considered layer (single enumeration, SE), and proceed with this decision to the next layer. Obviously, the tree width is minimized by this scheme. Good DFE performance thus heavily depends on making the correct decision in the initial detection layer (having the lowest diversity order). BLAST (or SQRD) should therefore be used to obtain the optimal (or nearly optimal) DF performance. DF detection is captured through the parameterization $\mathbf{M} = [1 \ \dots \ 1]$, $\mathbf{b} = [1 \ \dots \ 1]$, and $\mathbf{S} = [0 \ \dots \ 0]$ and the specification of the preprocessing algorithm. Consequently $\mu = \phi N_T$ and $|\mathcal{L}| = 1$.

Parallel Detector [14]: Rather than using SE at the first detected layer, the parallel detector uses full enumeration (FE) at this point, i.e., enumerates all candidates. All subsequent layers are detected using SE. The PD also adopts a special ordering, where the weakest received signal component is detected first. Subsequent layers use the BLAST ordering. The intuitive justification for such an approach is that in each layer where FE is used, no decision errors can occur. It is therefore desirable to use FE for the layers with the largest noise enhancement, to minimize performance loss. The PD uses the parameterization $\mathbf{M} = [Q \ ... \ Q]$, $\mathbf{b} = [Q \ 1 \ \dots \ 1]$, $\mathbf{S} = [0 \ \dots \ 0]$ and PD preprocessing. Thus, $\mu = \phi Q N_T$ and $|\mathcal{L}| = Q$.

B-Chase Detection [1], [9], [13], [16]: The B-Chase(ℓ) detector is a hard-output detector that generates a list of ℓ tentative decisions for the first detected symbol, and implements a bank of ℓ ordered decision-feedback detectors in parallel, one for each element of the list. In the case of hard-output detection, the final decision vector is the DF equalized output that minimizes the mean-squared error (MSE). The performance-complexity trade-off for Chase detection is easily adapted by adjusting ℓ, as Chase detection reduces to ordered DF when $\ell = 1$ and the PD when $\ell = Q$. Increasing ℓ improves performance at the cost of a complexity growing linearly in ℓ.

The B-Chase preprocessing approach results in substantial performance gains [1]. B-Chase preprocessing balances the goals of BLAST preprocessing (which performs well with SE) and PD ordering (which performs well with FE). It therefore gracefully trades off between the opposing design goals of maximizing vs. minimizing the SNR of the first detection layer by allowing the ordering algorithm to consider an increase in the number of enumerated child nodes as an effective SNR gain for the receiver. The B-Chase detector uses the parameterization $\mathbf{M} = [\ell \ ... \ \ell]$, $\mathbf{b} = [\ell \ 1 \ \dots \ 1]$, $\mathbf{S} = [0 \ \dots \ 0]$ and B-Chase preprocessing. Hence, it computes $\mu = \phi \ell N_T$ branch metrics and the list size is $|\mathcal{L}| = \ell$.

Fixed-Complexity Sphere Decoder [10], [15]: The FSD extends the PD to handle cases when the number of candidates enumerated at a detection layer is neither SE or FE. Specifically, when FE is used at any detection layer the FSD adopts the ordering criterion of the PD, otherwise it uses the BLAST ordering. Similar to the PD and the B-Chase detector, paths once generated are never pruned.

The FSD is capable of many parameterizations, where $\mathbf{S}$ is always the zero vector. The most effective parameterizations, however, are those of the PD and, for large dimensions such as 8×8 [10], the parameterization $\mathbf{M} = [Q \ Q^2 \ \dots \ Q^2]$ and $\mathbf{b} = [Q \ Q \ 1 \ \dots \ 1]$. In [15] it was shown that the FSD maintains the diversity order of the APP detector with a fixed complexity and order $\Theta(Q^{\sqrt{N_T}})$ if $N_R = N_T$, by using FE in the $p = \lfloor \sqrt{N_T} \rfloor$ first detection layers.

List Fixed-Complexity Sphere Decoder [20]: The list fixed-complexity sphere decoder (LFSD) is intended as a soft-output extension of the FSD. It builds on the FSD approach by typically computing more branch metrics than the FSD, in order for $\mathcal{L}$ to include more counter-hypotheses to the hard-output FSD decision vector. In [20] this was typically done using balanced powers of 2 for $b_2, \ \dots \ b_{N_T}$. In the event that this was not possible due to list length constraints, these powers of 2 are weighted to earlier layers in the detection tree.

