# Sphere Decoding with a Probabilistic Tree Pruning

Byonghyo Shim, *Member, IEEE,* and Insung Kang, *Member, IEEE,*

***Abstract*—In this paper, we present a near ML-achieving sphere decoding algorithm that reduces the number of search operations in the sphere-constrained search. Specifically, by adding a probabilistic noise constraint on top of the sphere constraint, a more stringent necessary condition is provided, particularly at an early stage, and hence branches unlikely to be survived are removed in the early stage of sphere search. The trade-off between the performance and complexity is easily controlled by a single parameter, so-called pruning probability. Through the analysis and simulations, we show that the complexity reduction is significant while maintaining the negligible performance degradation.**

***Index Terms*—Sphere decoding, maximum likelihood decoding, probabilistic noise constraint, sphere constraint, lattice, probabilistic tree pruning, multiple-input-multiple-output (MIMO) system**

## I. Introduction

The relationship between the transmitted symbol vector and the received signal vector of the communication system is commonly described by

$$\mathbf{y_c} = \mathbf{H_c s_c} + \mathbf{v_c} \tag{1}$$

where $\mathbf{s_c}$ is the transmitted symbol vector whose components are elements of a finite set of complex numbers, $\mathbf{y_c}$ is the complex received signal vector, $\mathbf{v_c}$ is the complex noise vector whose components are i.i.d. with a circularly symmetric complex Gaussian distribution, and $\mathbf{H_c}$ is a complex channel matrix. Under the assumption that the channel matrix is known, the maximum likelihood (ML) decoding problem is formulated as

$$\max_{\mathbf{s_c}} P_r(\mathbf{y_c} \,|\, \mathbf{H_c}, \mathbf{s_c}). \tag{2}$$

Equivalently,

$$\min_{\mathbf{s_c}} ||\, \mathbf{y_c} - \mathbf{H_c s_c} \,||^2. \tag{3}$$

Although the ML decoding is optimal for achieving the minimum error probability, it has not been paid much attention due to the exponential complexity in the dimension of the transmitted symbol vector, which results in an NP-hard problem [1], [5]. Recently, this problem has been re-visited due to the lower complexity decoding method proposed by Fincke and Pohst [1]–[7], popularly referred to as *Sphere Decoding* (SD) algorithm. The SD algorithm has received considerable attention as an effective detection scheme for MIMO systems in wireless channels [8], [14]–[16].

The principle of the SD algorithm is to search the nearest lattice point to the received signal vector within a sphere radius. A condition that the transmitted symbol vector should be within a sphere centered at the received signal vector is called the sphere constraint. By using QR decomposition, the skewed lattice $\mathbf{H_c s_c}$ is linearly transformed into another lattice $\mathbf{R s_c}$. Since $\mathbf{R}$ has an upper triangular structure, the transformed lattice is more suitable for searching. The nearest lattice point in the transformed lattice to the received signal vector, i.e., ML solution, is found by enumerating all the lattice points inside the sphere. Although the SD algorithm offers significant reduction in computational complexity, it still requires considerable amount of computations compared with MMSE or DFE based algorithms [5], [9], [19].

There have been some studies on the modification, mainly on the change of radius, of the SD algorithm to reduce the computational complexity. In [7], [9], a simple method to increase radius search (IRS) was proposed. In this approach, the SD algorithm starts with an initial radius $r_1$. When the search fails, the radius is increased to $r_2$ ($r_2 > r_1$) and this search is repeated until the ML solution is found. In [11], Wanlun and Giannakis improved this method by reusing the path information of the incomplete tree in case of the search failure. By identifying the most promising paths according to the average distance metric whenever the search fails, they avoided the redundant computation needed for starting the search again. Gowaikar and Hassibi proposed an algorithm called increasing radii algorithm (IRA) [12], [13]. While maintaining the basic structure of IRS, they further provided the schedule of radii $r_{\epsilon,i} = i + \delta_\epsilon \log m$ for each layer of the search where $\epsilon$ is the probability that the transmitted symbol is not in the set being searched and $\delta_\epsilon$ is the scaling factor depending on $\epsilon$. In the first run, $r_{\epsilon_1,i}$ $(i = 1, \cdots, m)$ are being used. When search fails, then $r_{\epsilon_2,i}$ are used with $\epsilon_2$ smaller than $\epsilon_1$ to generate bigger radius.

Our approach lies on an extension of the approaches in [12], [13] in the sense that we pursue further reduction of the computational complexity by exploiting an aggressive tree pruning strategy, which is realized by controlling the sphere radius per layer. Motivated by the fact that the sphere constraint of the SD algorithm offers a loose necessary condition, especially in the early layers of search, we model the contributions of unvisited layers by random variables and impose pruning conditions based on the statistics of these random variables. While the methods in [12], [13] try to obtain the $r_{\epsilon,i}$ from $\epsilon$, the pruning conditions in this paper are obtained by the pruning probability which controls the level of pruning. These additional conditions to the sphere constraint are collectively called probabilistic noise constraint. The probabilistic noise constraint tightens the necessary condition in each layer, particularly in the early layers. As a result, branches that are unlikely to survive are pruned before they are traced further into the child nodes. In this general framework, the

B. Shim was with Qualcomm Inc., CA. He is now with Dept. of Computer Science, Korea Univ., Seoul, Korea (email: bshim@korea.ac.kr)

I. Kang is with Qualcomm Inc., CA (email: insungk@qualcomm.com)

This research was supported in part by a Korea University Grant.

proposed SD algorithm, henceforth referred to as probabilistic tree pruning sphere decoding (PTP-SD) algorithm, can be explained as a hybrid of the sphere constraint and the noise constraint tree search algorithms. In the early layers, the noise constraint affects more, and in the later layers, the sphere constraint dominates the search. Further, the SD algorithm is interpreted as a special case of the PTP-SD algorithm where the pruning probabilities are set to zero.

This paper is organized as follows. In Section II, the Fincke and Pohst algorithm and its modified strategy called Schnorr-Euchner enumeration are briefly reviewed. In Section III, the proposed PTP-SD algorithm is presented. In Section IV, a lower bound analysis of the expected complexity for both SD and PTP-SD algorithms is provided. Section V presents simulation results for MIMO systems and comparisons with [13]. It is shown that the PTP-SD algorithm provides considerable complexity reduction with negligible performance loss. Section VI concludes this paper.

## II. Sphere Decoding

### A. Fincke and Phost algorithm

Lattice codes in Fincke and Pohst algorithm are constructed in an $n$-dimensional Euclidean space $\mathcal{R}^n$. Hence, the complex number signal model in (1) needs to be reformulated to a real number signal model. We define the real signal model components as follows:

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} \Re(\mathbf{y_c}) \\ \Im(\mathbf{y_c}) \end{bmatrix}, \ \mathbf{s} = \begin{bmatrix} \Re(\mathbf{s_c}) \\ \Im(\mathbf{s_c}) \end{bmatrix}, \ \mathbf{v} = \begin{bmatrix} \Re(\mathbf{v_c}) \\ \Im(\mathbf{v_c}) \end{bmatrix}, \\ \mathbf{H} &= \begin{bmatrix} \Re(\mathbf{H_c}) & -\Im(\mathbf{H_c}) \\ \Im(\mathbf{H_c}) & \Re(\mathbf{H_c}) \end{bmatrix} \end{aligned} \tag{4}$$

where $\Re(\,\cdot\,)$ and $\Im(\,\cdot\,)$ are the real and imaginary parts of its argument. Then the real number signal model is given by

$$\mathbf{y} = \mathbf{Hs} + \mathbf{v} \tag{5}$$

where $\mathbf{y} \in \mathcal{R}^n$, $\mathbf{s} \in \Lambda \subset \mathcal{Z}^m$, $\mathbf{v} \in \mathcal{R}^n$, and $\mathbf{H} \in \mathcal{R}^{n \times m}$.

Fincke and Pohst algorithm is summarized as follows: Let $c_0$ be the radius square of an $n$-dimensional sphere $S(\mathbf{y}, \sqrt{c_0})$ centered at $\mathbf{y}$. The ML solution is the nearest lattice point to $\mathbf{y}$. In order to find the ML point, instead of searching all lattice points in $\Lambda$, Fincke and Pohst algorithm searches the lattice points $\mathbf{Hs}$ inside of the sphere, i.e., $S(\mathbf{y}, \sqrt{c_0}) \cap \Lambda$. This necessary condition on the ML solution can be expressed as

$$c_0 \ \geq \ ||\,\mathbf{y} - \mathbf{Hs}\,||^2. \tag{6}$$

Note that (6) has no special structure to make the search convenient. In order to make a structure easy for searching, we first perform QR-decomposition of $\mathbf{H}$ as

$$\mathbf{H} = [\mathbf{Q} \ \ \mathbf{U}] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{7}$$

where $\mathbf{R}$ is an $m \times m$ upper triangular matrix with positive diagonal elements, $\mathbf{0}$ is an $(n-m) \times m$ zero matrix, and $\mathbf{Q}$ and $\mathbf{U}$ are $n \times m$ and $n \times (n-m)$ unitary matrices. Substituting (7) in (6), we have

$$d_0 \ \geq \ ||\,\mathbf{y}' - \mathbf{Rs}\,||^2 \tag{8}$$

where $\mathbf{y}' = \mathbf{Q}^T\mathbf{y}$ and $d_0 = c_0 - ||\mathbf{U}^T\mathbf{y}||^2$. We denote $\sqrt{d_0}$ as modified sphere radius. Due to the upper triangular $\mathbf{R}$ matrix, $\mathbf{y}' - \mathbf{Rs}$ has a structure enabling the progressive search as

$$\begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ \\ y_m' \end{bmatrix} - \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,m} \\ 0 & r_{2,2} & \cdots & r_{2,m} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{m-1,m} \\ 0 & 0 & \cdots & r_{m,m} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ \\ s_m \end{bmatrix}.$$

Hence, (8) can be expressed as

$$\begin{aligned} d_0 \ &\geq \sum_{j=1}^{m} (\,y_j' - \sum_{k=j}^{m} r_{j,k} s_k\,)^2 \qquad (9) \\ &= \ (y_m' - r_{m,m}s_m)^2 + (y_{m-1}' - \sum_{k=m-1}^{m} r_{m-1,k}s_k)^2 + \cdots. \end{aligned}$$

In the Viterbi algorithm parlance, terms in the righthand side correspond to the branch metrics. Since it is natural to start the search from the bottom layer and trace upward, with a reference of the bottom layer as the first layer, the recursive relationship becomes

$$P_{m+1} \ = \ 0 \tag{10}$$

$$P_k \ = \ P_{k+1} + B_k, \ k = m, m-1, \cdots, 1 \tag{11}$$

$$B_k \ = \ (\,y_k' - \sum_{j=k}^{m} r_{k,j}\,s_j\,)^2 \tag{12}$$

where $P_k$ is the $(m-k+1)$-th layer path metric and $B_k$ is the $(m-k+1)$-th layer branch metric. Since $P_k$ and $B_k$ are functions of $s_k, \cdots, s_m$, we hereby denote $P_k(s_k^m)$ and $B_k(s_k^m)$ when an explicit dependency is desired.

