# A Sphere Decoding Approach for the Vector Viterbi Algorithm

Peter Kairouz, Aolin Xu, Naresh Shanbhag, and Andrew Singer

Coordinated Science Laboratory / Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign, Urbana, Illinois, 61801
Email: {kairouz2, aolinxu2, shanbhag, acsinger}@illinois.edu

***Abstract*—High speed multi-input multi-output (MIMO) communication systems suffer from inter-channel and inter-symbol-interference (ICI and ISI). The vector Viterbi algorithm (VVA) is a maximum likelihood sequence detection (MLSD) algorithm for MIMO frequency selective channels. MLSD algorithms are desired because they minimize the probability of sequence detection error. However, they suffer from very high computational complexity. In this work, we show how a sphere decoding like algorithm can be used to reduce the complexity of VVA while preserving its optimality. For a $2 \times 2$ MIMO system with 16-QAM signal constellation, our algorithm cuts VVA's complexity by $50\%$ at an SNR of $10$ dB and by $60\%$ at an SNR of $15$ dB.**

## I. Introduction

The vector Viterbi algorithm (VVA) extends the conventional Viterbi algorithm (VA) to make it operate on vector transmitted symbols [1], [2]. What helped in the negligence of this algorithm is the emergence of multi-input multi-output (MIMO) Orthogonal Frequency Division Multiplexing (OFDM) which reduced the complexity of the receiver while still achieving MIMO capacity gains [3]. Let $L$, $N$, $N_t$, and $|\mathcal{A}|$ denote the channel memory, number of transmitted symbols per block, number of transmit antennas, and size of signal constellation respectively. The computational complexity of a MIMO OFDM based detection scheme is $O\left(N_t N \log(N) + N|\mathcal{A}|^{N_t}\right)$ in comparison to $O\left(N|\mathcal{A}|^{LN_t}\right)$ for the vector Viterbi algorithm. Thus, MIMO OFDM systems are computationally attractive when $L$ is large. Nonetheless, this advantage comes at the following expenses:

1) OFDM requires the addition of a cyclic prefix which reduces the rate of communication. If the channel is changing rapidly, the size of the block cannot be made long and hence the percentage overhead is even larger (up to $25\%$ in some cases).
2) The transmitter's complexity is increased because $N_t$ $N$-point IFFTs have to be computed at the transmitter side.
3) The orthogonality between sub-carriers might be lost due to Doppler shifts and channel nonlinearities.
4) The peak to average power (PAPR) ratio of an OFDM system is significantly larger than that of a single carrier system.

In addition to the above disadvantages, some communication technologies, such as under-water acoustic systems and fiber optic systems, cannot easily make use of OFDM due to various transmitter and receiver limitations. For example, the laser sources in fiber optic systems cannot be adequately modulated by arbitrary shaped signals having very high PAPR. Moreover, high PAPR signals excite fiber nonlinearities which destroy the orthogonality between sub-carriers. This is why the state-of-the-art optical communication systems use simple modulation schemes such as quadrature phase shift keying.

More importantly, single carrier systems are used in the latest wireless communication standards. For example, the Long Term Evolution Advanced (LTE-A) wireless standard does not use OFDM for the uplink channel due to its high PAPR [4]. Instead, LTE-A uplink systems use Single Carrier Frequency Division Multiple Access (SC-FDMA) modulation. Even though SC-FDMA divides the resources among users in the frequency domain, the transmission uses single carrier modulation. Therefore, it is natural to revisit single carrier systems and look for ways to reduce the computational complexity of sequence detection. Our sphere decoding (SD) approach for VVA is a promising technique that reduces VVA's complexity significantly while preserving its optimality.

The remainder of this paper is organized as follows. In Section II, we review the sphere decoding algorithm. In Section III, we show how SD can be used to perform MIMO MLSD at a reduced computational cost. In Section IV, we compare the complexity of VVA to our scheme and show that for a $2 \times 2$ MIMO system with 16-QAM signal constellation, up to $60\%$ of VVA's complexity can be saved using our approach.

