Soft-input Soft-output List Sphere Detection with a Probabilistic Radius Tightening

Jaeseok Lee, *Student Member, IEEE,*, Byonghyo Shim, *Senior Member, IEEE,*, and Insung Kang, *Member, IEEE,*

***Abstract*—In this paper, we present a low-complexity list sphere detection algorithm for achieving near-optimal *a posteriori probability* (APP) detection in an iterative detection and decoding (IDD). Motivated by the fact that the list sphere decoding searching a fixed number of candidates is computationally inefficient in many scenarios, we design a criterion to search lattice points with non-vanishing likelihood and then derive a hypersphere radius satisfying this condition. Further, in order to exploit the original sphere constraint as it is instead of using necessary conditioned version, we combine a probabilistic tree pruning strategy and the proposed list sphere search. Two features, tightened hypersphere radius and probabilistic tree pruning, collaborate and improve the search efficiency in a complementary fashion. Through simulations on 4×4 MIMO system, we show that the proposed method provides substantial reduction in complexity while achieving negligible performance loss over the conventional list sphere detection.**

***Index Terms*—Sphere decoding, Iterative detection and decoding, a posteriori probability, probabilistic radius tightening, multiple-input multiple-output system, complexity reduction.**

I. Introduction

As a means for achieving near-capacity performance in multiple-input multiple-output (MIMO) systems, *iterative detection and decoding* (IDD) has received much attention recently [2], [3]. By exchanging the soft information so called *extrinsic* information between the MIMO detector and the channel decoder which serves as *a priori* information to each other, IDD jointly improves the detection quality as well as the decoding reliability. The key ingredient of the IDD receiver is the *a posteriori probability* (APP) detector generating the posteriori log-likelihood ratio (LLR) which is given by

$$L(b_k) = \ln \frac{P(b_{k+} | \, \mathbf{y})}{P(b_{k-} | \, \mathbf{y})} \tag{1}$$

where b_k is the k-th information bit, $P(b_{k+})$ and $P(b_{k-})$ are the probabilities that $b_k = +1$ and $b_k = -1$, respectively, and $\mathbf{y}$ is the real-valued observation vector. Typically, $\mathbf{y} = \mathbf{H}\mathbf{s}+\mathbf{v}$ is obtained from the complex-valued observation $\mathbf{y}_c = \mathbf{H}_c\mathbf{s}_c + \mathbf{v}_c$ via the complex-to-real conversion where $\mathbf{s} \in \Lambda^\ell$ (Λ is a modulation set) and $\mathbf{v} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ are the transmitted symbol vector and Gaussian noise vector, respectively, and $\mathbf{H}$ is the channel matrix. Since $\mathbf{y}$ is connected to the bit vector $\mathbf{b}$ through $\mathbf{s}$ (i.e., $\mathbf{s} = f(\mathbf{b})$ where f is the function mapping a bit vector into a symbol vector), (1) can be rewritten as

$$L(b_k) = \ln \frac{P(b_{k+})}{P(b_{k-})} + \ln \frac{\sum_{\mathbf{b}_{k+}} P(\mathbf{y} \,|\, \mathbf{s}_{k+}) P(\mathbf{b}_{\bar{k}})}{\sum_{\mathbf{b}_{k-}} P(\mathbf{y} \,|\, \mathbf{s}_{k-}) P(\mathbf{b}_{\bar{k}})} \tag{2}$$

where $\mathbf{b}_{\bar{k}} = [b_1, \cdots, b_{k-1}, b_{k+1}, \cdots]$, $\mathbf{b}_{k+} = [b_1, \cdots, b_{k-1}, b_{k+}, b_{k+1}, \cdots]$ and $\mathbf{s}_{k+} = f(\mathbf{b}_{k+})$ ($\mathbf{b}_{k-}$ and $\mathbf{s}_{k-}$ are defined in the same way). Since the first term of (2) corresponds to the *a priori* information (denoted as $L_A(b_k)$) delivered from the channel decoder, the second term of (2) becomes an extrinsic LLR information $L_E(b_k)$ of the APP detector, which is rewritten as [2]

$$L_E(b_k) = \ln \frac{\sum_{\mathbf{s}_{k+}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}_{k+}\|^2}{2\sigma^2} + \frac{1}{2}\mathbf{b}_{\bar{k}}^T \mathbf{L}_{A,\bar{k}}\right)}{\sum_{\mathbf{s}_{k-}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}_{k-}\|^2}{2\sigma^2} + \frac{1}{2}\mathbf{b}_{\bar{k}}^T \mathbf{L}_{A,\bar{k}}\right)} \tag{3}$$

where $\mathbf{L}_{A,\bar{k}} = [\, L_A(b_1), \cdots, L_A(b_{k-1}), L_A(b_{k+1}), \cdots]^T$. The detail of these manipulations is straight forward and can be found in [3].