Soft Fixed-Complexity Sphere Decoder [21]: We now describe the soft fixed-complexity sphere decoder (SFSD) which, while not a single-pass approach, has important relationships to several approaches mentioned in this work. The SFSD, like the LFSD, is a soft-output extension of the FSD. However, it is more similar to previously reported SCA approaches [2]–[6], with the process of smart candidate adding being referred to as "bit-negating" and "path augmentation" [21]. Specifically, the SFSD can be thought of as the combination of the hard-output FSD approach, used to generate the set $\mathcal{L}_{\mathcal{FSD}}$, with an iterative SCA type approach used to generate the set $\mathcal{L}_{\mathcal{SCA}}$, where $\mathcal{L} = \mathcal{L}_{\mathcal{FSD}} \cup \mathcal{L}_{\mathcal{SCA}}$. Unlike other SCA approaches, the SFSD typically employs FSD ordering and FE in the first detection layer(s). When only one iteration is performed (SCA augmentation of a single path), the SFSD approach is similar to an algorithmic realization in [5]. With multiple iterations (SCA extended paths), SFSD performance can be improved, but the performance improvements are relatively small and detection complexity is substantially increased.

Parallel Smart Candidate Adding [6]: The parallel smart candidate adding algorithm is an efficient way to achieve near-capacity performance for soft-output MIMO detectors, with lower complexity than the soft-output approaches mentioned so far [2], [5], [20], [21], using a single pass through the detection tree. For a given ith layer in the detection tree, the PSCA algorithm finds the b_i child nodes with lowest branch metrics for each parent node. The algorithm then establishes, from all enumerated child nodes, the one with the lowest total path metric, the "partial MAP estimate". For this child node all sibling nodes (i.e., child nodes from the same parent), possessing a Hamming distance of 1 in relation to the bit

	BLAST B-Chase(1) FSD(1)	PD B-Chase(Q) FSD(Q)	B-Chase(ℓ)	M(2, 2) (e.g. 1)	LFSD (e.g. 1)	LFSD (e.g. 2)	PSCA (e.g. 1)	PSCA (e.g. 2)
$\mathbf{M}$	[1 1 1 1]	[Q Q Q Q]	[ℓ ℓ ℓ ℓ]	[2 2 2 2]	∞	∞	∞	∞
$\mathbf{b}$	[1 1 1 1]	[Q 1 1 1]	[ℓ 1 1 1]	[2 2 2 2]	[Q 2 1 1]	[Q 2 2 2]	[1 1 1 1]	[4 4 1 1]
$\mathbf{S}$	[0 0 0 0]	[0 0 0 0]	[0 0 0 0]	[0 0 0 0]	[0 0 0 0]	[0 0 0 0]	[1 1 1 1]	[1 1 1 1]
Preprocessing algorithm	BLAST	PD	B-Chase	SQRD	FSD	FSD	SQRD	SQRD
μ 16-QAM	4	64	4ℓ	14	112	240	44	96
SNR[dB] 10^{-2} Uncoded BER	16.05	12.85	–	13.70	–	–	13.06	12.81
SNR[dB] 10^{-5} Coded BER	13.08	9.67	–	11.41	9.33	8.84	9.37	8.95
μ 64-QAM	4	256	4ℓ	14	448	960	64	134
SNR[dB] 10^{-2} Uncoded BER	21.66	17.88	–	19.50	–	–	18.44	18.06
SNR[dB] 10^{-5} Coded BER	18.01	13.84	–	16.35	13.37	12.96	13.72	13.22

Table I
COMPARISON OF VARIOUS FIXED-COMPLEXITY BREADTH-FIRST MIMO DETECTION SCHEMES FOR A 4×4 MIMO CHANNEL.

pattern corresponding to the partial MAP estimate at the ith detection layer, are enumerated. This strategy ensures that a counter-hypothesis exists for every bit for the detector's soft output. The PSCA algorithm is captured through the parameterization $\mathbf{M} = [\infty \ \dots \ \infty]$, $\mathbf{S} = [1 \ \dots \ 1]$, and an appropriate selection of the preprocessing algorithm and the vector $\mathbf{b}$. Note that in the case $b_i > 3$ ($b_i > 2$ for the real-valued model) a slight variance in complexity is possible, since the b_i closest points will then generate a varying number of counter-hypotheses to the partial MAP estimate. Consequently, the provided μ figures correspond to an upper complexity bound. In practice, one would accept some redundant nodes in the detection tree and work with fixed complexity at this upper complexity bound.