## II. Sphere Decoding

A frequency flat MIMO system is described by

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

where $\mathbf{x}$ is an $N_t$-dimensional vector, $\mathbf{y}$ is an $N_r$-dimensional vector, $\mathbf{H}$ is an $N_r \times N_t$ matrix, and $\mathbf{v}$ is a zero mean complex Gaussian noise vector with a covariance matrix equal to $N_0 \mathbf{I}_{N_r}$. The optimal detector, in terms of minimizing the symbol error rate (SER), is the maximum likelihood (ML) detector and is given by

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{A}^{N_t}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{2}$$

where $\mathcal{A}$ represents the signal constellation set. The ML detector finds the nearest neighbor to the received vector among all possible constellation points (lattice points) by performing an exhaustive search. Unfortunately, the computational complexity of this algorithm is exponential in $N_t$. Nonetheless, there exists a clever algorithm that performs ML detection at a substantially lower cost.

The basic idea of Sphere Decoding (SD) is to perform a search over the constellation points that lie within a sphere of radius $r$ centered around the received vector [5]. This is done by representing the signal constellation via an $|\mathcal{A}|$-ary tree of depth $N_t$. The nodes at depth $i$ correspond to instances of the $(N_t - i + 1)^{th}$ entry in $\mathbf{x}$. Thus, the tree has $|\mathcal{A}|^{N_t}$ leaves, each corresponding to an instance of $\mathbf{x}$. Assume that $N_r \geq N_t$, then by the QR decomposition $\mathbf{H} = \mathbf{Q}\left[\mathbf{R}^* \mathbf{0}_{N_t \times (N_r - N_t)}\right]^*$, where $\mathbf{Q}$ is an $N_r \times N_r$ unitary matrix, $\mathbf{R}$ is an $N_t \times N_t$ upper triangular matrix, and $\mathbf{A}^*$ refers to the conjugate transpose of $\mathbf{A}$. As the norm is invariant to unitary transforms, the ML rule can be rewritten as

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x} \in \mathcal{A}^N}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \\ &= \underset{\mathbf{x} \in \mathcal{A}^N}{\operatorname{argmin}} \|\mathbf{Q}^*\mathbf{y} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(N_r - N_t) \times N_t} \end{bmatrix} \mathbf{x}\|^2 \\ &= \underset{\mathbf{x} \in \mathcal{A}^N}{\operatorname{argmin}} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|^2, \end{aligned} \tag{3}$$

where $\tilde{\mathbf{y}}$ is a vector containing the first $N_t$ entries of $\mathbf{Q}^*\mathbf{y}$. Due to the triangular structure of $\mathbf{R}$, the vector norm can now be rewritten as a sum of scalar norms

$$\begin{aligned} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|^2 &= \sum_{i=1}^{N_t} \left| \tilde{y}_i - \sum_{l=i}^{N_t} r_{i,l} x_l \right|^2 \\ &= \sum_{i=1}^{N_t} e_i\left(x_i, ..., x_{N_t}\right) \\ &= e_1\left(x_1, ..., x_{N_t}\right) + ... + e_{N_t}\left(x_{N_t}\right), \end{aligned} \tag{4}$$

where $e_i\left(x_i, ..., x_{N_t}\right) = \left|\tilde{y}_i - \sum_{l=i}^{N_t} r_{i,l} x_l\right|^2$. Note that the last $N_t - i + 1$ summands in (4) depend only on the last $N_t - i + 1$ transmitted symbols and they are all non-negative. We define the partial Euclidean distance (PED) as $p_i = \sum_{j=N_t-i+1}^{N_t} e_j\left(x_j, ..., x_{N_t}\right)$ for $i = 1, ..., N_t$. This sequence is computed recursively by traversing the tree from the root node down to a leaf node. For $i = 1$, $p_1 = e_1\left(x_{N_t}\right)$ and for $i = 2, ..., N_t$, $p_i = p_{i-1} + e_{N_t-i+1}\left(x_{N_t-i+1}, ..., x_{N_t}\right)$. Notice that $p_i$ is a non-negative and non-decreasing sequence and that $p_i \leq \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|^2$. Therefore, it is safe to drop all candidate vectors $\mathbf{x}^j$'s that end with the same $i$ symbols if any $p_i$ exceeds a specified radius $r$. This technique is referred to as tree pruning.

