Likelihood-Based Tree Search for Low Complexity Detection in Large MIMO Systems

Saksham Agarwal, Abhay Kumar Sah, and A. K. Chaturvedi, Senior Member, IEEE

Abstract—A recently reported result on large/massive multiple-input multiple-output (MIMO) detection shows the utility of the branch and bound (BB)-based tree search approach for this problem. We can consider strong branching for improving upon this approach. However, that will require the solution of a large number of quadratic programs (QPs). We propose a likelihood based branching criteria to reduce the number of QPs required to be solved. We combine this branching criteria with a node selection strategy to achieve a better error performance than the reported BB approach, that too at a lower computational complexity. Simulation results show that the proposed algorithm outperforms the available detection algorithms for large MIMO systems.

Index Terms—Large MIMO, massive MIMO, branch and bound, integer programming.

I. INTRODUCTION

With the advancement towards the upcoming 5G systems [1], the number of antennas in multiple-input multiple-output (MIMO) systems will be scaled up [2] to achieve higher data rates. Such large antenna systems are popularly referred to as large/massive MIMO systems. The detection of signals in such systems becomes challenging. This is because the existing near maximum likelihood (ML) detectors for conventional MIMO systems like the tree search algorithms in [3] and [4] are infeasible for large/massive MIMO systems owing to their increasing computational complexity with the number of antennas. On the other hand, the performance of low complexity detectors like minimum mean square error (MMSE) deteriorates with increasing number of antennas.

In the literature, various approaches like [5]–[9] have been proposed to address the issue of large MIMO detection. One recent approach [9] has formulated this problem as a mixed integer quadratic programming (MIQP) problem [10] and solved it using the standard branch and bound (BB) algorithm [11]. It has been shown that the BB based approach can perform significantly better than other existing approaches.

BB is a tree search based algorithm and requires strategies for branching and node selection. Among the various branching techniques available, strong branching (SB) is known to provide the optimal solution while doing minimal tree exploration [12]. However, BB using SB comes at a huge computational cost of solving $2N+1$ quadratic programs (QPs) at every node ($N$ denotes the size of the problem) and turns out to be practically infeasible even for small MIMO systems. Out of the $2N+1$ QPs, only one is used towards obtaining the solution and the remaining $2N$ are required for deciding which variable to branch upon. If we can branch without solving QPs, the complexity of this approach can be drastically reduced.

In this letter, we address this issue by using a metric [13] for determining the likelihood of a symbol being in error. Thus, we propose to branch the tree at the index where the magnitude of the likelihood is maximum. As a result, at every node, instead of $2N$ QPs we need to solve only a single QP. We further propose a node selection strategy which explores the tree in a way such that a good performance can be obtained at a reasonable complexity. We verify this using simulations and show that this provides a better error performance than the recently reported BB based technique [9].

The rest of this letter is organized as follows. We describe the system model, problem formulation, and a review of BB algorithm in Section II. Section III proposes the likelihood based tree search algorithm and its complexity analysis has been discussed in Section IV. Simulation results are presented in Section V. Finally, we conclude this letter in Section VI.

We use ‘A’ to denote matrices, ‘a’ to denote vectors, and ‘a’ to denote scalar quantities.

II. PRELIMINARIES

We consider a MIMO system with $N_t$ transmit and $N_r$ receive antennas. The input-output relationship is given by

$$\tilde{y} = \tilde{H}\tilde{x} + \tilde{n},$$

where $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{N_r}]^T \in \mathbb{C}^{N_r \times 1}$ is the received signal vector and $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{N_t}]^T \in \mathbb{C}^{N_t \times 1}$ is the transmitted signal vector where each $\tilde{x}_n$ belongs to an M-QAM square constellation $\tilde{x}$. $H \in \mathbb{C}^{N_r \times N_t}$ denotes the channel gain matrix with $C(0,1)$ elements and $\tilde{n} \in \mathbb{C}^{N_r \times 1}$ denotes the complex additive white Gaussian noise (AWGN) vector with i.i.d. $C(0,\sigma^2)$ entries. This system model (1) is valid for both, single user point to point large MIMO systems, as well as multi user massive MIMO systems [14].