As an efficient way to achieve a good approximation of (3), a non-trivial modification of the sphere decoding (SD) referred to as the list sphere decoding (LSD) has been suggested [2]. While the SD searches the single best symbol vector, i.e., the maximum-likelihood (ML) solution $\mathbf{s}_{ml}$ [4]–[6], the LSD finds N best symbol vectors and stores them into the list $\mathcal{L} = \{\mathbf{s}_1 = \mathbf{s}_{ml}, \ \mathbf{s}_2, \cdots, \mathbf{s}_N\}$ where $\|\mathbf{y} - \mathbf{H}\mathbf{s}_i\|^2 \leq \|\mathbf{y} - \mathbf{H}\mathbf{s}_j\|^2$ for $i \leq j$. Once the list search is finished, for each k, $\mathcal{L} = \{\mathbf{s}_1 = \mathbf{s}_{ml}, \mathbf{s}_2, \cdots, \mathbf{s}_N\}$ is divided into $\mathcal{L}_{k+}$ and $\mathcal{L}_{k-}$ where $\mathcal{L}_{k+}$ and $\mathcal{L}_{k-}$ are the set of symbol vector generated from b_{k+} and b_{k-}, respectively. Then the LLR of the k-th bit in (3) can be approximated as

$$L_E(b_k) \approx \ln \frac{\sum_{\mathbf{s}_{k+} \in \mathcal{L}_{k+}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}_{k+}\|^2}{2\sigma^2} + \frac{1}{2}\mathbf{b}_{\bar{k}}^T \mathbf{L}_{A,\bar{k}}\right)}{\sum_{\mathbf{s}_{k-} \in \mathcal{L}_{k-}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}_{k-}\|^2}{2\sigma^2} + \frac{1}{2}\mathbf{b}_{\bar{k}}^T \mathbf{L}_{A,\bar{k}}\right)}. \tag{4}$$

In spite of the remarkable complexity savings over an exhaustive enumeration in (3), computational overhead of this approach, mainly caused by the LSD operation, is still considerable. It is worth emphasizing that while the goal of the SD algorithm is to find the best lattice point $\mathbf{s}_{ml}$ and thus it is allowed to shrink the sphere radius whenever a new candidate lattice point satisfying the sphere constraint is found, such is not possible for the LSD since N best lattice points should be found and stored in the list. The sphere radius of the LSD is updated only when the list is filled with N lattice points and a new candidate replacing the existing one is found since otherwise N best lattice points cannot

J. Lee and B. Shim is with School of Information and Communication, Korea University, Seoul, Korea (email: {jslee,bshim}@isl.korea.ac.kr).

I. Kang is with Qualcomm Inc., CA 92121 USA (email: insungk@qualcomm.com).

This work is supported by Basic Science Research Program through National Research Foundation (NRF) funded by the MEST (No. 2011-0012525) and second Brain Korea 21 project.

A part of this paper was presented in Globecom 2010 [1].

be obtained. Accordingly, in many realistic scenarios where the list size N is non-trivial (such as system employing high order modulation (HOM) and/or search dimension is large), computational complexity of the N-LSD is quite demanding.

There have been number of studies that attempt to lessen the complexity of LSD based APP detection. In [2, Eq. (12)], a max-log approximation of $L_E(b_k)$ over $\mathcal{L}$ was presented. Focusing on the improvement of the tree-search efficiency, a modification of the sphere search algorithm was presented in [7]. In [8] a modified max-log approach searching two candidates, the ML solution and its binary complement, was proposed and extensions of this idea were presented in [9]. In these approaches, the cost functions of $\mathbf{s}_{ml}$ and its binary complement are used to generate the LLR. In order to alleviate the complexity associated with the binary complement search, various heuristics such as LLR clipping, channel matrix regularization, and imposing constraint on the maximum computational complexity were exploited. In [15], an approach to search the lattice point around the ML point was suggested. This approach, referred to as the shifted spherical list detector, is based on the ML search followed by the simple lattice point search. Additionally, extensions of K-algorithm [10] and M-algorithm [11], [12] for limiting computational complexity were proposed. Besides, approaches pursuing fixed complexity in the sphere search were also proposed in [13], [14].

In this paper, we put forth an approach optimizing the sphere radius for pursuing a reduction of complexity in the hypersphere search based APP detection, where the benefit of searching N best lattice points is approximately preserved. Towards this end, we divide the proposed method into two stages; 1) the SD based ML detection followed by the optimal sphere radius R_0 selection for the list search and 2) the LSD with the probabilistic tree pruning. Motivated by the observation that the effect of symbol vector $\mathbf{s}$ whose likelihood $P(\mathbf{y}\,|\,\mathbf{s})$ is much smaller than $P(\mathbf{y}\,|\,\mathbf{s}_{ml})$ on $L_E(b_k)$ is negligible, we set up a condition for the radius selection and then derive an optimal (minimal) sphere radius R_0 satisfying this condition. Since the radius is expressed as a function of $\mathbf{s}_{ml}$, we employ the SD in the first stage to search $\mathbf{s}_{ml}$ and then perform the LSD using R_0. It is worth noting that notwithstanding the use of tightened sphere radius, necessary condition imposed by the sphere constraint is not so effective in early layers of the search. In order to enhance the search efficiency of the LSD, we incorporate a mechanism pruning unpromising paths from the early stages of the tree, referred to as the probabilistic tree pruning [16]. Two key ingredients, radius tightening and probabilistic tree pruning, work in a complementary fashion; While the tightened hypersphere radius is effective mainly in the bottom layers of the search tree, probabilistic tree pruning is effective in the top layers of the search tree. From the simulation of IDD scheme on 4×4 MIMO channel, we show that the proposed algorithm, henceforth referred to as *LSD with a probabilistic radius tightening* (PRT-LSD), achieves substantial complexity reduction ($47\% \sim 77\%$) over the conventional IDD scheme [2] with negligible performance loss (within 0.2 dB).

