Statistical Pruning for Near Maximum Likelihood Detection of MIMO Systems

Tao Cui, Tracey Ho
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA, USA 91125
Email: {taocui, tho}@caltech.edu

Chintha Tellambura
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB, Canada T6G 2V4
Email: chintha@ece.ualberta.ca

Abstract— We show a statistical pruning approach for maximum likelihood (ML) detection of multiple-input multiple-output (MIMO) systems. We present a general pruning strategy for sphere decoder (SD), which can also be applied to any tree search algorithms. Our pruning rules are effective especially for the case when SD has high complexity. Three specific pruning rules are given and discussed. From analyzing the union bound on the symbol error probability, we show that the diversity order of the deterministic pruning is only one by fixing the pruning probability. By choosing different pruning probability distribution functions, the statistical pruning can achieve arbitrary diversity orders and SNR gains. Our statistical pruning strategy thus achieves a flexible trade-off between complexity and performance.

I. INTRODUCTION

Multiple-input multiple-output (MIMO) systems over a rich scattering wireless channel are capable of providing enormous capacity improvements without increasing the bandwidth or transmit power. In such MIMO systems, maximum-likelihood (ML) detection minimizes the overall error probability. However, the ML detection complexity increases exponentially in the number of transmit antennas. The zero-forcing (ZF) decision feedback detector (DFD) with optimal ordering, also known as the V-BLAST detector [1], performs poorly compared to the ML detector (MLD). The large gap in performance between ML and suboptimal detectors has motivated alternative detectors. In [2], the sphere decoder (SD) is proposed as an optimal (ML) detection method, which has low complexity in high SNR. However, recent results show that its average time complexity is exponential in the number of transmit antennas [3]. The SD complexity grows when the SNR is low, or when the number of transmit antennas is large.

The SD can be viewed as a tree search algorithm based on depth-first branch-and-bound (DFBnB) search [4]. The SD attempts to prune the nodes that are not on the optimal path; consequently the SD is an optimal algorithm. But the number of nodes pruned may not always high enough to significantly reduce the complexity. In this paper, we augment the SD with statistical pruning to reduce the complexity. Our main idea is to heuristically prune nodes on the search tree, discarding the nodes that do not seem to lead to high-quality solutions

from further examination. In [5], a statistical pruning algorithm called the increasing radii algorithm (IRA) is proposed by choosing an increasing radius sequence in the SD. Fixing the radius sequence in IRA, there exists a probability that no point lies within the reduced space. The IRA needs to be restarted again until a feasible point is found, which brings additional complexity. In addition, the IRA cannot attain different diversity orders and achieve a flexible performance-complexity tradeoff. We therefore propose a flexible framework for MIMO detection utilizing a statistical point of view. We present a general statistical pruning strategy, which includes IRA as a special case. We discuss three specific pruning rules, which are fail-proof and avoid restarting as IRA does. We also analyze the effect of the pruning probability on the performance. By choosing different pruning probability distribution functions according to the signal-to-noise ratio (SNR), statistical pruning can achieve arbitrary diversity orders and SNR gains. Our statistical pruning approach provides a flexible tradeoff between complexity and performance.

II. SYSTEM MODEL

We consider a spatial multiplexing MIMO system with n transmit antennas and m receive antennas. Source data are mapped into complex symbols from a finite constellation $\mathcal{Q}$. We assume a rich scattering memoryless (flat fading) channel. The received signals may be written as

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

where $\mathbf{x} = [x_1, \ldots, x_n]^T$, $x_i \in \mathcal{Q}$ is the transmitted signal vector, $\mathbf{y} = [y_1, \ldots, y_m]^T$, $y_i \in \mathcal{C}^n$ is the received signal vector, $\mathbf{H} = [h_{i,j}] \in \mathcal{C}^{m \times n}$ is the channel matrix, and $\mathbf{n} = [n_1, \ldots, n_m]^T$, $n_i \in \mathcal{C}^m$ is an additive white Gaussian noise (AWGN) vector. The elements of $\mathbf{H}$ are independent identically distributed (i.i.d.) complex Gaussian, $h_{i,j} \sim \mathcal{CN}(0,1)$. The components of $\mathbf{n}$ are i.i.d. and $n_i \sim \mathcal{CN}(0, \sigma_n^2)$. Throughout this paper, we assume that the channel is perfectly known to the receiver and $n \leq m$. If $n > m$, we can readily convert the rank deficient system into a full rank system with $n = m$ using the method in [6] and our proposed algorithm can also be applied to the resulting system. The MLD is given by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathcal{Q}^n} \|\mathbf{r} - \mathbf{H}\mathbf{x}\|^2. \qquad (2)$$

If $\mathcal{Q}$ is a subset of the integer set $\mathcal{Z}$, (2) is known as the closest vector problem (CVP) in lattice theory, known to be NP-hard. Exhaustive search for (2) has a complexity exponential in n.