Tree pruning is a smart way of eliminating the lattice points that do not lie inside the sphere of radius $r$. However, we still do not know how to choose $r$. This can be done in a variety of ways. A simple scheme would set $r$ to infinity and run a depth first search algorithm until the left most leaf node is reached. At this point, $r$ is updated to become equal to the Euclidean distance of that particular instance of $\mathbf{x}$. The depth first search algorithm is then resumed and the aforementioned pruning process is applied whenever some $p_i$ exceeds $r$. A leaf node is reached only if the distance between the received vector and that particular instance of $\mathbf{x}$ is less than $r$. In this case, the radius is updated to become equal to this new Euclidean distance and the process is continued until all leafs are either visited or pruned.

The complexity of SD is random as it depends on the quality of the channel realization which is a random variable. Moreover, the performance is a function of SNR. At high SNRs the savings are large because very few lattice points lie inside the sphere. However, marginal gains are achieved if the transformed lattice $\mathbf{H}\mathbf{x}$ happened to be such that all the points are close to each other. Therefore, SD still suffers from a worst case exponential complexity. Nonetheless, this happens at a very low probability (especially when the SNR is high). It was shown in [6], [7] that the expected complexity of SD is usually polynomial in $N_t$ for a wide range of SNRs and $N_t$. In fact for high SNRs, the expected complexity is cubic in $N_t$.

## III. Frequency Selective Systems

A frequency selective MIMO system is described by

$$\mathbf{y}[n] = \sum_{k=0}^{L-1} \mathbf{H}[k]\mathbf{x}[n-k] + \mathbf{v}[n], \tag{5}$$

where $\mathbf{y}[n]$ and $\mathbf{x}[n]$ are the detected and transmitted symbol vectors respectively. In (5), $L$ represents the number of nonzero taps in $\mathbf{H}[n]$, the channel's matrix impulse response, and is given by $T_d/T_s$ where $T_d$ is the channel's delay spread and $T_s$ is the sampling period. In our analysis, we assume that the channel's matrix impulse response $\mathbf{H}[n]$ is fixed for $N$ consecutive transmissions.

### A. Prior Work

Sphere decoding has been recently introduced as a low complexity detection algorithm for single carrier MIMO frequency selective systems [8]. We define the following vectors:

$$\begin{aligned} \mathbf{y} &= [\mathbf{y}^*[1], \mathbf{y}^*[2], \ldots, \mathbf{y}^*[N+L-1]]^* \\ \mathbf{v} &= [\mathbf{v}^*[1], \mathbf{v}^*[2], \ldots, \mathbf{v}^*[N+L-1]]^* \\ \mathbf{x} &= [\mathbf{x}^*[1], \mathbf{x}^*[2], \ldots, \mathbf{x}^*[N]]^*. \end{aligned} \tag{6}$$

Using (5) and (6), we can write the input-output relation for the dispersive channel in a matrix form as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{7}$$

where $\mathbf{H}$ is an $N_r\left(N + L - 1\right) \times N_t N$ block Toeplitz matrix given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}[0] & & & \\ \mathbf{H}[1] & \mathbf{H}[0] & & \\ & \ddots & \ddots & \\ & & \mathbf{H}[L-1] & \mathbf{H}[L-2] \\ & & & \mathbf{H}[L-1] \end{bmatrix}. \tag{8}$$

Given $\mathbf{x}$ we know that

$$f(\mathbf{y}|\mathbf{x}) \sim \mathcal{CN}(\mathbf{Hx}, N_0 \mathbf{I}_{N+L-1}), \quad (9)$$

where $\mathbf{Hx}$ is the mean vector and $N_0\mathbf{I}_{N+L-1}$ is the covariance matrix of the complex Gaussian distribution. Therefore, the optimal detection rule is given by

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{A}^{N_t N}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{Hx}\|^2. \quad (10)$$

A straightforward implementation will perform an exhaustive search over all $|\mathcal{A}|^{N_t N}$ possible transmit vectors, which is stupendously expensive for large $N_t$ or $N$. Observe that the problem in (10) is identical to the one in (3). Therefore, sphere decoding can be used to reduce the complexity of MLSD detection. However, the dimension of the equivalent frequency flat MIMO system is $N_r(N+L-1) \times N_t N$ and thus, the expected complexity of this approach is at best polynomial in $N_t N$. This figure can be significantly larger than $O\left(N|\mathcal{A}|^{LN_t}\right)$ for large $N$ or small $L$. Therefore, VVA seems to be more attractive for large $N$.