The rest of this paper is organized as follows. Section II provides a brief summary of the IDD system and LSD.

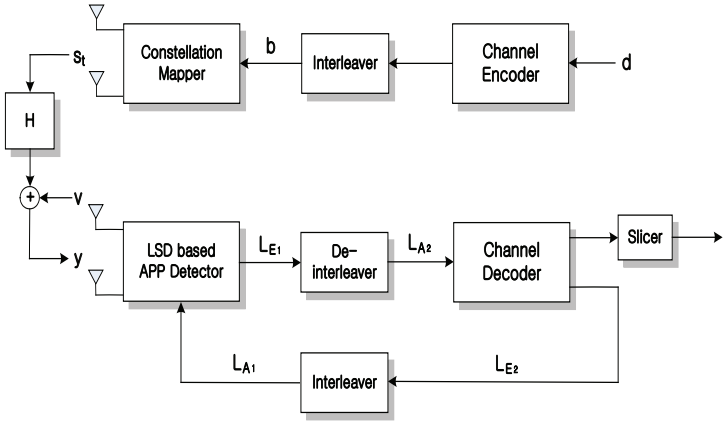


Fig. 1. The basic structure of IDD system.

In Section III, we present details of the proposed PRT-LSD algorithm including optimal sphere radius analysis and the list search with the probabilistic tree pruning. In Section IV, we provide simulation results including bit error rate (BER) performance, computational complexity, and extrinsic information transfer (EXIT) chart analysis, and we conclude our paper in Section V.

II. Iterative Detection and Decoding (IDD)

A. IDD System

Fig. 1 describes the basic structure of the IDD system, where $\mathbf{d}$ and $\mathbf{b}$ denote the information bit vector and the encoded and interleaved bit vector, respectively, and $\mathbf{s}_t \in \Lambda^\ell$ represents the transmitted symbol vector whose elements are chosen from the modulation set Λ with cardinality 2^q (i.e., $|\Lambda| = 2^q$). Since there exists one-to-one mapping between $\mathbf{b} = [\,b_1, \cdots, b_{q\ell}\,]^T$ and the symbol vector $\mathbf{s}_t = [\,s_1, \cdots, s_\ell\,]^T$, the input-output relationship of the system can be described as

$$\mathbf{y} \;=\; \mathbf{H}\mathbf{s}_t(\mathbf{b}) + \mathbf{v}. \tag{5}$$

In Fig. 1, the subscript "1" denotes the information associated with the APP detector, while the subscript "2" denotes the information associated with the channel decoder. After generating the extrinsic information $L_{E_1}(b_k)$, the APP detector passes the de-interleaved version of this soft information $L_{A_2}(b_k)$ into the channel decoder. Using this soft information as an input, channel decoder calculates its own extrinsic information $L_{E_2}(b_k)$. Lastly, interleaved version of $L_{E_2}(b_k)$, which serves as $L_{A_1}(b_k)$ in the APP detector, is fed back to the APP detector, thereby finishing an iteration. This job is repeated until suitably chosen performance criterion is satisfied or maximal number of iteration is reached [2].

B. List Sphere Decoding

Under the assumption that $\mathbf{H}$ is given and the QR-decomposition (QRD) of the channel matrix $\mathbf{H} = [\mathbf{Q}\ \mathbf{U}]\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}$ is applied, the ML solution becomes

$$\mathbf{s}_{ml} \;=\; \arg\min_{\mathbf{s}\in\Lambda^\ell} ||\,\mathbf{y} - \mathbf{H}\mathbf{s}\,||^2 \;=\; \arg\min_{\mathbf{s}\in\Lambda^\ell} \|\,\mathbf{y}' - \mathbf{R}\mathbf{s}\,\|^2 \tag{6}$$

where $\mathbf{y}' = \mathbf{Q}^T\mathbf{y}$. Further, by denoting a branch metric at layer $\ell - j + 1$ as $B_j(s_j^\ell) = |\, y'_k - \sum_{t=j}^{\ell} r_{k,t} s_t \,|^2$, we have

$$\mathbf{s}_{ml} \;=\; \arg\min_{\mathbf{s}\in\Lambda^\ell} \sum_{j=1}^{\ell} B_j(s_j^\ell). \tag{7}$$

The SD algorithm can be interpreted as a branch and bound (BB) based tree search algorithm [17]. In the first layer, i.e., the bottom row of $\mathbf{s}$, candidates satisfying $B_\ell(s_\ell) \le R_0^2$ are found (**bounding**). Once this step is finished, branching to the next layer of the best candidate (**branching**), $s_{\ell-1}$ satisfying $B_{\ell-1}(s_{\ell-1}^\ell) + B_\ell(s_\ell) \le R_0^2$ is searched. By repeating this step and updating the radius whenever a new lattice point $\mathbf{Rs}$ is found, the SD algorithm outputs the ML solution $\mathbf{s}_{ml}$ for which the cost function $\|\,\mathbf{y}' - \mathbf{Rs}\,\|^2 = \sum_{j=1}^{\ell} B_j(s_j^\ell)$ is minimized. In short, the salient features of the SD algorithm are that 1) the lattice point search is limited within the hypersphere of the radius R_0 and 2) R_0 is updated immediately after the lattice point satisfying the sphere constraint ($||\,\mathbf{y}' - \mathbf{Rs}\,||^2 \le R_0^2$) is found. It is worth mentioning that while the first feature of the SD is still valid for the LSD, second one is not directly satisfied due to the fact that N best lattice points should be found in the LSD. As mentioned, the sphere radius of the LSD is updated only when the list is full and a new lattice point replacing one in the list is searched. Hence, for the nontrivial list size N ensuring near-optimal performance, the required number of computations becomes far larger than that of the SD algorithm.