For the first layer (the bottom layer), the necessary condition for $\mathbf{Rs}$ being inside the new sphere, $S(\mathbf{y}', \sqrt{d_0})$, is

$$\underbrace{(y_m' - r_{m,m}s_m)^2}_{\text{path metric } P_m \text{ = branch metric } B_m} \leq d_0. \tag{13}$$

The corresponding range of $s_m$ is

$$\left\lceil \frac{y_m' - \sqrt{d_0}}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \frac{y_m' + \sqrt{d_0}}{r_{m,m}} \right\rfloor \tag{14}$$

where $\lceil\,\cdot\,\rceil$ and $\lfloor\,\cdot\,\rfloor$ are the ceiling and floor of its argument. In general, the necessary condition for the $(m-k+1)$-th layer is

$$P_k = P_{k+1} + B_k(s_k^m) \leq d_0. \tag{15}$$

Substituting (12) into (15), we have

$$\left( y_k' - \sum_{j=k+1}^{m} r_{k,j}\,s_j \, - r_{k,k}\,s_k \right)^2 \leq d_0 - P_{k+1}. \tag{16}$$

The corresponding range of $s_k$ is expressed as

$$s_{k,\min} \leq s_k \leq s_{k,\max} \tag{17}$$

where

$$s_{k,\min} = \left\lceil \frac{1}{r_{k,k}}(y'_k - \sum_{j=k+1}^{m} r_{k,j}\, s_j - \sqrt{d_0 - P_{k+1}}) \right\rceil \quad (18)$$

$$s_{k,\max} = \left\lfloor \frac{1}{r_{k,k}}(y'_k - \sum_{j=k+1}^{m} r_{k,j}\, s_j + \sqrt{d_0 - P_{k+1}}) \right\rfloor . \quad (19)$$

The lattice points satisfying (17) are searched to find ML solution. If no lattice point satisfying (17) for all $k$ is found, the search can restart with a larger sphere radius. Clearly, how to choose an initial sphere radius will affect the complexity and performance. Refer [7], [9] for more detail.

### B. Schnorr-Euchner Enumeration

In [1], [2], searching order in each layer is based on the lexicographic ordering called Pohst enumeration, which checks the candidate in the order of $s_{k,\min}$, $s_{k,\min}+1, \cdots, s_{k,\max}$. The complexity can be further reduced by Schnorr-Euchner enumeration [4]–[6], where the candidates are sorted and examined based on their path metric values. Recall that the path metric is expressed as $P_k = P_{k+1} + B_k$ and $P_{k+1}$ is common for all child nodes having the same parent, so the ordering is effectively based on the branch metric $B_k(s_k^m)$. The unknown in the branch metric $B_k(s_k^m)$ is $s_k \in \{s_{k,\min}, \cdots, s_{k,\max}\}$ with the cardinality to be $N_s = s_{k,\max} - s_{k,\min} + 1$. The candidates ordered by Schnorr-Euchner strategy are denoted as

$$s_{k,1}^o,\ s_{k,2}^o,\ \cdots s_{k,N_s}^o \quad (20)$$

where $i > j$ implies that $B_k(s_{k,i}^o) \geq B_k(s_{k,j}^o)$. The starting point is given by

$$s_{k,1}^o = \left\lceil \frac{1}{r_{k,k}} \left( y'_k - \sum_{j=k+1}^{m} r_{k,j}\, s_j \right) \right\rfloor \quad (21)$$

where $\lceil \cdot \rfloor$ is a rounding operator. In a nutshell, the Schnorr-Euchner (SE) enumeration has three distinct advantages over the Pohst enumeration. First, since the search starts from the candidate minimizing the branch metric, it is likely to find the right path earlier than the Pohst enumeration. Second, if the condition in (15) fails for $s_{k,i}^o$, then the condition will also fail for $s_{k,j}^o$, $j > i$. That is, we can skip the search for the rest of candidates in the layer, thereby saving complexity considerably. Finally, the initial lattice point we obtain from the SE enumeration is invariant if we set the sphere radius sufficiently large. Therefore, the SD algorithm is free from the initial radius selection. The initial point found by the SE enumeration is called *Babai* point [17]. Once the Babai point is found, the distance from the received signal vector to the Babai point is used as a new sphere radius.

## III. Sphere Decoding with a Probabilistic Tree Pruning

### A. Sphere Constraint

In the SD algorithm, the branches with path metrics larger than the given radius square are never searched, which reduces the complexity of the SD algorithm over an exhaustive search significantly. However, the necessary condition given by (9) is too loose for bottom layers. For an example, see (13) where only a single term needs to be less than or equal to $d_0$. As a result, branches more likely outside the sphere are retained until the search proceeds to the upper layers. Figure 1 illustrates a tree search for BPSK symbols where the search starts with an initial radius of $d_0 = 100$. In this example, all four nodes of the second layer ($k = 3$) are being searched even though the branches having the path metric 8 is highly likely to be outside of the sphere. When the dimension of the transmitted symbol vector and/or the constellation of the transmitted symbol vector component becomes large, the situation would be exacerbated [9], [10], [19].

### B. Probabilistic Noise Constraint

Recall that the SD algorithm enumerates all the candidates $s_1^m$ satisfying

$$B_1 + B_2 + \cdots + B_m \leq d_0. \quad (22)$$

However, only the contributions of visited layers are counted in the $(m-k+1)$-th layer search as

$$B_k + \cdots + B_m \leq d_0 \quad (23)$$

The key idea behind the probabilistic noise constraint is to use (22) instead of (23) for all layers in the search. Note, however, the branch metrics $B_1, \cdots, B_{k-1}$ are unavailable in the $(m-k+1)$-th layer. Assuming perfect decoding, these branch metrics are modeled by the noise statistics as

$$B_i = (\, y'_i - \sum_{j=i}^{m} r_{i,j}\, s_j \,)^2 = v_i^2 \;\text{ for }\; \mathrm{i} = 1, \cdots, \mathrm{k}-1, \quad (24)$$

where $v_i$ is the $i$-th component of the noise vector $\mathbf{v}$. From (11) and (24), the new necessary condition becomes

$$\sum_{i=1}^{m} B_i = P_k + \sum_{i=1}^{k-1} v_i^2 \leq d_0. \quad (25)$$

Since $v_1, \cdots, v_{k-1}$ are values from i.i.d. Gaussian distribution, $\sum_{i=1}^{k-1} v_i^2$ becomes the chi-square random variable with $k-1$ degrees of freedom. Denoting $\psi_{k-1} = \sum_{i=1}^{k-1} v_i^2$, we have

$$P_k(s_k^m) + \psi_{k-1} \leq d_0. \quad (26)$$

In order to formulate a probabilistic pruning method using a tightened necessary condition, we introduce a concept of pruning probability. On each node visited, we examine the probability that the rest of the tree is decoded perfectly so that the remaining portion is a pure noise contribution. If the probability of this event is too small and thus less than a threshold, then we regard this event as a rare one and prune the subtree starting from the the node. This concept is mathematically equivalent to

$$P_r(\psi_{k-1} \leq d_0 - P_k(s_k^m)) \ < \ P_\epsilon \quad (27)$$

where $P_\epsilon$ is the pruning probability. Rewriting (27), $F_{\psi_{k-1}}(d_0 - P_k(s_k^m)) < P_\epsilon$ where $F(\psi; k) = \frac{\gamma(k/2, \psi/2)}{\Gamma(k/2)}$ is

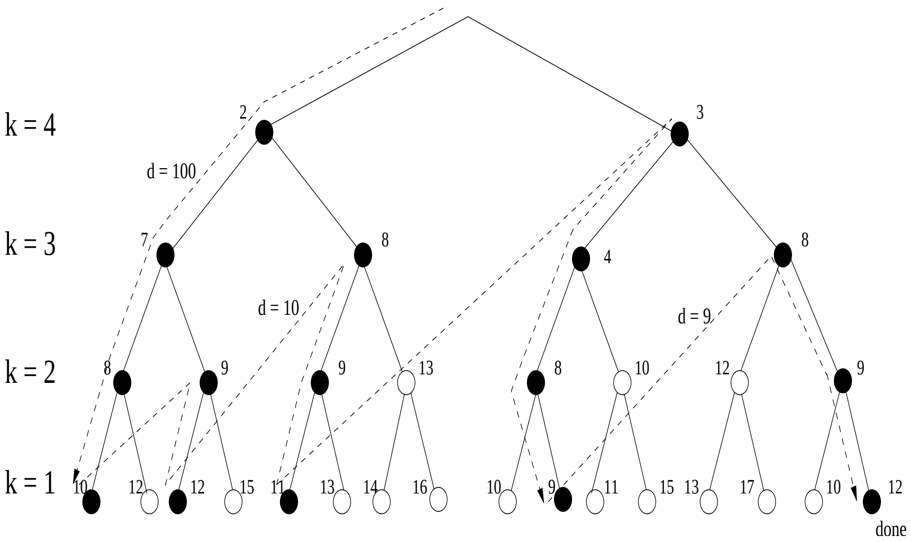


Fig. 1. Illustration of sphere decoding in a binary tree employing depth-first search and SE enumeration. The numbers labelled for each node are the path metrics. Note that the white nodes are skipped since they are outside of sphere constraint.