Other: There exist many other fixed (or quasi-fixed) complexity breadth-first algorithms. Examples of algorithms which we wish to mention, but due to space limitations cannot be treated in detail, include [22]–[24].

A summary of the above algorithmic placements is provided in Table IV-C for a 4×4 MIMO channel. In addition to specifying the parameterization of the aforementioned algorithms, it provides the number of branch metric calculations for a 4×4 MIMO system using 16-QAM and 64-QAM transmission alphabets and a complex system model, as well as the SNR required for a uncoded detection BER performance of 10^{-2} and a coded BER of 10^{-5}. The coded results are found using the simulation parameters presented in subsection V-C. Additionally, the preferred channel decomposition for each algorithm is provided[4]. Because multiple parameterizations are possible for the M, LFSD, and PSCA algorithms, typical parameterizations are provided, as evidenced by [6], [20], [22]. Results corresponding to these and other parameterizations will now be provided.

V. ANALYSIS

A. Simulation Setup

We consider transmission over a spatially and temporally i.i.d. fading 4×4 MIMO channel, using 16-QAM and 64-QAM modulation alphabets. The information block size (including tail bits) is 9216 bits. Detection is performed based on the complex-valued system model. Since tree search schemes with fixed (or tightly bounded) detection complexity benefit from the use of MMSE preprocessing, we employ unbiased MMSE detection [25] with all techniques. For coded transmission, we use a setup equivalent to the one in [26]: a rate 1/2 PCCC based on $(7_R, 5)$ convolutional codes using 8 internal iterations of logMAP decoding. The LLRs were clipped at a magnitude of ± 6 for all investigated techniques.

B. Hard-Output Results

Fig. 2 shows uncoded results for a 4×4 MIMO system employing 16-QAM and 64-QAM transmission. Curves with square markers denote the B-Chase family of hard-output detection algorithms, including the BLAST-ordered DF detector, B-Chase($\ell = 1$), and the parallel detector / FSD, B-Chase($\ell = Q$). Also shown in triangular markers is the M algorithm, where results are provided for the case $M = b = 2$. Joint maximum likelihood (JML) detection serves as reference.

A significant difference in performance between the various detection techniques can be observed at bit error rates of 10^{-3} (and below), a target BER level often considered for performance comparisons and algorithm optimization. However, at code rates of $R_c = 1/2$ and below, powerful coding schemes can achieve successful decoding at much higher error rates[5], even above 10^{-1}. A detector output BER level of 10^{-2} will thus be much more relevant for practical applications. In this regime, the difference in performance between the different techniques is much less significant (cf. Fig. 2), and their ability to generate reliable soft outputs will be of crucial importance.

In order to provide a more accurate picture of the relation between the aforementioned algorithms, Fig. 3 provides a performance-complexity plot for the detectors presented in Fig. 2, with the addition of the PSCA as reference, although the PSCA algorithm was intended as a soft-output detector. Complexity is measured in terms of the number of complex detection tree branch metric computations μ. The subscript $\mathbf{S}$ is used to denote that the vector to which it is attached is $\mathbf{S}$.

[4]All channel decompositions in this paper have complexity order $\Theta(N_T)^3$.

[5]Note that the Shannon bound for rate $1/2$ on the binary symmetric channel (i.e., hard output detection) is at an error rate around 16% [27].

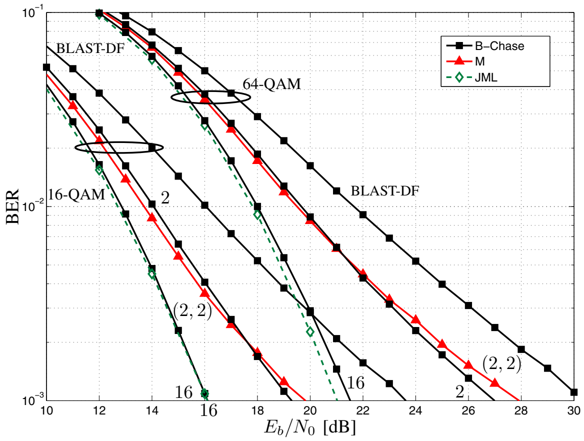

Figure 2. Hard-output BER performance for various fixed complexity breadth-first 4×4 MIMO detection schemes using 16-QAM and 64-QAM alphabets in Rayleigh fading (unbiased [25]).