C. Radius Selection for LSD

In order to compute the extrinsic LLR in (4), the hypersphere with the radius R_0 should contain at least N lattice points. A simple choice of R_0 is to use a multiple of the cost function of $\mathbf{s}_t$ where $\mathbf{s}_t$ is the transmitted symbol vector. Since $\|\mathbf{y} - \mathbf{Hs}_t\|^2 = \|\mathbf{v}\|^2$ where $\|\mathbf{v}\|^2 \sim \sigma^2\chi_\ell^2$ is a χ^2-distribution with ℓ degrees of freedom (DOF), $E[\|\mathbf{y} - \mathbf{Hs}_t\|^2] = \ell\sigma^2$. Now by multiplying a fuzzy factor K, one can obtain

$$R_0^2 = \ell\sigma^2 K \tag{8}$$

for searching N lattice points [2]. Unfortunately, computational complexity of this choice, when combined with the radius update mechanism of the LSD, will be overwhelming. Fig. 2 illustrates the LSD operation of the BPSK modulation when $R_0^2 = 20$ is chosen using $K = 5$. Since the initial radius is much larger than the cost function of the fourth best lattice point with cost 11, we observe that too many radius updates occur, which clearly gives a negative impact on the search complexity.

III. LSD with a Probabilistic Radius Tightening

As discussed, the efficiency of the LSD is tightly coupled with the choice of the initial radius R_0. If R_0 is chosen to be smaller than the cost function of the N-th best candidate $J(\mathbf{s}_N) = \|\mathbf{y} - \mathbf{Hs}_N\|^2$, the search operation will be finished without filling the list, deteriorating the accuracy of the approximation in (4). Whereas, if R_0 is chosen to be much larger than $J(\mathbf{s}_N)$, it requires too much search effort [18]. Since arbitrarily chosen N does not shed light on the optimal performance/complexity tradeoff, we can easily infer that searching the fixed number of candidates cannot be an efficient way in general. For example, if there is a huge difference between the likelihoods of $\mathbf{s}_{ml}$ and $\mathbf{s}_i$, i.e., $P(\mathbf{y}|\mathbf{s}_{ml}) \gg P(\mathbf{y}|\mathbf{s}_i)$, then the contribution of $P(\mathbf{y}|\mathbf{s}_i)$ on the LLR is negligible while requiring substantial search effort. In order to overcome this drawback, we devise a condition to skip lattice points hardly affecting the LLR and then derive an optimal sphere radius R_0 conforming this condition.

A. Probabilistic Likelihood Constraint

Suppose the list $\mathcal{L}$ contains N lattice points after the search, i.e., $\mathcal{L} = \{\mathbf{s}_1 = \mathbf{s}_{ml},\ \mathbf{s}_2,\ \cdots,\ \mathbf{s}_N\}$. Then we want all lattice points in $\mathcal{L}$ to satisfy

$$\rho P(\mathbf{y}\,|\,\mathbf{s}_{ml}) \le P(\mathbf{y}\,|\,\mathbf{s}_i) \tag{9}$$

for some $\rho \in (0, 1)$. Since the reliability of the LLR improves as the list size increases, relatively small ρ is desired for the performance viewpoint. On the other hand, as the small list size is preferred for implementation perspective, relatively large ρ is recommended for pursuing the search efficiency.

In order to search for the symbols satisfying (9), it is necessary to set an initial radius defined by ρ. Recalling that $P(\mathbf{y}\,|\,\mathbf{s}_i) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{J(\mathbf{s}_i)}{2\sigma^2})$, (9) can be rewritten as

$$J(\mathbf{s}_i) \le J(\mathbf{s}_{ml}) - 2\sigma^2\ln\rho,\ \ i = 1, 2, \cdots, N. \tag{10}$$

Since all lattice points in $\mathcal{L}$ need to satisfy (10), the desired initial radius R_0 satisfies

$$R_0 = \sqrt{J(\mathbf{s}_{ml}) - 2\sigma^2\ln\rho}. \tag{11}$$

Note that two parameters required for R_0 computation are $J(\mathbf{s}_{ml})$ and ρ. While $J(\mathbf{s}_{ml})$ is computed from the SD algorithm directly, bound of ρ needs to be obtained analytically. In particular, in order to minimize the complexity as well as the performance loss, it is desirable to obtain an upper bound of ρ.