### B. The Vector Viterbi Algorithm

The vector Viterbi algorithm (VVA) is the vector version of the popular Viterbi algorithm [1], [2]. The VVA uses the entire received sequence $\mathbf{y}$ to detect the transmitted sequence $\mathbf{x}$. Both $\mathbf{x}$ and $\mathbf{y}$ have been defined in (6). It is convenient to define the mean vector $\boldsymbol{\mu}(\mathbf{x}) = \mathbf{Hx}$ and divide it into $N+L-1$ sub-vectors where the $i^{th}$ sub-vector is given by

$$\boldsymbol{\mu}_k(\mathbf{x}) = \sum_{l=0}^{L-1} \mathbf{H}[l]\mathbf{x}[k-l]. \quad (11)$$

Similarly, we can divide the vector $\mathbf{y}$ into $N+L-1$ sub-vectors where the $k^{th}$ sub-vector is $\mathbf{y}_k = \mathbf{y}[k]$. The optimization problem in (10) can now be written as

$$\begin{aligned}\hat{\mathbf{x}} &= \underset{\mathbf{x}\in\mathcal{A}^{N_t N}}{\operatorname{argmin}} \|\mathbf{y} - \boldsymbol{\mu}(\mathbf{x})\|^2 \\ &= \underset{\mathbf{x}\in\mathcal{A}^{N_t N}}{\operatorname{argmin}} \sum_{k=1}^{N+L-1} \|\mathbf{y}_k - \boldsymbol{\mu}_k(\mathbf{x})\|^2 \\ &= \underset{\mathbf{x}\in\mathcal{A}^{N_t N}}{\operatorname{argmin}} \mathcal{P}_{N+L-1}(\mathbf{x}), \end{aligned} \quad (12)$$

where $\mathcal{P}_i(\mathbf{x}) = \sum_{k=1}^{i} \|\mathbf{y}_k - \boldsymbol{\mu}_k(\mathbf{x})\|^2$ is called the $i^{th}$ path metric. The VVA performs the above minimization with a complexity that is linear in $N$. Unlike the previously derived SD algorithm, VVA exploits the fact that the channel has memory limited to $L$. This is known as the Markovian property of the channel. We define the state $\mathcal{S}_k$ at time $k$ to be

$$\mathcal{S}_k = (\mathbf{x}[k-1], \mathbf{x}[k-2], \ldots, \mathbf{x}[k-L+1]) \quad (13)$$

As shown in Figure 1, the state evolution in time can be represented using a trellis diagram. Accordingly, finding the maximum-likelihood sequence estimate is equivalent to finding the shortest path through the trellis. Note that $\boldsymbol{\mu}_k(\mathbf{x})$ is only a function of $\mathbf{x}[k]$ and $\mathcal{S}_k$. Let $\mathcal{S}_k^j$ and $\mathbf{x}^i[k]$ represent instances of $\mathcal{S}_k$ and $\mathbf{x}[k]$ respectively. We associate the following branch metric

$$\mathcal{B}\left(\mathbf{y}[k], \mathcal{S}_k^j, \mathbf{x}^i[k]\right) = \|\mathbf{y}_k - \boldsymbol{\mu}_k\left(\mathcal{S}_k^j, \mathbf{x}^i[k]\right)\|^2 \quad (14)$$