We observe for hard-output detection that B-Chase detection has the most desirable performance-complexity tradeoff.

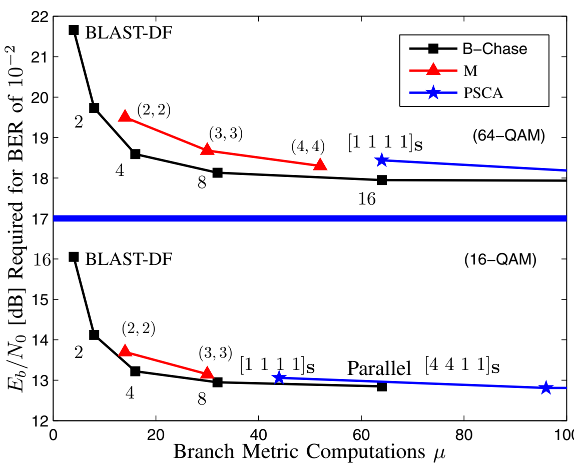

Figure 3. Performance vs. complexity for various fixed complexity hard-output breadth-first 4×4 MIMO detection schemes in Rayleigh fading.

C. Soft-Output Results

Fig. 4 provides performance results for 16-QAM transmission in a Turbo coded system setup. Results provided are for the B-Chase detection family, the M algorithm, the PSCA algorithm and the LFSD. For the B-Chase detection family results are given for BLAST-ordered DF, B-Chase(4), and the PD/B-Chase(Q)/FSD. For the M algorithm results are provided for the cases $M = b = 2$ and $M = b = 4$.

For the PSCA and LFSD algorithms results are provided for the representative cases outlined in Table IV-C. Results are not provided for the SFSD approach, because it is not a single-pass detection algorithm. Note that this algorithm could be converted to single-pass by using a slightly suboptimal PSCA-like approach. The complexity of such an approach would

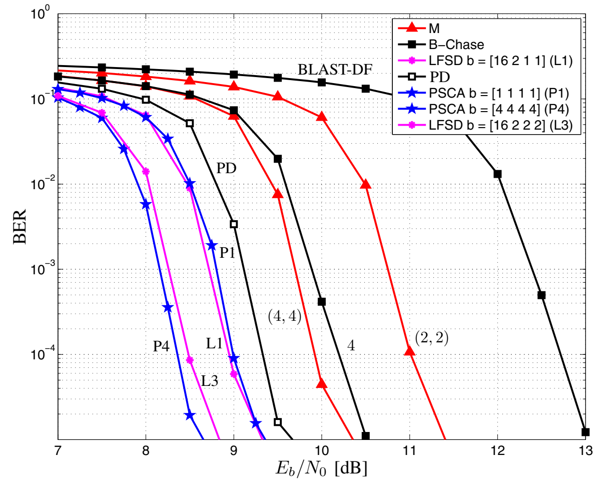

Figure 4. Performance for various soft-output 4×4 MIMO detection schemes using 16-QAM alphabets in Rayleigh fading.

be higher than PSCA (due to FE in the first layer) but less than LFSD, which uses larger b_i values than SCA approaches, resulting in a multiplicative factor on the number of detection tree branch metric computations.

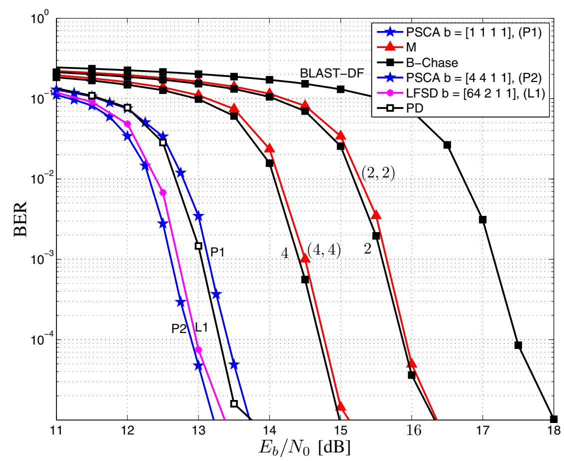

Figure 5. Performance for various soft-output 4×4 MIMO detection schemes using 64-QAM alphabets in Rayleigh fading.