B. Upper Bound of ρ

Our goal in this subsection is to obtain an upper bound of ρ used in the computation of R_0. First, it is clear from (9) that

$$N\rho g(\omega_{ml}) \le \sum_{i=1}^{N} g(\omega_i). \tag{12}$$

where $\omega_i = \frac{J(\mathbf{s}_i)}{\sigma^2} = \frac{\|\mathbf{y}-\mathbf{Hs}_i\|^2}{\sigma^2}$ and $g(x) = \exp\left(-\frac{x}{2}\right)$ and thus $P(\mathbf{y}|\mathbf{s}_i) = \frac{1}{\sqrt{2\pi\sigma^2}}g(\omega_i)$. From (12), we have

$$N\rho g\left(\omega_{ml}\right) \le \sum_{i=1}^{N} E\left[g\left(\omega_i\right)\right] \tag{13}$$

where $E[g(\omega_i)]$ is given by

$$E[g(\omega_i)] \;=\; \int_{\omega_i} g(\omega_i) f_G(g(\omega_i)) d\omega_i \tag{14}$$

$$\;=\; \int_{\omega_i} g(\omega_i) f_{\Omega_i}(\omega_i) d\omega_i \tag{15}$$

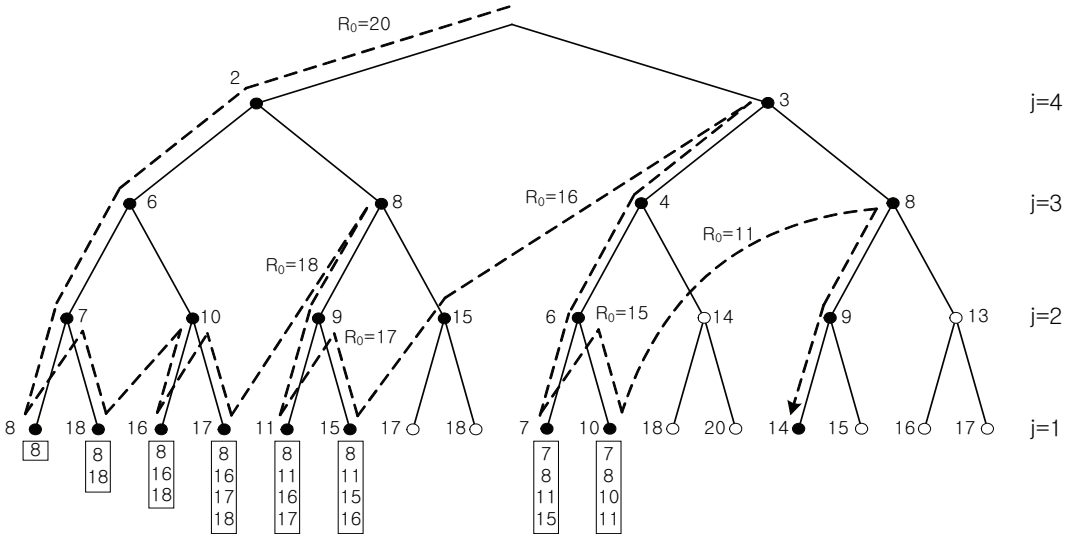


Fig. 2. Illustration of the conventional LSD search in a tree structure with $N = 4$ and BPSK modulation. Numbers inside the bottom rectangle denote the cost functions associated with lattice points in $\mathcal{L}$. Note that white nodes are skipped since they violate the hypersphere condition.

where $f_G(\cdot)$ and $f_{\Omega_i}(\cdot)$ are the probability density functions (PDF) of $g(\cdot)$ and ω_i, respectively, and (15) is because $g(\cdot)$ is one-to-one function. From (13) and (15), the upper bound of ρ becomes

$$\rho \leq \frac{1}{Ng\left(\omega_{ml}\right)} \sum_{i=1}^{N} \int_{\omega_i} g(\omega_i) f_{\Omega_i}(\omega_i) d\omega_i. \tag{16}$$

Now, what remains is the evaluation of the integral in (16). First, by denoting $\mu_i = \mathbf{H}(\mathbf{s}_t - \mathbf{s}_i)$, ω_i is expressed as

$$\begin{aligned} \omega_i &= \frac{\|\mathbf{y} - \mathbf{H}\mathbf{s}_i\|^2}{\sigma^2} = \frac{\|\mathbf{H}(\mathbf{s}_t - \mathbf{s}_i) + \mathbf{v}\|^2}{\sigma^2} \\ &= \frac{\|\mu_i + \mathbf{v}\|^2}{\sigma^2}. \end{aligned} \tag{17}$$

Then ω_i is a noncentral χ^2-distribution with PDF

$$\begin{aligned} &f_{\Omega_i}(\omega_i\ ; \nu, \lambda_i) = \\ &\frac{1}{2} \exp\left(-\frac{\omega_i + \lambda_i}{2}\right) \left(\frac{\omega_i}{\lambda_i}\right)^{\frac{\nu-2}{4}} I_{\frac{\nu}{2}-1}\left(\sqrt{\lambda_i \omega_i}\right), \end{aligned} \tag{18}$$

where $\nu = \ell$ is the DOF, $\lambda_i = \left\|\frac{\mu_i}{\sigma}\right\|^2$ is the noncentrality parameter and $I_a(\cdot)$ is a modified Bessel function of the first kind [19]. Using the PDF of the noncentral χ^2-distribution, the integration in the right-hand side of (16) becomes

$$\begin{aligned} \int_{\omega_i} g(\omega_i) f_{\Omega_i}(\omega_i) d\omega_i = \int_0^{\infty} \frac{g(\omega_i)}{2} \exp\left(-\frac{\omega_i + \lambda_i}{2}\right) \cdot \\ \left(\frac{\omega_i}{\lambda_i}\right)^{\frac{\nu-2}{4}} I_{\frac{\nu}{2}-1}\left(\sqrt{\lambda_i \omega_i}\right) d\omega_i. \end{aligned} \tag{19}$$