the cumulative distribution function of $\psi_k$, where $\Gamma(k)$ and $\gamma(k,x)$ are Gamma function and incomplete Gamma function, respectively [18]. Thus, we have

$$d_0 - P_k(s_k^m) \; < \; F^{-1}_{\psi_{k-1}}(P_\epsilon). \tag{28}$$

The $(m-k+1)$-th layer pruning parameter, $\beta_{k-1}$, is defined by

$$\beta_{k-1} \; = \; F^{-1}_{\psi_{k-1}}(P_\epsilon). \tag{29}$$

Rearranging the terms, we have

$$P_k(\, s_k^m \,) > d_0 - \beta_{k-1}. \tag{30}$$

If the path metric of a node is larger than $d_0 - \beta_{k-1}$ then the combination of this path metric and unvisited noise contribution is unlikely to satisfy the sphere constraint. Thus, we prune all branches under the node. The tightened necessary condition is given by

$$P_k(s_k^m) \leq d_0 - \beta_{k-1} = \tilde{d}_0(k). \tag{31}$$

This tightening of the necessary condition by the addition of $\beta_{k-1}$ is called the probabilistic noise constraint. Analogous to (16), the necessary condition for the branch metric $B_k(s_k^m)$ becomes

$$B_k(s_k^m) \leq \tilde{d}_0(k) - P_{k+1}(s_{k+1}^m). \tag{32}$$

It might be worth to make some qualitative remarks on the PTP-SD algorithm.

- Both contributions of visited layers and unvisited layers, $P_k(s_k^m)$ and $\psi_{k-1}$, are considered in the search.
- When the per-layer pruning parameter $(\beta_{k-1})$ increases, the effective sphere radius $(\tilde{d}_0(k))$ decreases as seen in (31). Since $\beta_{k-1}$ is an increasing function of $P_\epsilon$, this indicates that larger $\beta_{k-1}$ involves more aggressive tree pruning.
- The SD algorithm can be explained as a special case of the proposed PTP-SD algorithm where $P_\epsilon = 0$ and hence $\beta_{k-1} = 0 \;\; \forall \; k$.

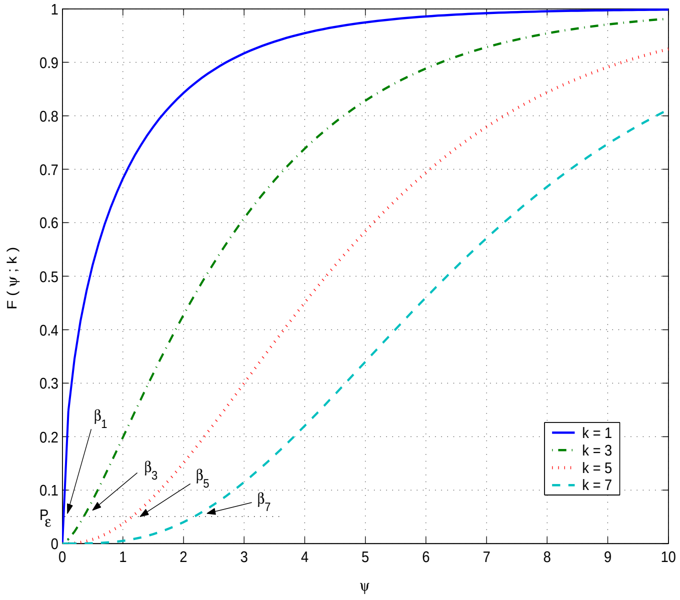


Fig. 2. $\beta_k$ and the CDF of $\psi_k$

- For the given pruning probability, $\beta_{k-1}$ would be larger leading to tighter sphere constraints in the early layers because $\psi_{k-1}$ has larger degrees of freedom in the early layers. This effect will diminish as $\beta_{k-1}$ would get smaller in the later layers (see Fig. 2).
- The number of additional operations required for the PTP-SD algorithm is minimal. At most $m$ values of $\beta_{k-1}$ are needed and they can be tabulated prior to the operation. Also, only single subtraction operation is required per layer to obtain $\tilde{d}_0(k)$.

The PTP-SD algorithm is described in Table I and the illustration of a binary tree search is shown in Fig. 3. Comparing with Fig. 1, we observe that more branches in the early layers are removed from the search. For example, branches with large path metrics (such as the branch with the path metric 8 in the second layer) is removed from the tree before it is further expanded into the third layer.

## IV. Complexity Analysis

In this section, we provide a complexity analysis of the SD and PTP-SD algorithms. Our analysis is different from [9] and similar to [19] in a sense that we obtain lower bounds of the algorithms. The reason is that 1) the expected complexity includes complicated integrals so that it is not easy to evaluate the behavior quickly and 2) we are mainly interested in the complexity comparison between the SD and PTP-SD algorithms. Thus, instead of obtaining an asymptotic rate for complexity as described in [19], we obtain a simple expression having an explicit dependency of the sphere radius to evaluate complexity difference between SD and PTP-SD. In deriving the complexity, we do not count QR decomposition complexity and only consider the complexity of the sphere search. Nevertheless, since QR decomposition complexity is well-known[1], one can easily include this complexity into the analysis. In addition, since the operations per node (branch metric computations and comparisons) are equal for both algorithms except one subtraction per layer to compute $\tilde{d}_0(k)$, we consider the number of nodes visited $N$ as a metric for the complexity, which is given by

$$\begin{aligned} N &= N_1 + N_2 + \cdots + N_m && (33) \\ &= \sum_{l=1}^{m} \sum_{s_l^m} \mathbf{1}_{\{P_l(\,s_l^m\,) < d_0\}} && (34) \end{aligned}$$

where $N_l$ is the number of nodes visited at the $(m-l+1)$-th layer and $\mathbf{1}_{\{X < \alpha\}}$ is the indicator function equal to 1 if $X < \alpha$ and 0 otherwise.

### A. SD Complexity

Before developing the expression in (34), we consider a system model with a real and square matrix $\mathbf{H}$ given by

$$\mathbf{y} = \mathbf{H}\tilde{\mathbf{s}} + \mathbf{v} \qquad (35)$$

where $\mathbf{H} \in \mathcal{R}^{m \times m}$ has i.i.d. entries $\sim N(0, \sigma_h^2)$, $\mathbf{v} \in \mathcal{R}^m$ has i.i.d. entries $\sim N(0, \sigma_v^2)$, and $\tilde{\mathbf{s}}$ is the transmitted symbol vector whose entries are i.i.d. and from the equiprobable set $\mathcal{S}= \{-\frac{L-1}{\tau}, -\frac{L-3}{\tau}, \cdots \frac{L-3}{\tau}, \frac{L-1}{\tau}\}$. Note that $\tau$ is a constant to make $E(|\tilde{s}_1 + j\tilde{s}_2|^2)$ to unity in complex modulation where $\tilde{s}_1, \tilde{s}_2 \in \mathcal{S}$. For example, $\tau = \frac{1}{\sqrt{2}}$ for QPSK and $\tau = \frac{1}{\sqrt{10}}$ for 16-QAM.

From this system model, the received SNR, denoted as $\xi$, becomes

$$\xi = \frac{E(\,||\mathbf{H}\tilde{\mathbf{s}}||^2\,)}{E(\,||\mathbf{v}||^2\,)} = \frac{E(\tilde{\mathbf{s}}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\mathbf{H}\tilde{\mathbf{s}})}{E(\mathbf{v}^{\mathbf{t}}\mathbf{v})} = \frac{m^2\sigma_h^2}{2m\sigma_v^2}. \qquad (36)$$

Hence,

$$\sigma_h^2 = \frac{2\xi\sigma_v^2}{m}. \qquad (37)$$

In addition, from (35), we have

$$\begin{aligned} ||\,\mathbf{y} - \mathbf{Hs}\,|| &= ||\,\mathbf{H}(\tilde{\mathbf{s}} - \mathbf{s}) + \mathbf{v}\,|| \\ &= ||\,\mathbf{R}(\tilde{\mathbf{s}} - \mathbf{s}) + \mathbf{Q}^{\mathbf{T}}\mathbf{v}\,|| = ||\,\mathbf{R}(\tilde{\mathbf{s}} - \mathbf{s}) + \tilde{\mathbf{v}}\,|| \end{aligned}$$

where $\tilde{\mathbf{v}} = \mathbf{Q}^{\mathbf{T}}\mathbf{v}$. Due to the use of the unitary matrix $\mathbf{Q}$, the statistics of $\tilde{\mathbf{v}}$ remain unchanged. Defining the partial norm of a vector $\mathbf{x}$ as $||\,\mathbf{x}\,||_l^m = (\sum_{i=l}^m |x_i|^2)^{1/2}$, the path metric in (11) becomes $P_l = (||\,\mathbf{R}(\tilde{\mathbf{s}} - \mathbf{s}) + \mathbf{Q}^{\mathbf{T}}\mathbf{v}\,||_l^m)^2$ and hence is a function of random vectors $\mathbf{s}_l^m$, $\tilde{\mathbf{s}}_l^m$, and $\tilde{\mathbf{v}}_l^m$.