Fig. 5 provides performance results for the same setup and the same classes of algorithms, but 64-QAM transmission and slightly different configurations for the PSCA and the LFSD algorithm. The performance gap between the PSCA and LFSD approaches relative to the several cases of the B-Chase and M detection algorithms is even wider than in Fig. 4. The soft-output PD is performing quite well, at the expense of having to compute 64 branch metrics in the first detection layer.

Fig. 6 provides a performance-complexity plot for a coded 4×4 MIMO system in Rayleigh fading under 16-QAM and 64-QAM transmission. The subscript $\mathbf{b}$ is used to denote that the displayed vector corresponds to the configuration of $\mathbf{b}$.

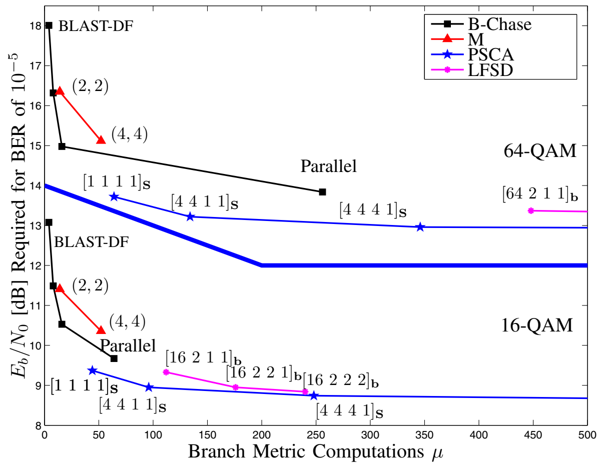

Figure 6. Performance vs. complexity for various soft-output MIMO detection schemes for 4×4 MIMO in Rayleigh fading.

For the coded case we observe that in order to achieve better performance it is necessary to use either LFSD or PSCA detection, although the PSCA algorithm achieves an operating point much lower on the complexity axis relative to the LFSD. This is due in part to the fact that the PSCA algorithm is interested in obtaining, if possible, only the bits relevant to the MaxLogMAP approximation and the LFSD is interested in obtaining candidate vectors which are close in Euclidean distance to the received vector.

VI. Conclusions

In this contribution, we presented a fixed complexity framework for breadth-first tree search detection in MIMO systems. Existing approaches for fixed-complexity breadth-first detection were placed into this framework and a performance-complexity analysis was performed. Results showed that, amongst hard-output detection schemes, B-Chase detection was the most desirable approach. Amongst soft-output detection schemes the PSCA and LFSD algorithm were shown to be viable options, but the PSCA algorithm was shown to have a more favorable performance-complexity tradeoff.

References

[1] D. W. Waters and J. R. Barry, "The Chase family of Detection Algorithms for MIMO Channels," *IEEE Transactions on Signal Processing*, vol. 56, pp. 739–747, Feb. 2008.

[2] P. Marsch, E. Zimmermann, and G. Fettweis, "Smart Candidate Adding: A new Low-Complexity Approach towards Near-Capacity MIMO Detection," in *13th European Signal Processing Conference (EUSIPCO'05)*, Antalya, Turkey, 04.-08. Sep. 2005.

[3] M. S. Yee, "Max-log-MAP sphere decoder," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, vol. 3, 18.-23. Mar. 2005, pp. 1013–1016.

[4] R. Wang and G. Giannakis, "Approaching MIMO channel capacity with Soft Detection Based on Hard Sphere Decoding," *IEEE Transactions on Communications*, vol. 54, pp. 587–590, Apr. 2006.

[5] E. Zimmermann and G. Fettweis, "Generalized Smart Candidate Adding for Tree Search Based MIMO Detection," in *ITG/IEEE Workshop on Smart Antennas (WSA'07)*, Vienna, Austria, Feb. 2007.