The system model in (1) can be expressed as an equivalent real-valued model

$$y = Hx + n,$$
where \( y = [g(\tilde{y})]^{T} \), \( \mathbf{x} = [g(\tilde{x})]^{T} \), \( \mathbf{n} = [g(\tilde{n})]^{T} \) and
\[
H = \begin{bmatrix}
\Re \{ \mathbf{H} \} & \Im \{ \mathbf{H} \} \\
\Re \{ \mathbf{H} \} & \Im \{ \mathbf{H} \}
\end{bmatrix}.
\]

The objective of the optimal (ML) detection problem is to find the transmitted vector \( \mathbf{x}^{*} \) which minimizes the Euclidean cost, that is
\[
x^{*} = \arg\min_{\mathbf{x} \in \mathcal{X}} \| \mathbf{y} - \mathbf{Hx} \|^{2},
\]
where \( \mathcal{X} \in \{-\sqrt{M} + 1, \ldots, -1, 1, \ldots, \sqrt{M} - 1\} \). As shown in [9], by using the linear transformation \( z_{i} = \frac{y_{i} + (\sqrt{M} - 1)H_{1}}{2} \) for generating reduced search spaces, as it does not create an incorrect solution vector. However, we do not know which symbols are in error. Fortunately, recently a metric which computes the likelihood of a symbol being in error, especially in the context of large MIMO systems, has been reported in [13]. In the MIQP problem (4), the likelihood of the error being in error can be expressed as [13]
\[
\eta_{i} = \frac{|q_{i} + q_{z}|}{\sqrt{q_{ii}}},
\]
where \( q_{ii} \) and \( q_{i} \) are the \((i, i)\) element and \(i\)th column of \( Q \), respectively, and \( g_{i} \) denotes the \(i\)th element of \( g \). Here \( \eta_{i} \) is the likelihood of \( z_{i} \) being in error, higher the value of \( \eta_{i} \) higher is the probability of \( z_{i} \) being in error. Let \( k \) be the index for which this likelihood metric is maximum. We branch upon the \(k\)th variable of a node \( S \) into two search spaces \( S_{k}^{(1)} \) and \( S_{k}^{(2)} \) as defined below
\[
S = \{ z : lb_{k} \leq z_{k} \leq ub_{k} \ \forall k \in \{1, 2, \ldots, 2N_{t}\} \},
\]
\[
S_{k}^{(1)} = S \cap \{ z : lb_{k} \leq z_{k} \leq \lfloor z_{k} \rfloor \},
\]
\[
S_{k}^{(2)} = S \cap \{ z : \lceil z_{k} \rceil \leq z_{k} \leq ub_{k} \}.
\]

### III. Likelihood Based Tree Search Algorithm

We can solve the problem with low complexity if we can avoid solving such high number of QPs for branching, while still retaining a reasonable error performance.

#### A. Error Likelihood Metric Based Variable Branching

Ideally, for the large MIMO detection problem, branching at a node should take place at the symbols which are in error. This would exclude the search space which would lead to an incorrect solution vector. However, we do not know which symbols are in error. Fortunately, recently a metric which computes the likelihood of a symbol being in error, especially in the context of large MIMO systems, has been reported in [13]. In the MIQP problem (4), the likelihood of the \(i\)th symbol being in error can be expressed as [13]
\[
\eta_{i} = \frac{|q_{i} + q_{z}|}{\sqrt{q_{ii}}},
\]
where \( q_{ii} \) and \( q_{i} \) are the \((i, i)\) element and \(i\)th column of \( Q \), respectively, and \( g_{i} \) denotes the \(i\)th element of \( g \). Here \( \eta_{i} \) is the likelihood of \( z_{i} \) being in error, higher the value of \( \eta_{i} \) higher is the probability of \( z_{i} \) being in error. Let \( k \) be the index for which this likelihood metric is maximum. We branch upon the \(k\)th variable of a node \( S \) into two search spaces \( S_{k}^{(1)} \) and \( S_{k}^{(2)} \) as defined below
\[
S = \{ z : lb_{k} \leq z_{k} \leq ub_{k} \ \forall k \in \{1, 2, \ldots, 2N_{t}\} \},
\]
\[
S_{k}^{(1)} = S \cap \{ z : lb_{k} \leq z_{k} \leq \lfloor z_{k} \rfloor \},
\]
\[
S_{k}^{(2)} = S \cap \{ z : \lceil z_{k} \rceil \leq z_{k} \leq ub_{k} \}.
\]

### B. Node Selection Strategy

Let us define a tree with depth \(d\) and breadth \(b\) as a \((d, b)\) tree. For chosen values of \(d\) and \(b\), the nodes having depth less than \(d\) are allowed to branch further in the tree. Building upon the idea in the previous subsection, we propose to branch upon \(b\) indices with the \(b\) largest values of the likelihood metric. Thus, during branching at any node, we create \(2b\) new nodes with corresponding search spaces \( S_{k}^{(1)} \) and \( S_{k}^{(2)} \) with \(k\) being the indices corresponding to the \(b\) largest value of the likelihood metric. We would like to search this \((d, b)\) tree such that after exploring the tree, we are likely to have the best possible solution within the tree. In view of this, we propose the following node selection and tree search strategy.