*Lemma 1:* The expected complexity at the $(m-l+1)$-th layer $E[N_l]$ is lower bounded by

$$E[N_l] \ge L^{m-l+1} \max\left(0,\, 1 - \frac{E[P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m)]}{d_0}\right) \qquad (38)$$

*Proof:* In this proof, we slightly abuse the notations and use bold face to emphasize random variables. From (34), we have

$$\begin{aligned} E[N_l] &= E[\sum_{s_l^m} \mathbf{1}_{\{P_l(\tilde{\mathbf{s}}_l^m, \tilde{\mathbf{v}}_l^m, s_l^m) < d_0\}}\,] \\ &= \sum_{s_l^m} E[\,\mathbf{1}_{\{P_l(\tilde{\mathbf{s}}_l^m, \tilde{\mathbf{v}}_l^m, s_l^m) < d_0\}}\,] \\ &\overset{(a)}{=} L^{m-l+1} \sum_{s_l^m} \frac{1}{L^{m-l+1}} P_r(P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m, s_l^m) \,<\, d_0) \\ &\overset{(b)}{=} L^{m-l+1} P_r(\,P_l(\,\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m) \,<\, d_0\,) && (39) \end{aligned}$$

where $(a)$ is from

$$E[\mathbf{1}_{\{\,f(x,y) \,<\, \alpha\}}] = P_r(\,f(x,y) \,<\, \alpha\,) \qquad (40)$$

and $(b)$ is because $P_r(s_l^m) = \frac{1}{L^{m-l+1}}$ and

$$P_r(\,g(\mathbf{x}, \mathbf{y}) < \alpha) = \sum_y P_r(g(\mathbf{x}, y) < \alpha)\, P_r(y)$$

Using the Markov inequality $P_r(\mathbf{x} \ge \alpha) \le \frac{E[\mathbf{x}]}{\alpha}$, we have

$$P_r(\mathbf{x} < \alpha) \ge \max\left(0,\, 1 - \frac{E[\mathbf{x}]}{\alpha}\right). \qquad (41)$$

Notice that the max operation is being used to ensure a non-negative bound for the probability. Using (39) and (41), we have

$$E[N_l] \ge L^{m-l+1} \max\left(0, 1 - \frac{E[P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m)]}{d_0}\right)$$

■

Our next step is to get the bound for $E[P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m)]$.

*Lemma 2:* $E[\,(\tilde{s} - s)^2\,] = \frac{2}{3\tau^2}(L^2 - 1)$

*Proof:* One can easily show this by using the uniformity of $\tilde{s}$ and $s$. ■

*Lemma 3:* $E[P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m)]$ is bounded by

$$E[P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m)] \le (m-l+1)\left(\frac{2m}{3\tau^2}(L^2-1)\sigma_h^2 + \sigma_v^2\right)$$

[1] In QR-decomposition, a real matrix $\mathbf{H}$ is factored as orthogonal matrix $\mathbf{Q}$ and upper triangular matrix $\mathbf{R}$. In order to perform QR-decomposition, modified Gram-schmidt orthogonalization, Householder reflection, or sequence of Givens rotations is popularly used. Refer [20]-[23] for detailed description of the algorithm.

TABLE I
**PTP-SD Algorithm**

| | |
|---|---|
| Input: | $d_0$, $\beta_(k)$ $\forall k$ , $\mathbf{y}'$, and $\mathbf{R}$ |
| Output: | $\hat{\mathbf{s}}$ |
| Variable: | $k$ denotes the $(m-k+1)$-th layer being examined<br>$i_k$ denotes the lattice point index sorted by the SE enumeration in the $(m-k+1)$-th layer. |
| **1** : | Set $k=m$, $\tilde{d}_0(m)=d_0-\beta_{m-1}$, $P_{m+1}=0$. |
| **2** : | Compute $s_{k,max}$ and $s_{k,min}$ by (18) and (19) with $d_0$ replaced by $\tilde{d}_0(k)$.<br>$N_s=s_{k,\max}-s_{k,\min}+1$.<br>Compute the branch metrics $B_k(s_k^m)=B_k(s_k,s_{k+1}^m)$ $\forall$ $s_k \in [s_{k,min}, s_{k,max}]$ by (12).<br>Obtain the sorted $s_{k,i_k}$ by the SE enumeration.<br>Set $i_k=0$. |
| **3** : | $i_k=i_k+1$.<br>If $i_k>N_s$, go to 4.<br>Else, go to 5. |
| **4** : | $k=k+1$.<br>If $k=m+1$, output the latest s and terminate.<br>Else, go to 3. |
| **5** : | Update the path metric $P_k(s_{k,i_k}, s_{k+1}^m)=P_{k+1}+B_k(s_{k,i_k}, s_{k+1}^m)$.<br>If $k=1$, go to 6.<br>Else, $k=k-1$, $\tilde{d}_k=d_0-\beta_{k-1}$ go to 2. |
| **6** : | If $P_1<d_0$, save $s$ and update $d_0=P_1$.<br>Go to 3. |

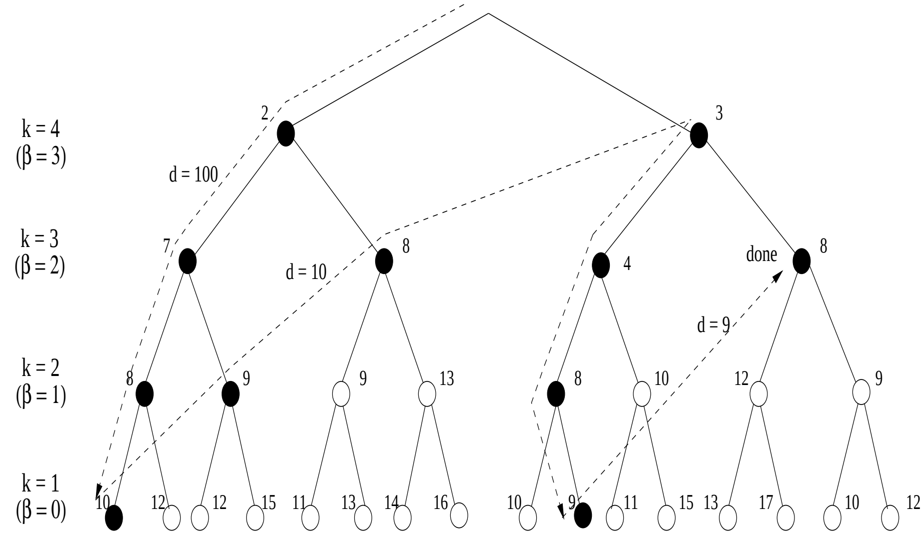


Fig. 3. Illustration of PTP-SD in a binary tree.

*Proof:* See Appendix A. ■

Using the lemmas we have developed so far, we obtain the lower bound of the expected complexity of the SD algorithm in the following theorem.

*Theorem 4:* The expected complexity $E[N]$ is lower bounded by

$$E[N] \geq \sum_{l=1}^{m} L^{m-l+1} \cdot \max\left(0,\, 1-\frac{(m-l+1)(\frac{4\xi}{3\tau^2}(L^2-1)+1)\sigma_v^2}{d_0}\right) \quad (42)$$

*Proof:* Using Lemma 1 and 3, and also noting that $\max(0, 1-y) \leq \max(0, 1-x)$ for $x \leq y$, we have

$$E[N_l] \geq L^{m-l+1} \max\left(0, 1-\frac{E[P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m)]}{d_0}\right)$$
$$\geq L^{m-l+1} \max\left(0,\ 1-\frac{(m-l+1)\left(\frac{2m}{3\tau^2}(L^2-1)\sigma_h^2+\sigma_v^2\right)}{d_0}\right).$$

Using (37), we further have

$$E[N_l] \geq L^{m-l+1} \max\left(0, 1-\frac{(m-l+1)(\frac{4\xi}{3\tau^2}(L^2-1)+1)\sigma_v^2}{d_0}\right)$$

Finally, since $E[N]=\sum_{l=1}^{m} E(\,N_l\,)$, we get (42). ■

From Theorem 4, we observe that the key factors affecting the complexity are the received SNR $\xi$, the number of modulation levels $L$, the matrix size $m$, and the sphere radius $d_0$. The fact that the complexity of the sphere decoding is a decreasing function of the received SNR is matching with our intuition. In high SNR, the sphere decoding algorithm might track the ML path directly so that the solution can be found with minimal complexity[2]. In low SNR, however, the sphere decoding algorithm will search most of the nodes in the tree so that the resulting complexity becomes exponential.

It is interesting to mention that the complexity is a function of the radius $d_0$. From the receiver design viewpoint, the number of levels, the matrix size, and the SNR are parameters dependent on the transmitter and the channel characteristics. Thus, a judicious control of the radius might be a useful way to improve the complexity. Suppose the radius $d_0$ is very large

$$d_0 \gg m(\frac{4\xi}{3\tau^2}(L^2-1)+1)\sigma_v^2 \tag{43}$$

then the complexity would be

$$E[N] \geq \sum_{l=1}^{m} L^{m-l+1}(1-\epsilon) = (1-\epsilon)\frac{L^{m+1}-L}{L-1} \tag{44}$$

for $\epsilon = \frac{m(\frac{4\xi}{3\tau^2}(L^2-1)+1)\sigma_v^2}{d_0} \ll 1$. Evidently, the complexity would be exponential. On the contrary, we expect the small complexity when $d_0$ is small[3]. But this case would have no meaning since no valid lattice point might exist inside the sphere.