Unfortunately, it is not easy to get a closed form solution of this integral. In this work, we use Patnaik's approximation for the noncentral χ^2-distribution [20] to evaluate (19). Patnaik's method is a simple yet an accurate way of approximating noncentral χ^2-distribution $f_{\Omega_i}(\omega_i\ ; \nu, \lambda_i)$ with ν and λ_i into the product of a constant $c_i = \frac{(\nu+2\lambda_i)}{(\nu+\lambda_i)}$ and the central χ^2-distribution $f_{X_i}(x_i; k_i)$ given by

$$c_i f_{X_i}(x_i; k_i) = \frac{c_i}{2^{k_i/2}\Gamma(\frac{k_i}{2})} x_i^{\frac{k_i}{2}-1} \exp\left(-\frac{x_i}{2}\right) \tag{20}$$

where $k_i = \frac{(\nu+\lambda_i)^2}{(\nu+2\lambda_i)}$ is the DOF and $\Gamma(\cdot)$ is the Gamma function [19]. Using Patnaik's approximation, (19) becomes

$$\begin{aligned} \int_{\omega_i} g(\omega_i) f_{\Omega_i}(\omega_i) d\omega_i &\approx \int_0^{\infty} g(x_i) c_i f_{X_i}(x_i; k_i) dx_i \\ &= \frac{c_i}{2^{k_i/2}\ \Gamma\left(\frac{k_i}{2}\right)} \int_0^{\infty} x_i^{\frac{k_i}{2}-1} e^{-x_i} dx_i \\ &= \frac{c_i}{2^{k_i/2}} \end{aligned} \tag{21}$$

where (21) is from the definition of the gamma function $\Gamma(k) = \int_0^{\infty} x^{k-1} e^{-x} dx$. Combining (16) and (21), one can obtain the upper bound of ρ as

$$\begin{aligned} \rho &\lessapprox \frac{1}{Ng\left(\omega_{ml}\right)} \sum_{i=1}^{N} \frac{c_i}{2^{k_i/2}} \\ &= \frac{1}{Ng(\omega_{ml})} \sum_{i=1}^{N} \epsilon_i \\ &= \varrho \exp\left(\frac{J(\mathbf{s}_{ml})}{2\sigma^2}\right) \end{aligned} \tag{22}$$

where $\epsilon_i = \frac{c_i}{2^{k_i/2}}$ and $\varrho = \frac{1}{N} \sum_{i=1}^{N} \epsilon_i$.

C. Radius Selection for PRT-LSD

By combining the desired initial radius R_0 and the upper bound of ρ in (22), we have

$$\begin{aligned} R_0 &= \sqrt{J(\mathbf{s}_{ml}) - 2\sigma^2 \ln \rho} \\ &\gtrapprox \sigma \sqrt{\ln\left(\frac{1}{\varrho^2}\right)}. \end{aligned} \tag{23}$$

Since any hypersphere radius satisfying this bound satisfies the condition in (10) as well, we choose the lower bound (minimum value) of R_0 in (23) as the radius for the list sphere search. It is worth making some qualitative remarks on the radius we obtained in (23).

- List sphere search can be initiated immediately after obtaining the ML solution $\mathbf{s}_{ml}$. Regarding $\mu_i = \mathbf{H}(\mathbf{s}_t - \mathbf{s}_i)$ (used for the computation of c_i and k_i), since $\mathbf{s}_t$ is unknown, μ_i cannot be directly computed and hence we need to use an approximation $\mu_i \approx \mathbf{H}(\mathbf{s}_{ml} - \mathbf{s}_i)$.
- We can deduce from (23) that the range of R_0^2 is $[\sigma^2 \ln(1/\varrho^2) \;\; \infty)$ which is automatically adjusted to the system setup and the channel condition.
- R_0 of the PRT-LSD is unrelated to the list size N and thus it is possible that the search is finished without filling the list.

D. LSD with Probabilistic Tree Pruning

The LSD should reject symbol vectors violating the hypersphere condition given by

$$\| \mathbf{y}' - \mathbf{R}\mathbf{s} \|^2 = \sum_{j=1}^{\ell} B_j(s_j^\ell) \le R_0^2. \tag{24}$$

In practice, only contributions of already visited layers can be counted and hence the actually used condition in the $\ell - j + 1$-th search layer becomes

$$P_j^\ell(s_j^\ell) = B_j(s_j^\ell) + \cdots + B_\ell(s_\ell^\ell) \le R_0^2 \tag{25}$$

where $P_j^\ell(s_j^\ell)$ is the current path metric. Since the partial path metric is compared with the hypersphere radius, tree pruning is not quite effective in the top layers of the search tree. In order to compensate the unvisited node contribution and thereby improve the pruning capability of the sphere search, probabilistic tree pruning was introduced [16], [21]. The key idea behind the probabilistic tree pruning is to estimate the noncausal path metric under a benign scenario where the rest of the search is perfect ($s_i = s_{t,i}$, $i = m, \cdots, \ell$). In this scenario, the branch metric $B_i, i \in \{1 \cdots, j-1\}$ is modeled as the square of the Gaussian noise as

$$B_m(s_m^\ell) = (\, y_m - \sum_{i=m}^{\ell} r_{m,i}\, s_i \,)^2 = v_m^2, \;\; m = 1, \cdots, j-1 \tag{26}$$

where v_m is the m-th component of $\mathbf{v}$. With this setup, the modified sphere condition becomes

$$\sum_{m=1}^{\ell} B_t(s_m^\ell) = P_j^\ell(s_j^\ell) + \sum_{m=1}^{j-1} v_m^2 \le R_0. \tag{27}$$