Assume at any depth \(i\), there exist a set of candidate nodes \( \{P_{1}, P_{2}, \ldots, P_{i}\} \), where subscript denotes the depth and superscript is used to index the candidate nodes. One of these nodes will be selected to branch further. We choose the node at which the corresponding solution vector has the least value of objective function (4) and denote it as \(P_{a}^{*}\). This can
Algorithm 1 Likelihood Based Tree Search

1: Parameters: $d$, $b$
2: Input: $Q$, $g$
3: Initialization: $List = \{P_0^i\}$, $i := 0$, $f^* := \infty$
4: while $List$ is not empty do
5:   Find $P_i^i \in List$ with minimum $f(z_{P_i^i})$ using (7)
6:   Keep $P_i^i$ and prune all other nodes
7:   if $f(z_{P_i^i}) < f^*$ then
8:      $f^* := f(z_{P_i^i})$
9:      $z^* := z_{P_i^i}$
10: end if
11: if node $P_i^i$ is at depth $d$ then
12:   Delete $P_i^i$
13:   continue
14: end if
15: Compute $\eta_i := \frac{|\gamma + Q_i x|}{\sqrt{2}} \forall i \in \{1, 2, \ldots, 2N_t\}$
16: Sort $\eta_i$’s in descending order and rank corresponding indices
17: for $j := 1:b$ do
18:   Choose index with rank $j$
19:   Branch $P_i^j$ at the chosen index
20:   Store the newly created nodes as $P_{i+1}^{2j-1}$ and $P_{i+1}^{2j}$
21:   Push $P_{i+1}^{2j-1}$ and $P_{i+1}^{2j}$ to $List$
22: end for
23: Delete $P_i^*$
24: $i := i + 1$
25: end while
26: return $z^*$

be mathematically expressed as

$$P_i^i = \arg\min_{P_i^i} f(z_{P_i^i}),$$

where $f(r) = \frac{1}{2}r^TQr + g^Tr$ and $z_{P_i^i}$ is the solution vector of node $P_i^i$. It may be noted that $z_{P_i^i}$ is obtained by rounding off the elements of the solution of the relaxed problem (5).

For generating the set of candidate nodes at $(i+1)$th depth, we prune the existing nodes from the $(d, b)$ tree except the node $P_i^i$, on which we further branch upon. After exploring the tree till depth $d$, the solution vector $z^*$ is given by

$$z^* = \arg\min_{\{z_k \mid k \in P_0^*, \ldots, P_d^*\}} f(z_k).$$

Combining all the above, we summarize the pseudo-code in Algorithm 1, where $z^*$ returns the desired solution. In the pseudo code $P_0^*$ represents the lone root node (initial candidate node) in the tree, which corresponds to the problem in (5). We refer to this as the likelihood based tree search (LBTS) algorithm. For a given $(d, b)$ tree, we believe that this tree search strategy can give a good error performance at a reasonable complexity, for large MIMO systems. This has been corroborated using simulations in Section V.

IV. Complexity Analysis of LBTS

The computational complexity of LBTS depends upon the number of QPs required to be solved. We consider the interior point QP solver [15] which has $O(N_t^3)$ complexity per iteration. The advantage of using an interior point solver is that the number of iterations, for a given QAM size, is independent of $N_t$ [16]. Thus, the overall complexity at a given node is of the same the order, i.e., $O(N_t^3)$. Out of approximately $(2b)^d$ nodes in a $(d, b)$ tree, $d \times 2b$ nodes are explored and since only one QP is required to be solved at every node, the overall complexity is $O(dbN_t^3)$.