with each branch emanating from $\mathcal{S}_k^j$ and terminating in $\mathcal{S}_{k+1}^i$. Note that the vectors $\mathbf{x}[k-1], \ldots, \mathbf{x}[k-L+2]$ are exactly the same for both states. Each state $\mathcal{S}_k^j$ can terminate in one of $|\mathcal{A}|^{N_t}$ states because the only new entry in $\mathcal{S}_{k+1}^i$ is $\mathbf{x}^i[k]$. The vector Viterbi algorithm uses dynamic programming to implement a breadth-first search on a trellis. The key observation is that the minimization could be solved recursively by noting that $\mathcal{P}_k = \mathcal{P}_{k-1} + \mathcal{B}\left(\mathbf{y}[k], \mathcal{S}_k^j, \mathbf{x}^i[k]\right)$. Therefore, to find the shortest path, it is sufficient to solve the following problem

$$\mathcal{P}_k^i = \min_{j\in\mathcal{F}} \mathcal{P}_{k-1}^j + \mathcal{B}\left(\mathbf{y}[k], \mathcal{S}_k^j, \mathbf{x}^i[k]\right), \quad (15)$$

for every $\mathcal{S}_k^i \in \mathcal{S}_k$ and $k = 1, ..., N+L-1$. In (15), $\mathcal{F}$ contains the indices of the states, at stage $k-1$, that are allowed to transition to $\mathcal{S}_k^i$. Observe that for $k = N+L-1$, the solution to $\min_i \mathcal{P}_{N+L-1}^i$ is the solution to the MIMO MLSD problem in (12).

### C. Combined SD-VVA

The computational complexity of VVA is equal to the product of the number of computations required per state ($|\mathcal{A}|^{N_t}$), the number of states per stage ($|\mathcal{A}|^{N_t(L-1)}$), and the number of stages ($N+L-1$). As a result, the complexity grows linearly with the block length and exponentially with the number of transmitters and memory length. In what follows, we derive a new, lower complexity, optimal sequence detection algorithm. The aim is to break down the exponential number of computations required per state to something polynomial (often cubic) in $N_t$. This reduction in complexity is made possible by observing that the selection of the surviving path for each state can be computed via a tree based algorithm similar to the one used in sphere decoding. We define a super state $\underline{\mathcal{S}}_{k-1}$ to be the set of states $\mathcal{S}_{k-1}$ that differ only by $\mathbf{x}[k-L+1]$. Observe, from Figure 1, that there is a transition from each $\mathcal{S}_{k-1}^j \in \underline{\mathcal{S}}_{k-1}^l$ to one $\mathcal{S}_k^i \in \underline{\mathcal{S}}_k^m$. Furthermore, the first $L-2$ entries in $\mathcal{S}_{k-1}^j$ are identical to the last $L-2$ entries in $\mathcal{S}_k^i$. Thus, the following holds:

$$\begin{aligned}\mathcal{P}_k^i &= \min_{j\in\mathcal{F}} \mathcal{P}_{k-1}^j + \mathcal{B}\left(\mathbf{y}[k], \mathcal{S}_k^j, \mathbf{x}^i[k]\right) \\ &= \min_{j\in\mathcal{F}} \mathcal{P}_{k-1}^j + \|\mathbf{y}_k - \boldsymbol{\mu}_k\left(\mathcal{S}_k^j, \mathbf{x}^i[k]\right)\|^2 \\ &= \min_{j\in\mathcal{F}} \mathcal{P}_{k-1}^j + \|\mathbf{z}_k - \mathbf{G}\mathbf{x}^j\|^2, \end{aligned} \quad (16)$$

where $\mathbf{z}_k = \mathbf{y}_k - \sum_{l=0}^{L-2} \mathbf{H}[l]\mathbf{x}^i[k-l]$, $\mathbf{G} = \mathbf{H}[L-1]$, and $\mathbf{x}^j = \mathbf{x}^j[k-L+1]$. Had the term $\mathcal{P}_{k-1}^j$ not existed in (16), this minimization would have resembled to the standard frequency flat MIMO ML detection problem in (3). In this case, the complexity can be reduced by solving for the surviving branch via a sphere decoding approach as detailed in Section II. However, in our case every path is biased by a different