### B. PTP-SD Complexity

Our goal in this subsection is to obtain an analytical bound of the PTP-SD algorithm analogous to Theorem 4. First, we take a look at the expression for the number of nodes at the $(m-l+1)$-th layer. Due to the different radius used per layer, we need to enumerate the necessary conditions from the first layer to the $(m-l+1)$-th layer in evaluating $N_l$. $N_l$ is expressed as

$$N_l = \sum_{s_l^m} \mathbf{1}_{\{B_m(s_m)<\tilde{d}_0(m),\ \cdots,\ B_m(s_m)+\cdots+B_l(s_l^m)<\tilde{d}_0(l)\}}. \tag{45}$$

Our next step is to get the lower bound of $E[N_l]$. The following lemma is useful for this step.

*Lemma 5:* The volume $\mathrm{vol}(x_1^n)$ of non-negative values $x_1, x_2, ..., x_n$ satisfying the following condition

$$\begin{aligned} x_1 &< a_1 \\ x_1 + x_2 &< a_2 \\ &\cdots \\ x_1 + x_2 + \cdots + x_n &< a_n \end{aligned} \tag{46}$$

is always larger than the volume $\mathrm{vol}'(x_1^n)$ satisfying

$$\frac{x_1}{a_1} + \frac{x_2}{a_2} + \cdots \frac{x_n}{a_n} < 1 \tag{47}$$

when $0 < a_1 < a_2 < \cdots < a_n$.

*Proof:* Multiplying $a_1$ into (47)

$$x_1 < a_1 - (\frac{a_1}{a_2}x_2 + \cdots \frac{a_1}{a_n}x_n) < a_1$$

Thus the interval of $x_1$ satisfying (47) is smaller than (46). In a similar way, multiplying $a_2$ into (47) then

$$x_1 + x_2 < a_2 - (\frac{a_2}{a_1} - 1)x_1 - \frac{a_2}{a_3}x_3 - \cdots - \frac{a_2}{a_n}x_n < a_2$$

where $\frac{a_2}{a_1} > 1$. By repeating this way, one can easily show that the interval of $x_k$ satisfying (46) is always larger than that of (47). ■

Using Lemma 5, we can get the lower bound of $E[N_l]$.

*Lemma 6:* The expected complexity at the $(m-l+1)$-th layer $E[N_l]$ is lower bounded by

$$E[N_l] \geq L^{m-l+1} \cdot \max\left(0,\ 1 - \left(\sum_{k=l}^{m}\frac{1}{\tilde{d}_0(k)}\right)(\frac{2m}{3\tau^2}(L^2-1)\sigma_h^2 + \sigma_v^2)\right) \tag{48}$$

*Proof:* See Appendix C. ■

Using Lemma 6, we can get the lower bound of the expected complexity of the PTP-SD algorithm.

*Theorem 7:* The expected complexity $E[N]$ is lower bounded by

$$E[N] \geq \sum_{l=1}^{m} L^{m-l+1} \cdot \max\left(0,\ 1 - \frac{(m-l+1)(\frac{4\xi}{3\tau^2}(L^2-1)+1)\sigma_v^2}{\bar{d}_l'}\right) \tag{49}$$

*Proof:* By denoting $\frac{1}{\bar{d}_l'} = \frac{1}{m-l+1}\sum_{i=l}^{m}\frac{1}{\tilde{d}_0(i)}$, (48) is rewritten as

$$E[N_l] \geq L^{m-l+1} \cdot \max\left(0, 1 - \frac{(m-l+1)\left(\frac{2m}{3\tau^2}(L^2-1)\sigma_h^2 + \sigma_v^2\right)}{\bar{d}_l'}\right)$$

Using (37) and noting that $E[N] = \sum_{l=1}^{m} E(N_l)$, we get (49). ■

It is interesting to compare (42) and (49), which are the lower bounds for the SD and PTP-SD algorithms, respectively. The term $\frac{1}{d_0}$ of the SD algorithm is replaced by the arithmetic mean of $\frac{1}{\tilde{d}_0(l)}$, ..., $\frac{1}{\tilde{d}_0(m)}$ in the PTP-SD algorithm, i.e.,

$$\begin{aligned} \frac{1}{\bar{d}_l'} &= \frac{1}{m-l+1}\sum_{i=l}^{m}\frac{1}{\tilde{d}_0(i)} && (50) \\ &= \frac{1}{m-l+1}\sum_{i=l}^{m}\frac{1}{d_0 - \beta_{i-1}} && (51) \end{aligned}$$

where $\beta_0 = 0$. Since the complexity decreases with $\bar{d}_l'$ and also recall that $\beta$ is an increasing function of $P_\epsilon$, we can deduce that the complexity is reduced by increasing $P_\epsilon$. Notice also that as discussed in Section III.B, $P_\epsilon = 0$ will result in the same expression for (42) and (49).

[2] The ML solution can be found in $m$ steps and $m-1$ more steps are needed for termination so that $E[N] \sim 2m-1$ for very high SNR scenarios.

[3] Since (42) is the lower bound, even if $d_0 < (\frac{4\xi}{3\tau^2}(L^2-1)+1)\sigma_v^2$ and hence $E[N] \geq 0$, it does not necessarily guarantee the small complexity in a rigorous sense.

## V. Simulation and Discussions

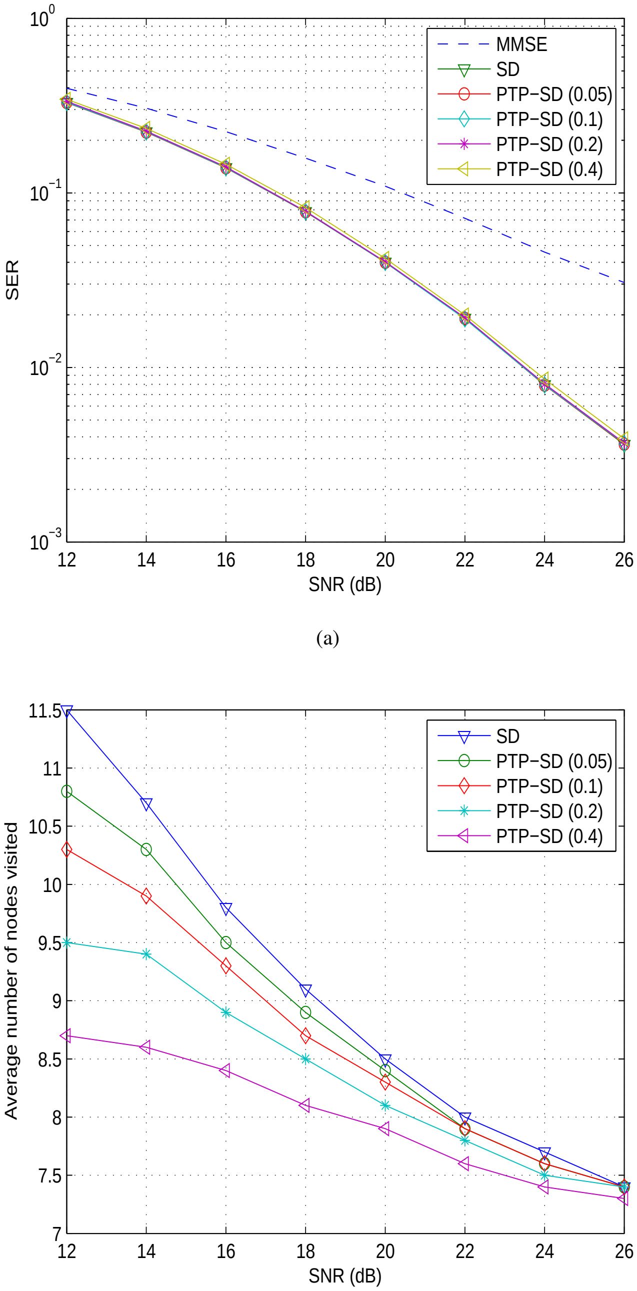
### A. Simulation Setup

In this section, we observe the performance and complexity of the proposed PTP-SD algorithm over the SD algorithm as well as the MMSE equalizer. We also present the comparison between PTP-SD and IRA [13]. The simulation setup is based on the $L^2$-QAM transmission over MIMO (multiple-input multiple-output) systems in Rayleigh fading channel, where the $\mathbf{H}$ matrix is modeled by independent Gaussian random variables. We use $2 \times 2$, $4 \times 4$, and $10 \times 10$ MIMO system and the levels of QAM are $L = 4$ (16-QAM), $L = 8$ (64-QAM), and $L = 2$ (QPSK), respectively. The PTP-SD and the reference SD algorithm in our study employs the depth-first search and the SE enumeration. Hence, we set the initial radius to $d_0 = \infty$ and update it whenever a new candidate $\mathbf{s}$ is found (i.e., if $d_0 \geq ||\,\mathbf{y}' - \mathbf{Rs}\,||^2$). In addition, we consider four different pruning probabilities ($P_\epsilon = 0.05$, $0.1$, $0.2$, and $0.4$) for observing the trade-off between the performance and complexity. As a metric for measuring performance and complexity, we employ the symbol error rate (SER) and the average number of nodes visited. We set our target SER to $10^{-2}$ and run at least $20,000$ channel realization for each SNR point.

### B. Simulations

We first consider a $2 \times 2$ MIMO system with 16-QAM modulation. In this case, $L = 4$ and $m = 4$ after conversion to the real number signal model. Hence, the worst case complexity becomes $4^4 = 256$. As shown in Fig. 4(a), as SNR increases, the SD algorithm provides considerable gain over the MMSE equalizer resulting in more than 3 dB gain at SER of $10^{-1}$. The SD and PTP-SD algorithms, however, lie on top of each other so that the performance loss of the PTP-SD algorithm is negligible for all the pruning probabilities we tested. Figure 4(b) plots the average number of nodes visited for the SD and PTP-SD algorithms. Generally, we observe that the complexity of the PTP-SD algorithm is smaller than that of the SD algorithm and becomes smaller as $P_\epsilon$ increases. For example, the complexity reduction at SNR$=12$ dB is around $25\%$ with $P_\epsilon = 0.4$. When the SNR increases, as discussed in Section IV-A, the SD algorithm tends to track the ML path directly so that the complexity benefit of the PTP-SD becomes smaller.