[6] E. Zimmermann, D. L. Milliner, G. Fettweis, and J. R. Barry, "A parallel smart candidate adding algorithm for soft-output MIMO detection," in *Proc. 7th International ITG Conference on Source and Channel Coding (SCC 08)*, Ulm, Germany, Jan. 2008.

[7] J. Jalden and B. Ottersten, "Parallel Implementation of a Soft Output Sphere Decoder," in *Conference Record of the 39th Asilomar Conference on Signals, Systems and Computers*, Oct. 2005, pp. 581–585.

[8] M. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, pp. 2389–2402, Oct. 2003.

[9] D. W. Waters and J. R. Barry, "The Chase Family of Detection Algorithms for MIMO Channels," in *International Global Communications Conference (Globecom 2004)*, vol. 4, 29.-3. Nov./Dec. 2004, pp. 2635–2639.

[10] L. G. Barbero and J. S. Thompson, "A Fixed-Complexity MIMO Detector Based on the Complex Sphere Decoder," in *IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC 06)*, Cannes, France, Jul. 2006.

[11] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for wireless communication at high spectral efficiency," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1841–1852, Nov. 1999.

[12] D. Wübben, R. Böhnke, V. Kühn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronic Letters*, vol. 37, pp. 1348–1350, Oct. 2001.

[13] D. W. Waters, "Signal Detection Strategies and Algorithms for Multiple-Input Multiple-Output Channels," PhD in Electrical and Computer Engineering, Georgia Institute of Technology, Nov. 2005.

[14] Y. Li and Z.-Q. Luo, "Parallel detection for V-BLAST system," in *Proc. IEEE International Conference on Communications (ICC 02)*, vol. 1, New York, NY USA, Apr. 2002, pp. 340–344.

[15] J. Jalden, L. G. Barbero, B. Ottersten, and J. S. Thompson, "Full Diversity Detection in MIMO Systems with a Fixed-Complexity Sphere Decoder," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '07)*, Honolulu, Hawaii, USA, Apr. 2007.

[16] D. W. Waters and J. R. Barry, "The Sorted-QR Chase Detector for Multiple-Input Multiple-Output Channels," in *IEEE Wireless Communications and Networking Conference 2005*, New Orleans, Louisiana, USA, 13.-17.Mar. 2005, pp. 538–543.

[17] J. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Transactions on Communications*, vol. 32, pp. 169–176, Feb. 1984.

[18] W. H. Chin, "QRD Based Tree Search Data Detection for MIMO Communication Systems," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 3, Stockholm, Sweden, May 2005, pp. 1624–1627.

[19] 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, pp. 491–503, Mar. 2006.

[20] L. G. Barbero and J. S. Thompson, "Extending a Fixed-Complexity Sphere Decoder to Obtain Likelihood Information for Turbo-MIMO Systems," *IEEE Transactions on Vehicular Technology*, 2008, to appear.

[21] L. G. Barbero, T. Ratnarajah, and C. Cowan, "A low-complexity soft-mimo detector based on the fixed-complexity sphere decoder," in *IEEE International Conference on acoustics, speech and signal processing (ICASSP '08), to appear*, Las Vegas, USA, 30.-4. Mar./Apr. 2006.

[22] D. L. Milliner and J. R. Barry, "An Adaptive M-Algorithm for Detection of Multiple-Input Multiple-Output Channels," in *IEEE Signal Processing Advances in Wireless Communications, (SPAWC'07)*, Helsinki, Finland, 17.-20. Jun. 2007.

[23] K. Jeon, H. Kim, and H. Park, "An Efficient QRD-M Algorithm Using Partial Decision Feedback Detection," in *40th Asilomar Conference on Signals, Systems, and Computers*, Oct. 2006.

[24] W. Shin, H. Kim, M. Son, and H. Park, "An improved LLR computation for QRM-MLD in coded MIMO systems," in *IEEE Vehicular Technology Conference (VTC2007 Fall)*, Baltimore, Maryland, USA, Sep. 2007.

[25] E. Zimmermann and G. Fettweis, "Unbiased MMSE Tree Search MIMO Detection," in *International Symposium on Wireless Personal Multimedia Communications (WPMC'06)*, San Diego, USA, 17.-20. Sep. 2006.

[26] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, pp. 389–399, Mar. 2003.

[27] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.