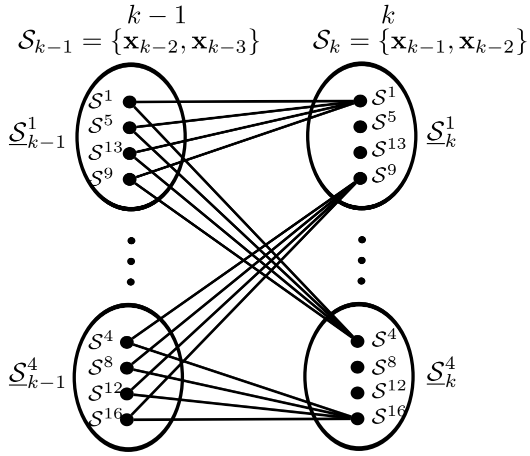


Fig. 1. Trellis super state grouping in a $2 \times 2$ system with BPSK signalling and memory length of 3

quantity $\mathcal{P}^j_{k-1}$ that is only determined when we traverse the tree from the root node down to a leaf node. This problem is clearly more complicated than the frequency flat MIMO ML one and cannot be solved using the standard sphere decoding algorithm. Nonetheless, we suggest modifying the tree representation by appending the path metrics to the leaf nodes. This is depicted in Figure 2 where the tree has been extended to incorporate the effect of $\mathcal{P}^j_k$'s. Thus, we can now run a depth first search SD algorithm using the tree in Figure 2 to solve for the $i^{th}$ state's path metric at $k$. The very first time the radius is updated it will include both the branch and path metrics of the state that corresponds to the left most branch in the tree. A smart approach that would lead to larger computational savings would first rearrange the branches of the tree in Figure 2 so that the left most leaf node corresponds to the instance that has the smallest path metric and the right most leaf node corresponds to the instance that has the largest path metric. This leads to an improved performance as the radius is always chosen to be as small as possible. However, this approach necessitates the need to keep a sorted list of path metrics which complicates the implementation of the algorithm and increases the number of comparisons needed.

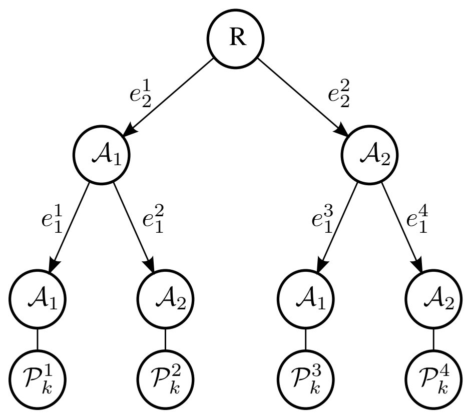


Fig. 2. Tree representation for a $2 \times 2$ BPSK system at stage $k+1$

The performance of the combined SD-VVA approach depends on how large the path metrics are relative to the branch metrics. Little savings can be achieved if the $\mathcal{P}^j_k$'s are much larger than the weights ($e^j_i$'s) shown in Figure 2. In this case, almost all leaf nodes would have to be visited. Therefore, large savings can be achieved if the trellis is shortened from $N$ to $5L$. This ensures that the path metrics do not accumulate and are still comparable to all other weights and thus, pruning will be a lot more effective. However, this technique is sub-optimal. In addition, we will show in Section IV that computational gains are large even for $N = 10^3$.

## IV. Complexity Analysis & Results

The per state computational complexity of the VVA is given by

$$\begin{aligned} N_{add} &= 3N_t|\mathcal{A}|^{N_t} \\ N_{mult} &= N_t|\mathcal{A}|^{N_t} \\ N_{cmp} &= |\mathcal{A}|^{N_t} - 1, \end{aligned} \tag{17}$$

where $N_{add}$ and $N_{mult}$ represent the number of real additions and complex multiplications respectively. For $N_t \geq 2$, the per state computational complexity of full tree search, without tree pruning, is given by

$$\begin{aligned} N_{add} &\approx 4|\mathcal{A}|^{N_t} \\ N_{mult} &\approx |\mathcal{A}|^{N_t} \end{aligned} \tag{18}$$