We first transform (2). Column reordering can be applied to $\mathbf{H}$ by using V-BLAST or other ordering schemes and the resulting matrix is $\mathbf{G} = \mathbf{H}\mathbf{\Pi}$, where $\mathbf{\Pi}$ is the column permutation matrix. Let the QR factorization of $\mathbf{G}$ be $\mathbf{G} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ where $\mathbf{R} = [r_{i,j}]$ is an $n \times n$ upper-triangular matrix, $\mathbf{0}$ is an $(m-n) \times n$ zero matrix, $\mathbf{Q}_1$ is an $m \times n$ unitary matrix and $\mathbf{Q}_2$ is an $m \times (m-n)$ unitary matrix. Eq. (2) is equivalent to

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{Q}^n} \|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \tag{3}$$

where $\mathbf{y} = \mathbf{Q}_1^H \mathbf{r}$. Note that Eq. (3) is the basis for the SD. In the following, without loss of generality, we consider (3) and $\mathcal{Q}$ to be real, i.e., pulse amplitude modulation (PAM). If $\mathcal{Q}$ is complex, we can decouple (1) into a real system. The SD or the modified SD in [7] can be applied to the resulting system. The details are omitted for brevity.

III. SPHERE DECODING ALGORITHM

Here, using (3), we briefly explain the basic SD. When $\mathbf{x}$ is an integer vector, the set generated by $\mathbf{R}\mathbf{x}$ is called an integer lattice with $\mathbf{R}$ being the generator matrix of the lattice. The key idea of the SD is to process only the lattice points that are sufficiently 'close' to the received signal vector. This is ensured by generating all the lattice points inside a hypersphere centered around $\mathbf{y}$. The lattice point $\mathbf{R}\mathbf{x}$ lies in a sphere of radius d if, and only if

$$\|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \le d^2. \tag{4}$$

When the left-hand side of the above inequality is expanded, the first term depends only on x_n, the second term on x_n, x_{n-1} and so forth. Therefore, a necessary condition for the lattice point $\mathbf{R}\mathbf{x}$ to lie inside the sphere is that $d^2 \ge (y_n - r_{n,n}x_n)^2$, which is equivalent to x_n belonging to the interval

$$\left\lceil \frac{-d + y_n}{r_{n,n}} \right\rceil \le x_n \le \left\lfloor \frac{d + y_n}{r_{n,n}} \right\rfloor \tag{5}$$

where $\lceil \cdot \rceil$ denotes the smallest integer greater than or equal to its argument and $\lfloor \cdot \rfloor$ denotes the largest integer less than or equal to its argument. For each candidate x_n satisfying the above bound, we define $d_{n-1}^2 = d^2 - (y_n - r_{n,n}x_n)^2$. We can get a stronger necessary condition for x_{n-1}. We continue in the same process for x_{n-2} and so on. In general, the bounds for x_k become

$$LB_k \le x_k \le UB_k \tag{6}$$

where

$$LB_k = \left\lceil \frac{-d_k + y_k - \sum_{j=k+1}^{n} r_{k,j}x_j}{r_{k,k}} \right\rceil, \tag{7}$$

$$UB_k = \left\lfloor \frac{d_k + y_k - \sum_{j=k+1}^{n} r_{k,j}x_j}{r_{k,k}} \right\rfloor, \tag{8}$$

and $d_k^2 = d_{k+1}^2 - \left(y_{k+1} - \sum_{j=k+1}^{n} r_{k+1,j}x_j \right)^2$. When the SD finds a candidate lattice point $\mathbf{x}'$ satisfying (4), the radius d is reduced. This continues until no further lattice points are

found within the hypersphere. The lattice point achieves the smallest value of $\|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2$ is deemed as the ML solution. If no point in the sphere is found, the sphere is empty and the search fails. In this case, the initial search radius d must be increased and the search restarted.

IV. STATISTICAL PRUNING

The main idea of statistical pruning is to prune (based on heuristics) the nodes that do not seem to lead to the ML solution. Statistical pruning is performed in such a way that a near ML solution, and hopefully an ML solution, can be found with greatly reduced complexity.

We give the statistical pruning procedure embedded in SD for MIMO detection in Algorithm 1, where k denotes the current level in the search tree, $\mathcal{H}_k$ the set of heuristic rules for node pruning in the k-th level, d_k^2 is the current upper bound, and LB_k, UB_k are defined in (7) and (8). $\mathbf{x}_{min}$ returns the optimal solution. The algorithm is invoked as Prune$(n, \mathcal{H}_{k-1}, d_n^2, y_n)$, where $d_n^2 = d^2$ or $d_n^2 = +\infty$, and d is the initial radius.