V. Simulation Results

In this section, we study the performance of the proposed LBTS algorithm for large MIMO systems of different sizes, considering a Rayleigh fading channel. We first evaluate the effect of $d$ and $b$ on the bit error rate (BER) performance. For this, we consider a $32 \times 32$ (i.e., $N_t = N_r = 32$) 16-QAM system and present the results for $E_b/N_0 = 15$ dB in Fig. 1(a). From the figure, it can be viewed that the performance saturates with increasing values of $d$ and $b$. However, complexity increases linearly with $d \times b$ (explained in Section IV). In the sequel, we choose $d = 3$ and $b = 3$ as this seems to provide a reasonable trade-off between performance and complexity.

Next, we compare the performance for LBTS $(d, b)$ with some recent algorithms for large MIMO detection. Thus, we choose ULAS [17], MLAS [5], and RTS [6], along with BB $(l, m)$ [9] where $l$ and $m$ are simulation parameters. The results have been shown for a $64 \times 64$ 16-QAM system and
a $32 \times 32$ 64-QAM system in Fig. 1(b) and Fig. 1(c), respectively. To start with, we choose the value of $(l,m)$ such that the number of QPs required by BB $(l,m)$ are same as for LBTS with $d=3$ and $b=3$. In other words, both have similar complexity. Hence, we choose the value of $(l,m)$ as $(5,2)$. We prefer this over other possible pairs like $(3,3)$ and $(4,2)$ having similar complexities, owing to its superior BER performance. From the figures one can observe that LBTS $(3,3)$ outperforms all of the other algorithms and BB is its best competitor. At a BER of $10^{-3}$, compared to BB $(5,2)$, LBTS $(3,3)$ has a gain of around $2.5$ dB and $3.5$ dB for $64 \times 64$ 16-QAM system and $32 \times 32$ 64-QAM system, respectively. Next, we allow BB $(l,m)$ to explore more number of nodes (i.e., to solve more number of QPs) by selecting $l=36$ and $m=2$. It may be noted that the BER performance of BB $(l,m)$ is more sensitive to $l$ than $m$, and hence increasing the value of $l$ has more payoff [9]. Even after allowing to solve a significantly higher number of QPs, i.e., 144 compared to 18 for LBTS $(3,3)$, the latter has still a performance gain of $1.5$ dB, at the same BER of $10^{-3}$.

Further, we examine the error performance for increasing number of antennas and show the corresponding BER curves in Fig. 2. From the figure, we can see that LBTS outperforms the rest of the algorithms by a large margin.

Lastly, in Table I, we compare the number of arithmetic operations for a $64 \times 64$ 16-QAM system and a $32 \times 32$ 64-QAM system. It can be observed that LBTS $(3,3)$ requires lowest number of arithmetic operations after ULAS. As an illustration, for a $32 \times 32$ 64-QAM system at $E_b/N_0=18$ dB, the savings in complexities are nearly $23\%$, $90\%$, $86\%$, and $98\%$ compared to BB $(5,2)$, BB $(36,2)$, MLAS, and RTS, respectively. Only ULAS has a lower complexity, however it has an inferior BER performance (see Fig. 1(b) and Fig. 1(c)). It may be noted that the complexities of LBTS $(3,3)$, BB $(5,2)$, and BB $(36,2)$ are directly proportional to the number of QPs solved viz. 18, 20, and 144 respectively.

VI. CONCLUSION

We have considered an MIQP based approach to propose LBTS - a low complexity tree search algorithm for detection in large MIMO systems. The algorithm is based on a novel branching strategy using a recently proposed error likelihood metric which helps in searching the tree more efficiently. We combine it with a node selection strategy to obtain a good trade-off between performance and complexity. Simulation results show that LBTS provides a better error performance, at a lower complexity, compared to existing algorithms for large MIMO systems.

### TABLE I

**Comparison Based on Arithmetic Operations ($\times 10^5$) Per Bit**

<table>
<thead>
<tr>
<th>Detection Algorithms</th>
<th>$32 \times 32$ 64-QAM</th>
<th>$64 \times 64$ 16-QAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLAS</td>
<td>34.237</td>
<td>37.250</td>
</tr>
<tr>
<td>RTS</td>
<td>560.104</td>
<td>292.026</td>
</tr>
<tr>
<td>ULAS</td>
<td>0.213</td>
<td>0.199</td>
</tr>
<tr>
<td>BB $(5,2)$</td>
<td>6.852</td>
<td>6.335</td>
</tr>
<tr>
<td>BB $(36,2)$</td>
<td>49.580</td>
<td>50.667</td>
</tr>
<tr>
<td>LBTS $(3,3)$</td>
<td>4.725</td>
<td>4.868</td>
</tr>
</tbody>
</table>

**References**