The exact expressions and derivation of (19) and (18) can be found in Appendix A. Observe that we can still achieve computational gains even if we perform a naive tree search without any pruning. For example, in a $2 \times 2$ MIMO system with 16-QAM signal constellation, a full tree search algorithm saves $33\%$ of the real additions and $46\%$ of the complex multiplications when compared to VVA. More importantly, the full tree search algorithm has a fixed computational complexity. However, we can achieve larger gains by using the combined SD-VVA algorithm described in the previous section. Unlike VVA or full tree search, the combined SD-VVA algorithm has a random complexity that depends on the SNR and channel statistics. In order to quantify the average computational gains, we computed, via simulations, the average complexity of the combined VVA-SD algorithm and compared it to VVA for various settings. In our experiments, we chose a $2 \times 2$ MIMO system with 16-QAM signal constellation, $L = 3$, and $N = 10^3$. The results are summarized in Table I. As

TABLE I
SD-VVA vs. VVA

| **constellation** | $N_{add}$ | $N_{mult}$ |
|---|---|---|
| **16-QAM** (5 dB) | $43\%$ | $49\%$ |
| **16-QAM** (10 dB) | $53\%$ | $54\%$ |
| **16-QAM** (15 dB) | $64\%$ | $62\%$ |

discussed in Section II, the algorithm's performance improves with increasing SNR. For 16-QAM signal constellations, the computational complexity of the VVA is reduced by $50\%$ when the SNR is around 10 dB and by $60\%$ when the SNR is around 15 dB.

## V. Conclusion

Even though our approach provides substantial complexity gains, the number of states is still exponential in $N_t$ and $L$. Therefore, for large $N_t$ or $L$, performing exact MLSD might be expensive despite the reductions shown in the previous section. In this case, we can use a variety of techniques to further reduce the complexity. This, however, sacrifices optimality. For example, when $L$ is large, a linear channel shortening filter can be used to reshape the channel's impulse response such that most of the signal's energy is concentrated in the first few $L'$ taps, where $L' < L$. If $N_t$ is large, we can save a lot by keeping the best $K$ states (states with the least path metrics) at each stage instead of keeping track of all $|\mathcal{A}|^{N_t(L-1)}$ states. The choice of $K$ is determined by a reasonable performance-complexity tradeoff assessment.

The combined SD-VVA algorithm reduces the complexity of VVA while preserving its optimality. This algorithm is attractive whenever performance is not to be compromised. In LTE-A systems, the uplink can afford running expensive detection algorithms because the computations are taking place at the base station side. Moreover, the combined SD-VVA algorithm can be easily modified to output likelihoods, soft decisions, that are fed to the channel decoder. Future work will look at the architectural implementation and design of the combined SD-VVA algorithm.

## Appendix

For VVA, the following operation needs to be performed for each state

$$\mathcal{P}_k^i = \min_{j\in\mathcal{F}} \mathcal{P}_{k-1}^j + \|\mathbf{y}_k - \boldsymbol{\mu}_k\left(\mathcal{S}_k^j, \mathbf{x}^i[k]\right)\|^2,$$

where $\boldsymbol{\mu}_k\left(\mathcal{S}_k^j, \mathbf{x}^i[k]\right) = \sum_{l=0}^{L-2}\mathbf{H}[l]\mathbf{x}^i[k-l] + \mathbf{H}[L-1]\mathbf{x}^j[k-L+1]$ is precomputed for all $i$ and $j$. There are $|\mathcal{A}|^{N_t}$ incoming branches for each state. To compute each branch metric, $N_t$ complex additions, $N_t$ complex multiplications, and $N_t - 1$ real additions are needed. Each computed branch metric has to be added to its corresponding path metric. This requires an additional real addition. Finally, to perform the $\min$, $|\mathcal{A}|^{N_t} - 1$ comparisons are needed. Therefore, the per-state complexity of the VVA algorithm is given by

$$\begin{aligned} N_{add} &= 3N_t|\mathcal{A}|^{N_t} \\ N_{mult} &= N_t|\mathcal{A}|^{N_t} \\ N_{cmp} &= |\mathcal{A}|^{N_t} - 1. \end{aligned} \tag{19}$$

Here, $N_{add}$ refers to the total number of real additions, $N_{mult}$ refers to the total number of complex multiplications, and $N_{cmp}$ refers to the number of comparisons. We assume that every complex addition is equivalent to two real additions.