Prune: $(k, \mathcal{H}_k, d_k, \rho_k)$

1 Generate all children in the k-th level; the candidate set $\mathcal{A} = [LB_k, LB_k + 1, \ldots, UB_k] \cap \mathcal{Q}$;
2 Sort $\mathcal{A}$ in increasing order of cost $c_i = (\rho_k - r_{k,k}a_i)^2$;
3 Discard a node in $\mathcal{A}$ if it can be pruned by a rule in $\mathcal{H}_k$, and $\mathcal{A}$ has l nodes left;
4 **for** $i \leftarrow 1$ **to** l **do**
5      $x_k = s_i$;
6   **if** $c_i < d_k^2$ **then**
7   **if** $k = 1$ **then**
8              $\mathbf{x}_{min} = \mathbf{x}$;
9              $d_n^2 = \|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2$;
10   **else**
11              $\rho_{k-1} = y_{k-1} - \sum_{j=k}^{n} r_{k-1,j}x_j$;
12   Prune$(k - 1, \mathcal{H}_{k-1}, d_k^2 - c_i, \rho_{k-1})$
13   **end**
14   **end**
15 **end**

Algorithm 1: Sphere decoder with statistical pruning

Note that the SD with statistical pruning in Algorithm 1 is the same as Schnorr and Euchner SD variant [8] except that it may abandon a node based on heuristic rules $\mathcal{H}_k$. This simple but critical difference makes the SD with statistical pruning terminate sooner than the original SD. The set $\mathcal{A}$ is sorted in line 2 because the nodes with smaller are c_i more likely to lead to high-quality solutions. In Algorithm 1, different heuristic rules $\mathcal{H}_k$ are employed on different levels. In the first few levels, the pruning rules should be strong since the bound d_k^2 is loose, while the pruning rules can be weak in the last few levels when the bound d_k^2 is tight.

The SD with statistical pruning may fail to find a valid solution because the heuristic pruning rules are so strong that at a certain level all the children nodes are discarded. In order to increase the probability that a valid node can be found,

we can keep at least one child instead of abandoning all the children of a node. Intuitively, the modified rule chooses to keep the child node with the minimum cost among all the children. If all the other children nodes are pruned, the SD with statistical pruning returns the V-BLAST solution. Therefore, the modified pruning strategy at least achieves the performance of V-BLAST, and the SD with statistical pruning does not need to be run again, which saves complexity.

Let p_k denote a probability associated with the k-th level and t is the size of $\mathcal{A}$ before pruning. The candidate set $\mathcal{A}$ has been sorted. We give a specific pruning rule in the following:

Pruning Rule 1 ($\mathcal{H}_k^1$): The probability that the child a_i is not pruned is $f(i, p_k)$, and $f(i, p_k)$ is a nonincreasing function in i with $f(1, p_k) = 1$ and $0 \leq f(i, p_k) \leq 1$.

The pruning rule incorporates the heuristic rule that at least one child is kept by setting $f(1, p_k) = 1$. $f(i, p_k)$ chosen as a nonincreasing function in i, because, intuitively, a child with a smaller cost may more likely lead to the optimal solution. The children can be pruned independently or dependently. We have two specific cases for the pruning rule.

Case I: $f(i, p_k) = p_k^{i-1}$.

To achieve the $f(i, p_k)$ in this case, we prune the children dependently. For example, if child a_i ($i \geq 2$) is pruned, all the remaining $t - i$ children $a_{i+1}, \ldots, a_t$ are pruned, which agrees with the distribution $f(i, p_k) = p_k^{i-1}$. We call it *dependent pruning rule*. Note that there are other ways to achieve the same probability $f(i, p_k)$. However, the way we have introduced is easy to be implemented.

Case II: $f(i, p_k) = p_k$, for $t \geq i \geq 2$, and $f(i, p_k) = 1$, for $i = 1$.

The children are pruned independently in this case. All the children except the first one are pruned with equal probability. This pruning rule is also rational since in general, we do not have priori information on which child leads to the optimal solution and should be pruned, or we can identify the optimal solution immediately. We call it *random pruning rule*. Since $f(i, p_k)$ in Case I is less than that in Case II, the dependent pruning rule is stronger than the random pruning rule.

In both cases, we control the strength of pruning through p_k. For example, if $p_k = 0$, the statistical pruning returns the solution of V-BLAST if **H** is reordered with the V-BLAST ordering, while it simply reduces to the SD if $p_k = 1$. p_k is also a nondecreasing function in k since we need strong pruning rules in the first few stages of the tree search, and the pruning rules can be weaken with the decreasing of k.

Pruning rule 1 is defined from a probabilistic point of view. A natural way may be defining the pruning rule based on the cost c_i associated with each child directly. From intuition, if a child node has a significantly large cost increase from its parent, it is more likely that the child may lead to a solution of a large cost. We associate a threshold δ_k with the k-th level. The second pruning rule is:

Pruning Rule 2 ($\mathcal{H}_k^2$): a_1 is always not pruned. For $i = 2, \ldots, t$, if the cost c_i of a child node a_i is larger than δ_k, a_i is pruned.