Next, we consider a $4 \times 4$ MIMO system with 64-QAM modulation, which results in $m = 8$ and $L = 8$ in the real number signal model. The worst case complexity becomes $L^m = 16,777,216$. As shown in Fig. 5(a), the performance gain of the SD algorithm over the MMSE equalizer is noticeably better than that of the $2 \times 2$ MIMO system providing almost 6 dB gain at SER of $10^{-1}$. In addition, we see that the performance difference between the SD and PTP-SD algorithms is negligible till $P_\epsilon = 0.2$. Even at $P_\epsilon = 0.4$, the performance loss of PTP-SD is around $0.4$ dB. The complexity curves in Fig. 5(b) shows similar but better results than the $2 \times 2$ scenario so that the PTP-SD algorithm shows considerable savings, particularly at low and mid SNR. The complexity reduction for $P_\epsilon = 0.2$ and $0.4$ at 20 dB SNR are 28 and $33\%$, respectively.

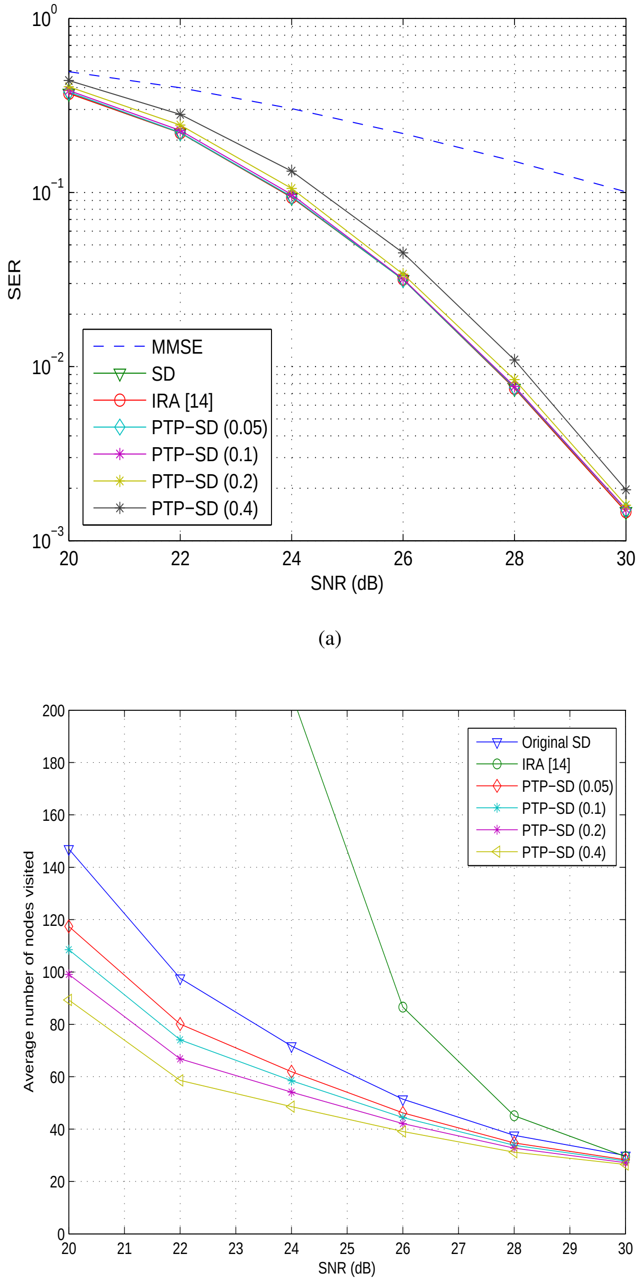


(a)

(b)

Fig. 4. Performance and complexity of SD algorithms for $2 \times 2$ MIMO system with 16-QAM modulation: (a) SER and (b) complexity.

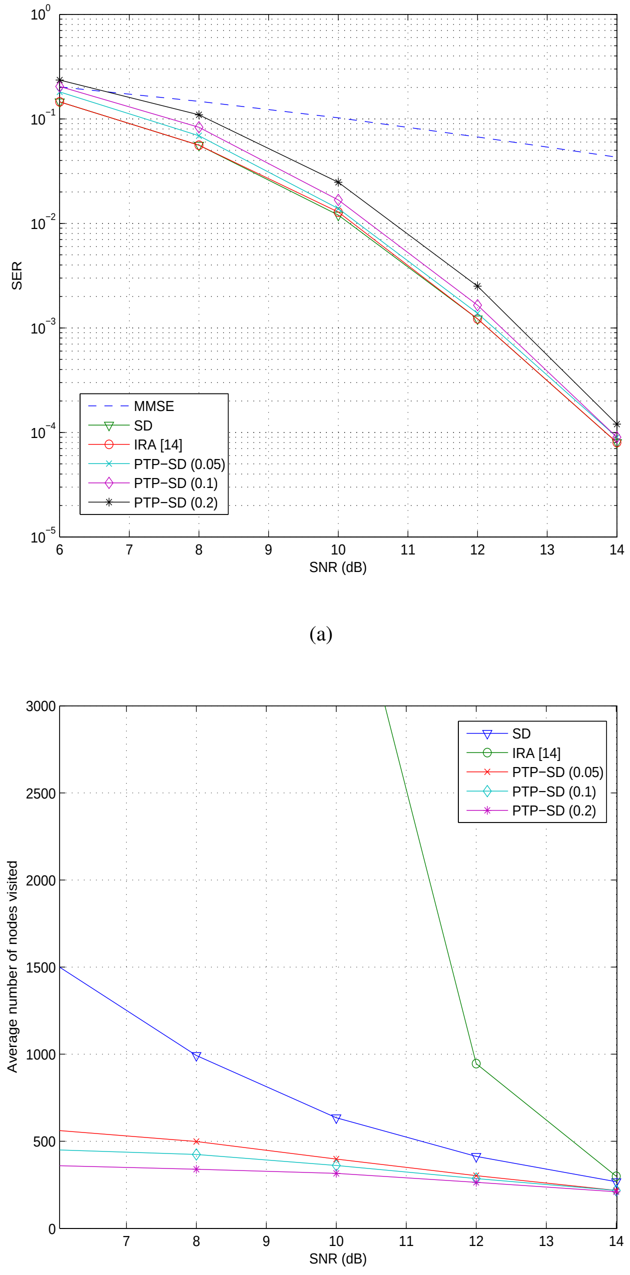
We also examine the $10 \times 10$ MIMO system with QPSK modulation ($m = 20$ and $L = 2$). Although the worst case complexity is smaller than the previous case ($2^{20} < 8^8$), the complexity increase is substantial as shown in Fig. 6(b) since the SE enumeration is less efficient for QPSK modulation than 64-QAM modulation,. For this case where the depth is far larger than the width of the tree ($m \gg L$), pruning operations become more effective. In fact, as seen from Fig. 5(b), the



(a)

(b)

Fig. 5. Performance and complexity of SD algorithms for $4 \times 4$ MIMO system with 64-QAM modulation: (a) SER and (b) complexity.



(a)

(b)

Fig. 6. Performance and complexity of SD algorithms for $10 \times 10$ MIMO system with QPSK modulation: (a) SER and (b) complexity.

complexity reduction of the PTP-SD for the $10 \times 10$ system is noticeably larger than that for the $4 \times 4$ system resulting in $76\%$ and $50\%$ reduction in the complexity at 6 dB and 10 dB SNR, respectively, for $P_\epsilon = 0.2$. The price is the performance loss so that $0.7$ dB loss is observed at the target SER. Note that to achieve the performance close to ML receiver, one can choose smaller $P_\epsilon$ at the expense of slight increase in complexity.

### C. Comparisions with IRA

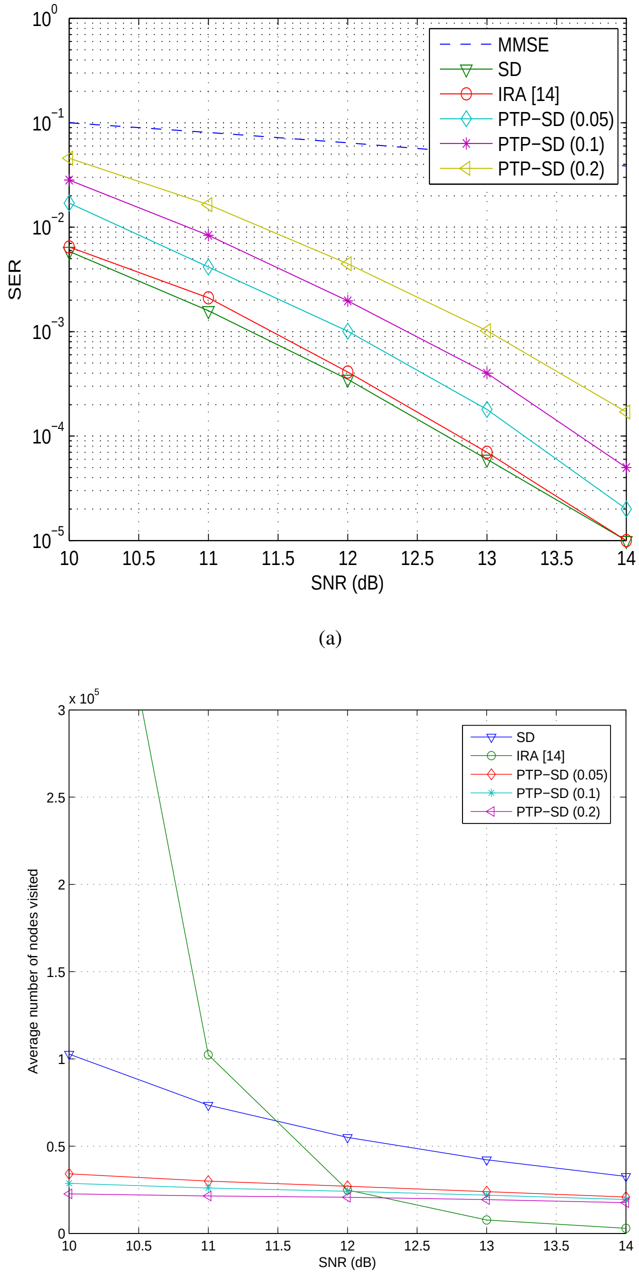
In this subsection, we compare the PTP-SD with IRA method [12], [13]. The $\epsilon$ and $\delta$ parameters for the radius schedule of the IRA are chosen based on Table V in [13]. Although the two methods are similar in principle, the resulting sphere radius per layer is distinct since the radius selection of the PTP-SD is developed from the pruning probability and that of the IRA is derived from the probability that the detected symbol is not the ML solution. Further, since the IRA relies on pre-defined radius schedule and the radius of the PTP-SD is updated whenever a new candidate is found, we observe the clear difference in the performance and complexity. Indeed, as displayed in Fig. 5, 6, and 7, PTP-SD has a benefit in complexity while the IRA is closer to the ML performance. Also, it is interesting to note that the speed of complexity reduction for the IRA is faster than that of the PTP-SD, in particular for large system, so that the IRA method seems to be useful for high SNR regime (e.g., $m \geq 20$ and required SER $< 10^{-3}$). This is intuitively reasonable since the Babai point can be a loose initial point for large systems so that properly designed fixed radius schedule might be better than the dynamic radius update with an arbitrary large initial radius. However, since the complexity of the PTP-SD is smaller than the IRA when SER $> 10^{-3}$, PTP-SD might be more useful for communication systems operating at low and middle SNR regimes.