Since $v_1, \cdots, v_{j-1}$ are values from i.i.d. Gaussian distribution, $\sum_{m=1}^{j-1} v_m^2$ becomes the χ^2-random variable with $j-1$ DOF. Denoting $\Phi_{j-1} = \sum_{m=1}^{j-1} v_m^2$, (27) can be rewritten as

$$P_j^\ell(s_j^\ell) + \Phi_{j-1} \le R_0. \tag{28}$$

Note that we cannot directly use (28) for the sphere search since Φ_{j-1} is a random variable. As a way to get around this problem, we examine the probability that the rest of the tree is detected perfectly so that the remaining portion is a pure noise contribution. If the probability of this event (i.e., $P(\Phi_{j-1} + P_j^\ell(s_j^\ell) \le R_0)$) is smaller than a pre-defined threshold P_ϵ so called a pruning probability, we treat this event as unpromising one and prune all subtrees starting from this node. This condition can be expressed as [16]

$$P(\Phi_{j-1} \le R_0 - P_j^\ell(s_j^\ell)) \;\le\; P_\epsilon. \tag{29}$$

When applying the probabilistic tree pruning into the LSD, however, the branch metric in (26) becomes overly conservative choice since the LSD searches for N-best candidates ($\mathbf{s}_1 = \mathbf{s}_{ml}, \mathbf{s}_2, \cdots, \mathbf{s}_N$). Therefore, instead of using perfect symbol detection assumption, we employ noise plus detection error model as

$$\begin{aligned} B_m(s_m^\ell) &= \left| y_m - \sum_{i=m}^{\ell} r_{m,i}\, s_i \right|^2 \\ &= \left| \sum_{i=m}^{\ell} r_{m,i}(s_{t,i} - s_i) + v_m \right|^2 \\ &= \left| \sum_{i=m}^{\ell} r_{m,i} e_i + v_m \right|^2 \\ &\approx |\delta_m + v_m|^2 \end{aligned} \tag{30}$$

for $m = 1, \cdots, j-1$ where $s_{t,i}$ is the i-th component of $\mathbf{s}_t$ and $\delta = \mathbf{R}(\mathbf{s}_{ml} - \mathbf{s})$ where (30) is from $\mathbf{R}(\mathbf{s}_t - \mathbf{s}) \approx \mathbf{R}(\mathbf{s}_{ml} - \mathbf{s})$. Using (30), modified sphere condition of the LSD becomes

$$\sum_{m=1}^{\ell} B_m(s_m^\ell) = P_j^\ell(s_j^\ell) + \sum_{m=1}^{j-1} |\delta_m + v_m|^2 \le R_0^2. \tag{31}$$

Let $\Phi'_{j-1} = \sum_{m=1}^{j-1} \left(\frac{\delta_m + v_m}{\sigma}\right)^2$ then Φ'_{j-1} becomes the noncentral χ^2-random variable with $j-1$ DOF and (31) can be rewritten as $\frac{P_j^\ell(s_j^\ell)}{\sigma^2} + \Phi'_{j-1} \le \frac{R_0^2}{\sigma^2}$. Similar to (29), the pruning condition can be expressed as

$$P\left(\Phi'_{j-1} \le \frac{R_0^2 - P_j^\ell(s_j^\ell)}{\sigma^2}\right) \;\le\; P_\epsilon. \tag{32}$$

Since the left-hand side of (32) is the cumulative density function (CDF) of noncentral χ^2-random variable with DOF $j-1$, we have

$$R_0^2 - P_j^\ell(s_j^\ell) \;\le\; \beta_{j-1} = \sigma^2 F_\Phi^{-1}(P_\epsilon\,;\, j-1, \lambda) \tag{33}$$

where $F_\Phi^{-1}(P_\epsilon\,;\, j-1, \lambda)$ is the inverse CDF of noncentral χ^2-distribution and $\lambda = \sum_{i=1}^{j-1} \left(\frac{\delta_i}{\sigma}\right)^2$. Similar to (20), using Patnaik's approximation [20], the noncentral χ^2-distribution is approximated as

$$\eta f_X(x;\kappa) = \frac{\eta}{2^{\kappa/2}\Gamma(\frac{\kappa}{2})} x^{\frac{\kappa}{2}-1} \exp\left(-\frac{x}{2}\right) \tag{34}$$

where $\eta = \frac{(j-1+2\lambda)}{(j-1+\lambda)}$, $\kappa = \frac{(j-1+\lambda)^2}{(j-1+2\lambda)}$ and $f_X(x;\kappa)$ is the χ^2-distribution with DOF κ and thus the pruning condition in (33) becomes

$$R_0^2 - P_j^\ell(s_j^\ell) \;<\; \sigma^2 F_X^{-1}(P_\epsilon \,/\, \eta\,;\, \kappa) = \beta'_{j-1}. \tag{35}$$

Equivalently,

$$R_0^2 - \beta'_{j-1} \;<\; P_j^{\ell}(s_j^{\ell}). \tag{36}$$

The pruning condition in (36) can be interpreted as follows; if the path metric in layer $\ell - j + 1$ is larger than $R_0^2 - \beta'_{j-1}$, then the rest of search is unlikely to satisfy the hypersphere condition even for an ideal scenario where the contribution of remaining nodes is noise only. Therefore, whenever a path s_j^{ℓ} satisfies this pruning condition, we immediately remove the subtree of the node and hence can avoid a waste of computations.