The child node a_1 is kept to make sure that the pruning rule always returns a valid solution. Since $\mathcal{A}$ is in nondecreasing order of cost, if node a_i is pruned, all the children a_j for $j \geq i$ are pruned. δ_k also controls the strength of pruning. A smaller δ_k represents a stronger pruning rule, and a greater δ_k represents a weaker rule. For example, if $\delta_k = 0$, the statistical pruning returns the solution of V-BLAST provided that **H** is reordered with the V-BLAST ordering. If $\delta_k = +\infty$, it simply reduces to the SD. We call this pruning rule *cost pruning rule*.

We will show how to choose p_k and δ_k to achieve a flexible performance-complexity tradeoff in the next section. Pruning rules 1 and 2 can also be combined to possibly take advantage of both the cost intuition and probabilistic pruning. Let the minimum cost in the k-th level be c_m. The third pruning rule is:

Pruning Rule 3 ($\mathcal{H}_k^3$): The probability that the child a_i is not pruned is $f(c_i, p_k)$, and $f(c_i, p_k)$ is a nonincreasing function in c_i with $f(c_m, p_k) = 1$ and $0 \leq f(c_i, p_k) \leq 1$.

Examples of $f(c_i, p_k)$ are $f(c_i, p_k) = p_k^{c_i - c_m}$, $f(c_i, p_k) = \max\{1 - \frac{c_i - c_m}{c_m}(1 - p_k), 0\}$. We call this pruning rule *hybrid pruning rule*.

Clearly, with the same p_k, case I is a stronger pruning rule than case II. For simplicity, in the following, we assume that $p_n =, \ldots, = p_1 = p$ and $n = m$, and only analyze the performance of case II. The performance of case II provides a lower bound on that of case I.

Next, we analyze the performance of random pruning rule and cost pruning rule. δ_k and p_k are determined to achieve different diversity orders and performance gains. To make the analysis tractable, reordering is not applied to **H**.

We first consider the pruning rule 1. For simplicity, we assume that $n = m$ and $p_n =, \ldots, = p_1 = p$ temporarily. Only the performance of random pruning is analyzed. We show next that if p is kept a constant at all SNRs or deterministic pruning, the random pruning can only achieve a diversity order one. We derive the upper bound for the frame error rate (FER), P_f. Let $\mathbf{x}^{(1)} = \left[x_1^{(1)}, \ldots, x_n^{(1)} \right]^T$ denote the transmitted vector and $\hat{\mathbf{x}} = [\hat{x}_1, \ldots, \hat{x}_n]^T$ denote the vector returned by the statistical pruning detector. We have $P_f = Pr(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)})$. Denote A as the event that $\mathbf{x}^{(1)}$ is not pruned. By the total probability theorem [9], we have

$$\begin{aligned} P_f &= Pr(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A^c) Pr(A^c) + Pr\left(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A\right) Pr(A) \\ &= Pr(A^c) + Pr\left(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A\right) Pr(A), \end{aligned} \quad (9)$$

where $Pr(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A^c) = 1$. We first derive $Pr(A^c)$ (or $1 - Pr(A)$) and then analyze the second term of (9). Let $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_n]^T$ be the temporary value for $\mathbf{x} = [x_1, \ldots, x_n]^T$ during the statistical pruning search as in Algorithm I, which corresponds to the path from the root to the leaf node corresponding to $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_n]^T$. Denote A_i the event that $x_i^{(1)}$ is not pruned. Note that $Pr(A) = Pr(\tilde{\mathbf{x}} = \mathbf{x}^{(1)})$, which is the probability that the leaf node corresponding to $\mathbf{x}^{(1)}$ is visited

during the search. We then get

$$
\begin{aligned}
Pr(A) &= Pr(\tilde{\mathbf{x}} = \mathbf{x}^{(1)}) \\
&= Pr(\tilde{\mathbf{x}} = \mathbf{x}^{(1)} | \tilde{x}_n = x_n^{(1)}) Pr(\tilde{x}_n = x_n^{(1)}) \\
&\quad + Pr(\tilde{\mathbf{x}} = \mathbf{x}^{(1)} | \tilde{x}_n \neq x_n^{(1)}) Pr(\tilde{x}_n \neq x_n^{(1)}) \\
&= Pr(\tilde{\mathbf{x}} = \mathbf{x}^{(1)} | \tilde{x}_n = x_n^{(1)}) Pr(A_n),
\end{aligned}
\tag{10}
$$

where $Pr(\tilde{x}_n = x_n^{(1)}) = Pr(A_n)$, and $Pr(\tilde{\mathbf{x}} = \mathbf{x}^{(1)} | \tilde{x}_n \neq x_n^{(1)}) = 0$. By similar argument, we can expand (10) as

$$
Pr(A) = Pr(A_n) \prod_{i=1}^{n-1} Pr(A_i | \tilde{x}_{i+1} = x_{i+1}^{(1)}, \ldots, \tilde{x}_n = x_n^{(1)}). \tag{11}
$$