### *D. Simulation vs. Lower Bound Analysis*

Finally, we compare the simulation results and lower bound analysis in Section IV. In Fig. 8, we plotted the results for system with $20 \times 20$ MIMO and QPSK modulation. We observe that the lower bound results are somewhat loose possibly due to the simplification in the derivation (e.g., Markov inequality). We can, however, see the agreement between the lower bound analysis and the simulation for the relative difference between SD and PTP-SD. In the low and mid SNR regime, the amount of reduction in complexity is considerable and it becomes smaller as the SNR increases. Since the simulation below the tested SNR range ($<$ 10 dB) is extremely time consuming, derived expressions can be used as a tool for checking the advantage of the PTP-SD over the SD algorithm.

## VI. Concluding Remarks

In this paper, we investigated a reduced complexity SD algorithm referred to as probabilistic tree pruning sphere decoding (PTP-SD). Motivated by the fact that the SD algorithm does not fully utilize the sphere constraint during the early layers of sphere search, we employed a probabilistic noise constraint that tightens the sphere constraint at each layer. This strengthens the ability to prune the branches unlikely to be chosen as the ML solution. From the simulation, we observed that the performance loss of the PTP-SD algorithm over the SD algorithm is negligible while providing considerable reduction in complexity. Other than the MIMO applications we considered in this paper, the PTP-SD algorithm can be extended to many applications requiring lower complexity such as MLSE for multi-dimensional constellation systems, MIMO MAP detectors with a priori feedback information [24], or multi-stage decoding [27]. Also, PTP-SD can be combined with preprocessing techniques such as the basis reduction and detection ordering [5], [25], [26] to achieve further reduction in complexity.



(a)

(b)

Fig. 7. Performance and complexity of SD algorithms for $20 \times 20$ MIMO system with QPSK modulation: (a) SER and (b) complexity.

## Appendix A
## Proof of Lemma 3

From (19), we have $B_i(\tilde{\mathbf{s}}_i^m, \ \tilde{\mathbf{v}}_i^m, \ \mathbf{s}_i^m) = (\sum_{k=i}^m r_{ik}(\tilde{s}_k - s_k) + \tilde{v}_k)^2$ and thus

$$\begin{aligned} E[B_i(\tilde{\mathbf{s}}_i^m, \ \tilde{\mathbf{v}}_i^m, \ \mathbf{s}_i^m \ )] &= E\left[(\sum_{k=i}^m r_{ik}(\tilde{s}_k - s_k) + \tilde{v}_k)^2\right] \\ &= \sum_{k=i}^m E[r_{ik}^2] E[(\tilde{s}_k - s_k)^2] + \sigma_v^2 \\ &\overset{(a)}{=} \frac{2}{3\tau^2}(L^2 - 1) \sum_{k=i}^m E[r_{ik}^2] + \sigma_v^2 \qquad (52) \end{aligned}$$

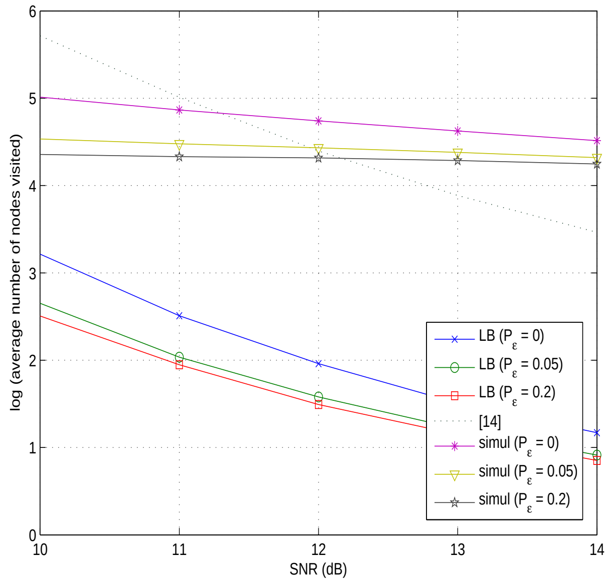


Fig. 8. Expected number of nodes visited: lower bound vs. simulation.

where $(a)$ is from Lemma 2.

Since $P_l(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m) = \sum_{i=l}^m B_i(\tilde{\mathbf{s}}_i^m,\ \tilde{\mathbf{v}}_i^m,\ \mathbf{s}_i^m)$, we further have

$$\begin{aligned} E[P_l(\tilde{\mathbf{s}}_l^m,\ \ \tilde{\mathbf{v}}_l^m\ \ ,\ \mathbf{s}_l^m)] &= E[\sum_{i=l}^m B_i(\tilde{\mathbf{s}}_i^m,\ \tilde{\mathbf{v}}_i^m,\ \mathbf{s}_i^m)] \\ &= \sum_{i=l}^m E[B_i(\tilde{\mathbf{s}}_i^m,\ \tilde{\mathbf{v}}_i^m,\ \mathbf{s}_i^m)] \\ &\overset{(a)}{=} \frac{2}{3\tau^2}(L^2-1)\sum_{i=l}^m\sum_{k=i}^m E[r_{ik}^2] + (m-l+1)\sigma_v^2 \\ &\overset{(b)}{=} \frac{2}{3\tau^2}(L^2-1)\sum_{k=l}^m\sum_{i=l}^k E[r_{ik}^2] + (m-l+1)\sigma_v^2 \\ &\overset{(c)}{\le} \frac{2}{3\tau^2}(L^2-1)\sum_{k=l}^m E[||\mathbf{R}_k||^2] + (m-l+1)\sigma_v^2 \\ &\overset{(d)}{=} \frac{2}{3\tau^2}(L^2-1)\sum_{k=l}^m E[||\mathbf{H}_k||^2] + (m-l+1)\sigma_v^2 \\ &= \frac{2m}{3\tau^2}(L^2-1)(m-l+1)\sigma_h^2 + (m-l+1)\sigma_v^2 \end{aligned}$$

where $\mathbf{R}_k$ and $\mathbf{H}_k$ are the $k$-th columns of $\mathbf{R}$ and $\mathbf{H}$ matrices, respectively. Note that $(a)$ is from (52) and $(b)$ is the change of the summation indices over the lower triangular portion of $\mathbf{R}$ matrix. $(c)$ is from the fact that

$$E[||\mathbf{R}_k||^2] = \sum_{i=1}^k E[r_{ik}^2] \ge \sum_{i=l}^k E[r_{ik}^2], \tag{53}$$

and $(d)$ is because $\mathbf{R}_k = \mathbf{Q}^T\mathbf{H}_k$ and hence $E[||\mathbf{R}_k||^2] = E[||\mathbf{Q}^T\mathbf{H}_k||^2] = E[||\mathbf{H}_k||^2]$.