For the full tree search algorithm, the following operation needs to be performed for each state

$$\begin{aligned} \mathcal{P}_k^i &= \min_{j\in\mathcal{F}} \mathcal{P}_{k-1}^j + \|\mathbf{z}_k - \mathbf{G}\mathbf{x}^j\|^2 \\ &= \min_{j\in\mathcal{F}} \mathcal{P}_{k-1}^j + \|\tilde{\mathbf{z}}_k - \mathbf{R}\mathbf{x}^j\|^2, \end{aligned} \tag{20}$$

where $\mathbf{G} = \mathbf{Q}\left[\mathbf{R}^*\mathbf{0}_{N_t\times(N_r-N_t)}\right]^*$ by the QR decomposition and $\tilde{\mathbf{z}}_k$ corresponds to the first $N_t$ entries of $\mathbf{Q}^*\mathbf{z}_k$. We assume that $\sum_{l=0}^{L-2}\mathbf{H}[l]\mathbf{x}^i[k-l]$ and $\mathbf{R}\mathbf{x}^j$ are precomputed for all $i$ and $j$. First, to compute $\mathbf{z}_k = \mathbf{y}_k - \sum_{l=0}^{L-2}\mathbf{H}[l]\mathbf{x}^i[k-l]$, $N_r$ complex additions are needed. The result has to be multiplied by $\mathbf{Q}^*$ to obtain $\tilde{\mathbf{z}}_k$. This requires $(N_r-1)N_t$ complex additions and $N_rN_t$ complex multiplications. Next, we have to compute all the partial Euclidean distances. In a $|\mathcal{A}|$-ary tree of depth $N_t$, there are $\sum_{i=1}^{N_t}|\mathcal{A}|^i$ edges. Therefore, in order to compute the weights $e_l\left(x_l^j, ..., x_{N_t}^j\right)$ of all edges, we need $\sum_{i=1}^{N_t}|\mathcal{A}|^i$ complex additions and $\sum_{i=1}^{N_t}|\mathcal{A}|^i$ complex multiplications. After having computed the weights of all edges in the tree, we need to traverse the tree from the root node to every leaf node to add the weights of all edges to each other and then add the result to the path metric. This requires $\sum_{i=2}^{N_t}|\mathcal{A}|^i + |\mathcal{A}|^{N_t}$ real additions. The number of comparisons that are needed is identical to VVA. Therefore, the per-state complexity of full tree search is given by

$$\begin{aligned} N_{add} &= \sum_{i=2}^{N_t}|\mathcal{A}|^i + |\mathcal{A}|^{N_t} + 2\sum_{i=1}^{N_t}|\mathcal{A}|^i + 2(N_r-1)N_t + 2N_r \\ N_{mult} &= \sum_{i=1}^{N_t}|\mathcal{A}|^i + N_rN_t \\ N_{cmp} &= |\mathcal{A}|^{N_t} - 1. \end{aligned} \tag{21}$$

## References


[1] W. van Etten, "Maximum likelihood receiver for multiple channel transmission systems," *Communications, IEEE Transactions on*, vol. 24, no. 2, pp. 276 – 283, Feb. 1976.

[2] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268 – 278, March 1973.

[3] G. Stuber, J. Barry, S. McLaughlin, Y. Li, M. Ingram, and T. Pratt, "Broadband MIMO-OFDM wireless communications," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 271 – 294, Feb. 2004.

[4] S. Parkvall, E. Dahlman, A. Furuskar, Y. Jading, M. Olsson, S. Wanstedt, and K. Zangi, "LTE-Advanced - evolving LTE towards IMT-Advanced," in *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, Sept. 2008, pp. 1 –5.

[5] 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, 1985.

[6] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm II. generalizations, second-order statistics, and applications to communications," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2819 – 2834, Aug. 2005.

[7] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. expected complexity," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2806 – 2818, Aug. 2005.

[8] H. Vikalo and B. Hassibi, "Maximum-likelihood sequence detection of multiple antenna systems over dispersive channels via sphere decoding," *EURASIP J. Appl. Signal Process.*, vol. 2002, pp. 525–531, January 2002.