Let B_i denote the event that $x_i^{(1)}$ is not the first element of $\mathcal{A}$ in Algorithm I. We have

$$
\begin{aligned}
Pr(A_n^c) &= Pr(A_n^c | B_n) Pr(B_n) + Pr(A_n^c | B_n^c) Pr(B_n^c) \\
&= (1 - p) Pr(B_n),
\end{aligned}
\tag{12}
$$

where $Pr(A_n^c | B_n) = 1 - p$, and $Pr(A_n^c | B_n^c) = 0$. The union bound for $Pr(B_n)$ is given by

$$
\begin{aligned}
Pr(B_n) \leq \underset{r_{n,n}}{E} \underset{x_n^{(1)}}{E} &\left[\sum_{x_n^{(2)} \neq x_n^{(1)}} Pr \left(\left| y_n - r_{n,n} x_n^{(2)} \right|^2 \right. \right. \\
&\left. \left. < \left| y_n - r_{n,n} x_n^{(1)} \right|^2 \middle| x_n^{(1)}, r_{n,n} \right) \right],
\end{aligned}
\tag{13}
$$

where $x_n^{(2)}$ is the nearest neighbor of $x_n^{(1)}$. From [9], the squared norm of the entries of upper-triangular matrix $\mathbf{R}$ have χ^2 distribution with different degrees of freedom without column reordering, specifically, $|r_{i,i}|^2 \sim \chi^2(2i)$, for $i = 1, \ldots, n$ and $|r_{i,j}|^2 \sim \chi^2(2)$, for $j > i$, where $\chi^2(k)$ denotes the chi-squared distribution with k degrees of freedom. We can obtain

$$
\begin{aligned}
&Pr \left(\left| y_n - r_{n,n} x_n^{(2)} \right|^2 < \left| y_n - r_{n,n} x_n^{(1)} \right|^2 \middle| x_n^{(1)}, r_{n,n} \right) \\
&= Q \left(\sqrt{\left| r_{n,n} \left(x_n^{(2)} - x_n^{(1)} \right) \right|^2 / 2\sigma^2} \right),
\end{aligned}
\tag{14}
$$

where $Q(\cdot)$ is the Q-function. Using the Chernoff bound for the Q-function, $Pr(B_n)$ can be bounded as

$$
\begin{aligned}
Pr(B_n) &\leq \underset{r_{n,n}}{E} \underset{x_n^{(1)}}{E} \left[\sum_{x_n^{(2)} \neq x_1^{(1)}} \exp \left(\frac{-r_{n,n}^2 \left| x_n^{(2)} - x_n^{(1)} \right|^2}{4\sigma^2} \right) \right] \\
&= \underset{x_n^{(1)}}{E} \sum_{x_n^{(2)} \neq x_1^{(1)}} \frac{1}{1 + \frac{\left| x_n^{(2)} - x_n^{(1)} \right|^2}{4\sigma^2}} \\
&\leq \frac{|\mathcal{Q}|}{1 + d_{\min}^2 / 4\sigma^2},
\end{aligned}
\tag{15}
$$

where $d_{\min}$ is the minimum Euclidean distance of $\mathcal{Q}$. Therefore, we can bound $Pr(A_n^c)$ as

$$
Pr(A_n^c) \leq (1 - p) \frac{|\mathcal{Q}|}{1 + d_{\min}^2 / 4\sigma^2}. \tag{16}
$$

Similarly, we can have

$$
\begin{aligned}
&Pr(A_i | \tilde{x}_{i+1} = x_{i+1}^{(1)}, \ldots, \tilde{x}_n = x_n^{(1)}) \\
&\leq (1 - p) \frac{|\mathcal{Q}|}{(1 + d_{\min}^2 / 4\sigma^2)^{n-i+1}}, \quad i = 1, \ldots, n - 1.
\end{aligned}
\tag{17}
$$

Finally, $Pr(A)$ (or $1 - Pr(A^c)$) is obtained as

$$
\begin{aligned}
&Pr(A) \\
&= Pr(A_n) \prod_{i=1}^{n-1} Pr(A_i | \tilde{x}_{i+1} = x_{i+1}^{(1)}, \ldots, \tilde{x}_n = x_n^{(1)}) \\
&= \prod_{i=1}^{n-1} \left(1 - Pr(A_i^c | \tilde{x}_{i+1} = x_{i+1}^{(1)}, \ldots, \tilde{x}_n = x_n^{(1)}) \right) \\
&\qquad \times (1 - Pr(A_n^c)).
\end{aligned}
\tag{18}
$$

In high SNR, $Pr(A_n^c)$ and $Pr(A_i^c | \tilde{x}_{i+1} = x_{i+1}^{(1)}, \ldots, \tilde{x}_n = x_n^{(1)})$ are small. $Pr(A^c)$ can be well approximated as

$$
\begin{aligned}
Pr(A^c) &\approx Pr(A_n^c) \\
&\quad + \sum_{i=1}^{n-1} Pr(A_i^c | \tilde{x}_{i+1} = x_{i+1}^{(1)}, \ldots, \tilde{x}_n = x_n^{(1)}) \\
&\leq (1 - p) \sum_{i=1}^{n} \frac{|\mathcal{Q}|}{(1 + d_{\min}^2 / 4\sigma^2)^i}.
\end{aligned}
\tag{19}
$$