# Appendix B
# Proof of Lemma 6

From (45), we have

$$\begin{aligned} E[N_l] &= E[\sum_{s_l^m} \mathbf{1}_{\{B_m<\tilde{d}_0(m),\cdots,B_m+\cdots+B_l<\tilde{d}_0(l)\}}] \\ &\overset{(a)}{=} \sum_{s_l^m} P_r(B_m<\tilde{d}_0(m),\ \cdots, B_m+\cdots+B_l<\tilde{d}_0(l)) \\ &\overset{(b)}{\ge} \sum_{s_l^m} P_r(\frac{B_m}{\tilde{d}_0(m)} + \frac{B_{m-1}}{\tilde{d}_0(m-1)} + \cdots + \frac{B_l}{\tilde{d}_0(l)} < 1) \end{aligned}$$

where the random variables $(\tilde{\mathbf{s}}_l^m,\ \tilde{\mathbf{v}}_l^m)$ are omitted for notational convenience and $(a)$ is by (40). Since $\beta_{k-1}$ is decreasing for the same pruning probability for all layers, $\tilde{d}_0(k) = d_0 - \beta_{k-1}$ becomes an increasing sequence and thus $(b)$ holds by Lemma 5. Since $\frac{B_m}{\tilde{d}_0(m)} + \frac{B_{m-1}}{\tilde{d}_0(m-1)} + \cdots + \frac{B_l}{\tilde{d}_0(l)}$ is a function of $\mathbf{s}_l^m$, $\tilde{\mathbf{s}}_l^m$, and $\tilde{\mathbf{v}}_l^m$, we hereby denote it as $g(\mathbf{s}_l^m, \tilde{\mathbf{s}}_l^m, \tilde{\mathbf{v}}_l^m)$. Then, we have

$$\begin{aligned} E[N_l] &\ge \sum_{s_l^m} P_r(\,g(\mathbf{s}_l^m, \tilde{\mathbf{s}}_l^m, \tilde{\mathbf{v}}_l^m) < 1) \\ &\ge L^{m-l+1} P_r(\,g(\mathbf{s}_l^m, \tilde{\mathbf{s}}_l^m, \tilde{\mathbf{v}}_l^m) < 1) \end{aligned} \tag{54}$$

Similar to Lemma 1, by using the Markov inequality, (54) becomes

$$E[N_l] \ge L^{m-l+1}\max(0,\ 1 - E[g(\mathbf{s}_l^m, \tilde{\mathbf{s}}_l^m, \tilde{\mathbf{v}}_l^m)]). \tag{55}$$

Recall from (19) that

$$E[B_i(\tilde{\mathbf{s}}_i^m,\ \tilde{\mathbf{v}}_i^m,\ \mathbf{s}_i^m)] = \frac{2}{3\tau^2}(L^2-1)\sum_{k=i}^m E[r_{ik}^2] + \sigma_v^2 \tag{56}$$

and hence we have

$$\begin{aligned} E[\ g(\tilde{\mathbf{s}}_l^m\ \ ,\ \tilde{\mathbf{v}}_l^m,\ \mathbf{s}_l^m)] &= \sum_{i=l}^m \frac{E[B_i(\tilde{\mathbf{s}}_i^m,\ \ \tilde{\mathbf{v}}_i^m,\ \ \mathbf{s}_i^m)]}{\tilde{d}_0(i)} \\ &= \frac{2}{3\tau^2}(L^2-1)\sum_{i=l}^m\sum_{k=i}^m \frac{E[r_{ik}^2]}{\tilde{d}_0(i)} + \sigma_v^2\sum_{i=l}^m \frac{1}{\tilde{d}_0(i)} \\ &= \frac{2}{3\tau^2}(L^2-1)\sum_{k=l}^m\sum_{i=l}^k \frac{E[r_{ik}^2]}{\tilde{d}_0(i)} + \sigma_v^2\sum_{i=l}^m \frac{1}{\tilde{d}_0(i)} \\ &\overset{(a)}{\le} \frac{2}{3\tau^2}(L^2-1)\sum_{k=l}^m\sum_{i=l}^k \frac{E[r_{ik}^2]}{\tilde{d}_0(k)} + \sigma_v^2\sum_{i=l}^m \frac{1}{\tilde{d}_0(i)} \\ &\overset{(b)}{\le} \frac{2}{3\tau^2}(L^2-1)\sum_{k=l}^m \frac{E[||\mathbf{R}_k||^2]}{\tilde{d}_0(k)} + \sigma_v^2\sum_{i=l}^m \frac{1}{\tilde{d}_0(i)} \\ &= \frac{2}{3\tau^2}(L^2-1)\sum_{k=l}^m \frac{E[||\mathbf{H}_k||^2]}{\tilde{d}_0(k)} + \sigma_v^2\sum_{i=l}^m \frac{1}{\tilde{d}_0(i)} \\ &= \frac{2m}{3\tau^2}(L^2-1)\sigma_h^2\sum_{k=l}^m \frac{1}{\tilde{d}_0(k)} + \sigma_v^2\sum_{i=l}^m \frac{1}{\tilde{d}_0(i)} \end{aligned}$$

where $\mathbf{R}_k$ and $\mathbf{H}_k$ are the $k$-th columns of $\mathbf{R}$ and $\mathbf{H}$ matrices, respectively. Notice that $(a)$ is true because $\tilde{d}_0(k) = d_0 - \beta_{k-1} \le \tilde{d}_0(i) = d_0 - \beta_{i-1}$ for $i \le k$ and $(b)$ is by (53). This completes the proof.

## ACKNOWLEDGMENT

The authors would like to thank R. Gowaikar, W. Zhao, and J. Choi for helpful discussions, and the associate editor and the anonymous reviewers and for their careful reading and valuable suggestions that improved the presentation of the paper.

## REFERENCES

[1] 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.

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

[3] E. Viterbo, E. Biglieri, "A universal lattice decoder," in *GRETSI 14-eme Colloque*, Jun-les-pins France, Sept. 1993.

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

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

[6] M. O. Damen, H. Gamel, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2389-2402, Oct. 2003.

[7] E. Viterbo, J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1639-1642, July 1999.

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

[9] B. Hassibi and H. Vikalo, "The expected complexity of sphere decoding Part I: Theory," *IEEE Trans. Signal Proc.*, vol. 53, pp. 2806-2818, Aug. 2005.

[10] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm II. Generalizations, second-order statistics, and applications to communications," *IEEE Trans. Signal Proc.*, vol. 53, pp. 2819-2834, Aug. 2005.

[11] W. Zhao and G. B. Giannakis, "Sphere decoding algorithms with improved radius search," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1104-1109, July 2005.

[12] R. Gowaikar and B. Hassibi, "Efficient Statistical Pruning for Maximum Likelihood Decoding", *IEEE ICASSP*, Vol. 5, pp. 49-52, 2003.

[13] R. Gowaikar and B. Hassibi, "Statistical Pruning for Near-Maximum Likelihood Decoding", *IEEE Trans. Sig. Proc.*, vol. 55, no. 6, pp. 2661-2675, June 2007.

[14] E. Telatar, "Capacity of multi-antenna Gaussian channels,". *European Trans. on Telecommunications*, vol. 10 pp. 585-596, Nov. 1999.

[15] G. J. Foschini and M. J. Gans, "On Limits of Wireless Communication a Fading Environment when Using Multiple Antennas," *Wireless Personal Communications*, vol. 6, pp. 311-335, March 1998.

[16] D. Gesbert, M. Shafi, D. Shiu, P. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 281-302, April 2003.

[17] L. Babai, *On Lovasz' lattice reduction and the nearest lattice point problem*, Combinatorica, vol. 6, pp. 1-13, 1986.

[18] S. M. Ross, *Probability Models*, Academic Press, 2000.

[19] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Proc.*, vol. 53, pp. 1474-1484, April 2005.

[20] B. Hassibi, "An efficient square-root algorithm for BLAST," in *Proc. Int. Conf. Acoustic, Speech, Signal Processing*, June 2000, pp. 5-9.

[21] D. Wubben, R. Bohnke, V. Kuhn, and K. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE VTC*, Oct. 2003, pp. 508-512.

[22] G. H. Golub and C. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1993.

[23] T. K. Moon, W. C. Striling, *Mathematical methods and algorithms for signal processing*, Prentice Hall, 1999.

[24] B. M. Hochwald and S. Brink, "Achieving near-capacity on a multiple-antenna channels," *IEEE Trans. on Comm.*, vol. 51, pp. 389-399, March 2003.

[25] W. Zhao and G. B. Giannakis, "Reduced complexity closest point decoding algorithms for random lattices," *IEEE Trans. Wireless Commun.*, vol. 5, no. 1, pp. 101-111, Jan. 2006.

[26] A. D. Murugan, H. El Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering the sequential decoder," *IEEE Trans. Inform. Theory*, Mar. 2006.

[27] T. Cui and C. Tellambura, "Approximate ML detection for MIMO systems using multistage sphere decoding," *IEEE Sig. Proc. Letters*, vol. 12, pp. 222-225, March 2005.

PLACE
PHOTO
HERE

**Byonghyo Shim (Member, IEEE)** received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University, Korea, in 1995 and 1997, respectively and the M.S. degree in mathematics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 2004 and 2005, respectively.

From Sept. 1997 to June 2000, He was an Officer and an Full-time Lecturer in the department of electronics engineering at Korean AirForce Academy, and from Jan. 2005 to Sept. 2007, he was a Staff Engineer in the modem group at Qualcomm Inc., Inc., San Diego. Since Sept. 2007, he has been an Assistant Professor in the Dept. of Computer Science at Korea University, Seoul, Korea. He has also held short-term research position in DSP group of LG electronics and DSP R&D center of Texas instruments. His research interests include signal processing for communication, estimation and detection, and applied linear algebra.

Dr. Shim was the recipient of 2005 M. E. Van Valkenburg research award from ECE department of University of Illinois. He is a member of Sigma Xi and Tau Beta Pi.

PLACE
PHOTO
HERE

**Insung Kang (Member, IEEE)** was born in Pusan, Korea, in 1960. He received the B.S. degree in control and instrumentation engineering from Seoul National University, Korea, in 1982, the M.S. degree in electrical engineering from the University of Dayton, OH, in 1993, and the Ph.D. degree in electrical and computer engineering from Purdue University, IN, in 1997, respectively.

From 1982 to 1992, he was a Research Engineer and Manager at a subsidiary of LG Electronics Inc. working on many projects including super conductive MRI-CT and H.261-based video phone. From 1997 to 2000, he was with the Network Solutions Sector of Motorola, Inc., where he worked on WCDMA base station system. From 2000 to 2002, he was with Sorrento Telecom, Inc., where he designed the CDMA2000 mobile station receivers. From 2003 to 2006, he was with Via Telecom, Inc., where he designed the CDMA2000 EVDV mobile station receiver. Since August 2006, he has been with Qualcomm, Inc. where he is working on LTE, WIMAX, and DVB-H system. His current research interests include MIMO receiver design, iterative decoding, and interference cancellation.