We then bound $Pr\left(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A\right)$ in (9) in the following. Denote the set of all the visited leaf nodes by $\mathcal{I}$, which is the candidate set for the output of the statistical pruning detection. Since some leaf nodes may be pruned, $|\mathcal{I}| \leq |\mathcal{Q}|^n$. In case of A, $\mathbf{x}^{(1)} \in \mathcal{I}$. The union bound for $Pr\left(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A\right)$ is

$$
\begin{aligned}
&Pr\left(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A\right) \\
&\leq \frac{1}{|\mathcal{Q}|^n} \sum_{\mathbf{x}^{(1)} \in \mathcal{Q}^n} \sum_{\substack{\mathbf{x}^{(2)} \in \mathcal{I}, \\ \mathbf{x}^{(2)} \neq \mathbf{x}^{(1)}}} Pr \left(\left\| \mathbf{y} - \mathbf{R}\mathbf{x}^{(2)} \right\|^2 \leq \left\| \mathbf{y} - \mathbf{R}\mathbf{x}^{(1)} \right\|^2 \right).
\end{aligned}
\tag{20}
$$

By using the Chernoff bound for the Q-function, it can be readily obtained that

$$
\begin{aligned}
&Pr \left(\left\| \mathbf{y} - \mathbf{R}\mathbf{x}^{(2)} \right\|^2 \leq \left\| \mathbf{y} - \mathbf{R}\mathbf{x}^{(1)} \right\|^2 \right) \\
&\leq \frac{1}{\left(1 + \left\| \mathbf{x}^{(1)} - \mathbf{x}^{(2)} \right\|^2 / 4\sigma^2 \right)^n} \leq \frac{1}{\left(1 + \frac{d_{\min}^2}{4\sigma^2} \right)^n}.
\end{aligned}
\tag{21}
$$

Thus, we get

$$
\begin{aligned}
Pr\left(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)} | A\right) &\leq \frac{1}{|\mathcal{Q}|^n} \sum_{\mathbf{x}^{(1)} \in \mathcal{Q}^n} \sum_{\substack{\mathbf{x}^{(2)} \in \mathcal{I}, \\ \mathbf{x}^{(2)} \neq \mathbf{x}^{(1)}}} \frac{1}{\left(1 + \frac{d_{\min}^2}{4\sigma^2} \right)^n} \\
&\leq |\mathcal{I}| \frac{1}{\left(1 + \frac{d_{\min}^2}{4\sigma^2} \right)^n} \leq \left(\frac{|\mathcal{Q}|}{1 + \frac{d_{\min}^2}{4\sigma^2}} \right)^n.
\end{aligned}
\tag{22}
$$

Combining (22) and (19), we have

$$
\begin{aligned}
P_f =& Pr(A^c) + Pr\left(\hat{\mathbf{x}} \neq \mathbf{x}^{(1)}|A\right) Pr(A) \\
\leq& (1-p) \sum_{i=1}^{n} \frac{|\mathcal{Q}|}{\left(1 + \frac{d_{\min}^2}{4\sigma^2}\right)^i} + \left(\frac{|\mathcal{Q}|}{1 + \frac{d_{\min}^2}{4\sigma^2}}\right)^n
\end{aligned}
\tag{23}
$$

In high SNR, the symbol error rate P_s can be approximated by P_f, $P_s \approx P_f$, where a frame error is caused by a single symbol error with high probability.

From (22), if p is fixed for all SNRs and $p \neq 1$, the first term dominates P_f. As (23) is only an upper bound, we can only conclude that the diversity order of the deterministic pruning is at least one. But from simulation results, we find the diversity order of the deterministic pruning is indeed one[1]. Eq. (22) also indicates if the random pruning wants to achieve a diversity order n, $1-p$ must at least decrease as fast as $\frac{1}{\left(1+d_{\min}^2/4\sigma^2\right)^{n-1}}$. Therefore, p must vary according to the SNR or σ^2. We thus choose

$$
p = 1 - \xi \left(\frac{1}{1 + d_{\min}^2/4\sigma^2}\right)^{K-1}.
\tag{24}
$$

where ξ is a constant. Substituting (24) into (22), we have

$$
P_f \leq \xi \sum_{i=1}^{n} \frac{|\mathcal{Q}|}{\left(1 + \frac{d_{\min}^2}{4\sigma^2}\right)^{K+i-1}} + \left(\frac{|\mathcal{Q}|}{1 + \frac{d_{\min}^2}{4\sigma^2}}\right)^n
\tag{25}
$$

If $K < n$, the first term dominates P_f in high SNR and the other terms can be neglected. Therefore, the random pruning achieves at least a diversity order K. From (25), we can see that ξ controls the SNR gain achieved by the statistical pruning.

If we want to achieve the full diversity order n, we can choose

$$
p = 1 - \beta P_{ML} \left(\frac{|\mathcal{Q}|}{1 + d_{\min}^2/4\sigma^2}\right)^{-1}.
\tag{26}
$$

Substituting (26) into (22), we have

$$
P_f \leq \beta P_{ML} \sum_{i=1}^{n} \frac{1}{\left(1 + \frac{d_{\min}^2}{4\sigma^2}\right)^{i-1}} + \left(\frac{|\mathcal{Q}|}{1 + \frac{d_{\min}^2}{4\sigma^2}}\right)^n,
\tag{27}
$$

where β controls the SNR loss incurred by the statistical pruning compared with the MLD. When SNR becomes high, the asymptotic form of P_{ML} can be found in [10].

From the derivation of (23), we do not need to keep p as a constant for all p_k, $k = 1, \ldots, n$. For example, if we want to achieve a diversity order K, we can choose

$$
p_k = 1 - \xi \left(\frac{1}{1 + d_{\min}^2/4\sigma^2}\right)^{K-k}.
\tag{28}
$$

When $k > K$, we have $p_k < 0$ in high SNR, which means all the children are pruned except the one with minimum cost. We thus can simply set $p_k = 0$ for $k > K$. Similar to (24) the

---

[1]This suggests that more fine analysis is desired to obtain the lower bound on the performance of deterministic pruning.

performance of the random pruning with (28) can be derived as

$$
P_f \leq \xi \frac{K(K+1)}{2} \sum_{i=1}^{n} \frac{|\mathcal{Q}|}{\left(1 + \frac{d_{\min}^2}{4\sigma^2}\right)^{K+i-1}} + \left(\frac{|\mathcal{Q}|}{1 + \frac{d_{\min}^2}{4\sigma^2}}\right)^n,
\tag{29}
$$

which achieves at least a diversity order K. Compare (25) and (29), different SNR gains are obtained by (28) and (24) with different complexity.

For the cost pruning rule (pruning rule 2), we assume that the initial radius is infinite. With the similar analysis as the pruning rule 1, in order to achieve a diversity order at least K, we set δ as the solution of

$$
1 - \gamma\left(1, \frac{\delta}{2\sigma_n^2}\right) = \frac{\xi}{(1 + d_{\min}^2/4\sigma_n^2)^K},
\tag{30}
$$

where $\gamma(\alpha, x)$ is an incomplete gamma function. We simply set $\delta_k = \delta$ for $n > k > n - K + 1$, and $\delta_k = 0$ for $k \leq n - K + 1$. Therefore, the cost threshold δ only depends on SNR and K but not on k, which reduces the parameters to control the performance and complexity tradeoff.

Remarks:

- The statistical pruning can also be applied to other types of tree search algorithms for MIMO detection such as best first search, breadth first search and iterative deepening. For example, for best first search, we can use statistical pruning to reduce both computational complexity and memory size.

- The IRA in [5] is only a special case of the general statistical pruning algorithm given by Algorithm 1. The IRA needs to be run iteratively until a feasible point is found, which brings additional complexity. In addition, IRA cannot attain different diversity orders and achieve a flexible performance-complexity tradeoff.

- The upper bound in (22) may not be tight. The p given in (24) and (26) may achieve better performance than that suggested by (22). This also holds for δ in (30).

- From our simulation results in Section VI, we find that the performance difference between random pruning and dependent pruning is small by using the same p defined in (24) and (26). It seems that with the same p random pruning only has an SNR gain over dependent pruning even though the dependent pruning rule is stronger than the random pruning rule. We can also apply (24) and (26) for dependent pruning to achieve the same diversity order. However, the analysis for dependent pruning seems to be intractable.

- We always keep the radius infinity throughout the analysis. If the radius is reduced during the detection, the probability that the true solution is pruned may be increased. Therefore, the performance analysis in this section is a lower bound for the true performance.

V. SIMULATION RESULTS

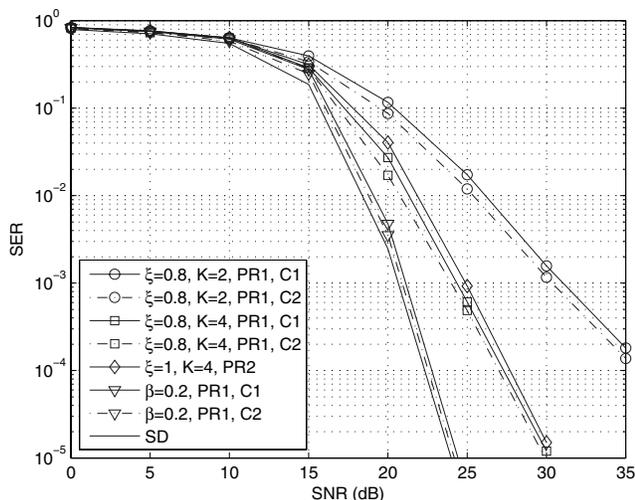We now simulate our SD with statistical pruning for an uncoded MIMO system with 8 transmit and 8 receive antennas

Fig. 1. Performance comparison of SD with different statistical pruning rules for an 8×8 MIMO system with 16QAM.
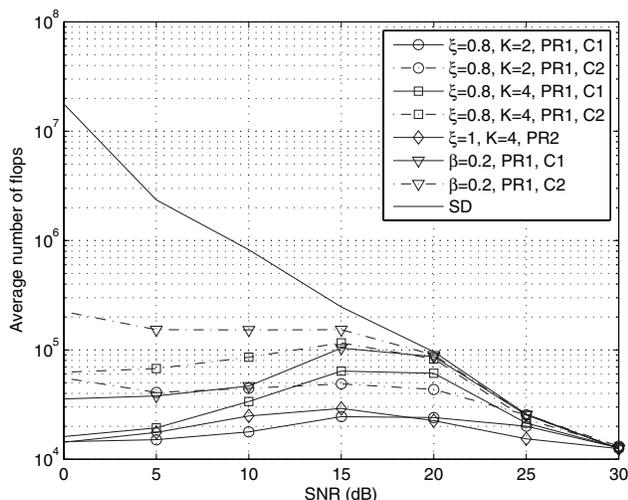

Fig. 2. Complexity comparison of SD with different statistical pruning rules for an 8×8 MIMO system with 16QAM.

over a flat Rayleigh fading channel. 16QAM is used. The MATLAB V5.3 command "flops" is used to count the number of flops. Only the flops of the search algorithm are counted by ignoring the preprocessing stage. The SD with statistical pruning is compared with SD for both performance and complexity. PR1 and PR2 denote pruning rules 1 and 2. In PR1, C1 and C2 denote cases I and II.

Fig. 1 shows the symbol error rate (SER) of the SD with different statistical pruning rules. With different K in (24) and (30), our SD with statistical pruning achieves different diversity orders. If p is chosen according to (26), our new detector performs close to the MLD and achieves a diversity order 8 for pruning rule I. For the pruning rule I, with the same ξ and K, case I performs worse than case II, which agrees with our analysis that case I is a stronger pruning rule than case II. However, the performance loss due to the use of case I is small as compared with case II.

Fig. 2 compares the average number of flops for the SD with different statistical pruning rules. In low SNR, our new detector has much less complexity than the SD in all cases, while the performance difference is negligible in this region. The complexity saving can be 3 order of magnitude. In high SNR, the complexity our new detector converges to that of the SD for all the ξ and K settings. When p is chosen according to (24), the SD with statistical pruning only has less than 0.5 dB loss over the MLD, while the complexity saving in low SNR is significant. Increasing K, the complexity increases in all cases. For the pruning rule I, the complexity of case I is less than that of case II with the same ξ and K. The complexity of the SD with statistical pruning can be made to be a constant if ξ is chosen according to SNR. From the simulation results, we see that the statistical pruning strategy indeed achieves a flexible tradeoff between complexity and performance.

VI. CONCLUSION

In this paper, we have given a unified framework for MIMO detection with statistical pruning. A general pruning rule is given in the context of the sphere decoder. Our pruning rules are especially useful when the number of transmit antennas is large or the SNR is low. For the random pruning rule, fixing the pruning probability, we show that the diversity order is only one by deriving the union bound on the symbol error probability. We have suggested choosing the pruning probability according to the noise variance. By choosing a proper pruning distribution function, our statistical pruning can achieve arbitrary diversity orders and SNR gains. Interesting further work includes deriving the asymptotic symbol error probability for different pruning rules, and obtaining a precise expression for p with certain diversity order and SNR gain requirement.

REFERENCES

[1] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture," *Electronics Letters*, vol. 35, no. 1, pp. 14–15, Jan. 1999.

[2] E. Viterbo and J. Bouros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.

[3] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Processing*, vol. 53, no. 4, pp. 1474 – 1484, April 2005.

[4] W. Zhang, *State Space Search: Algorithms, Complexity and Applications*. New York: Springer-Verlag, 1999.

[5] R. Gowaikar and B. Hassibi, "Efficient statistical pruning for maximum likelihood decoding," in *Proc. of ICASSP*, 2003, pp. 49–52.

[6] T. Cui and C. Tellambura, "An efficient generalized sphere decoder for rank-deficient MIMO systems," *IEEE Commun. Lett.*, vol. 9, no. 5, pp. 423 – 425, May 2005.

[7] ——, "Joint channel estimation and data detection for OFDM systems via sphere decoding," in *Proc. of GLOBECOM*, Dec. 2004.

[8] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181–191, 1994.

[9] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Education, Dec. 2001.

[10] X. Zhu and R. D. Murch, "Performance analysis of maximum likelihood detection in a MIMO antenna system," *IEEE Trans. Commun.*, vol. 50, pp. 187 – 191, Feb. 2002.