The illustration of a binary tree search for the proposed PRT-LSD algorithm is shown in Fig. 3. With $\varrho = 0.004$, one can get $R_0^2 = 11$. Although cost functions of the final lattice points are 7, 8, 10 and 11 (same as the example in the Fig. 2), we can observe that lots of unpromising nodes are skipped due to the use of tight radius. As will be shown in the simulation, the benefit of the probabilistic radius tightening is pronounced when the system parameters become large.

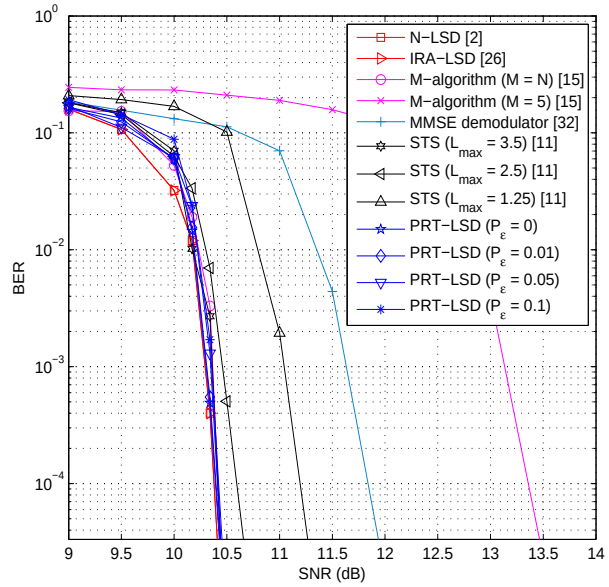
IV. Simulation and Discussions

A. Simulation Setup

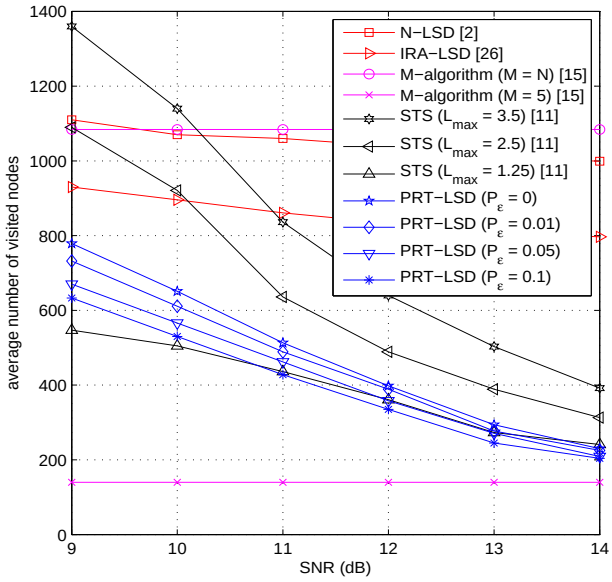
In this section, we observe the performance and complexity of the proposed PRT-LSD algorithm along with previously proposed APP detection approaches. We also provide an extrinsic information transfer (EXIT) chart analysis [22] to validate near-optimality of the proposed method.

The simulation setup is based on the 16-QAM transmission with gray mapping. The signal is transmitted over 4×4 MIMO system in Rayleigh fading channel where elements of the channel matrix $\mathbf{H}$ are modeled by independent Gaussian random variable. The half rate turbo coding with feedback polynomial $1+D+D^2$ and feedforward polynomial $1+D^2$ is used to encode a binary sequence $\mathbf{d}$, and a random interleaver is employed to generate the interleaved bit sequence $\mathbf{b}$. For each code block, four IDD operations are performed with eight iterations within the Turbo decoder. As a metric for measuring performance and complexity, the bit error rate (BER) and the average number of visited nodes are employed. In addition, at least $50,000$ channel realizations are tested for each SNR point and channel is assumed to be known at the receiver. For comprehensive view, we perform simulations on the following algorithms:

1) N-LSD [2] : set the initial radius with $K = 5$ in (8). R_0 is updated with the maximal cost function in $\mathcal{L}$ only when the list is filled. In case the search is done without filling the list, it is repeated with an increased radius ($K := K + 1$).
2) modified N-LSD using an increasing radii algorithm (IRA-LSD) : set the initial radius R_0 based on ϵ and δ [23]. When the search is finished without filling the list, R_0 is re-computed with smaller ϵ and then the search is repeated.
3) M-algorithm [12] : choosing M best paths for each layer (use $M = N$ and $M = 5$).
4) Single tree search (STS) [9] : set initial radius $R_0 = \infty$ and update R_0 to the cost function of Babai point. $\mathbf{s}_{ml}$ as well as its binary complement is found concurrently. Also, LLR-correction technique is applied [28].
5) MMSE-based demodulator [29] : perform MMSE detection followed by per-layer LLR computation.
6) PRT-LSD : set R_0 based on (23) with four distinct pruning probabilities $P_\epsilon = \{0,\, 0.01,\, 0.05,\, 0.1\}$.



(a) BER performance



(b) Average number of visited nodes

Fig. 4. Performance and complexity of 4×4 MIMO system with 16-QAM modulation ($N = 50$).

B. Performance and Complexity

Fig. 4(a) shows the BER performance of the APP detection schemes for 16-QAM transmission. In this simulation,

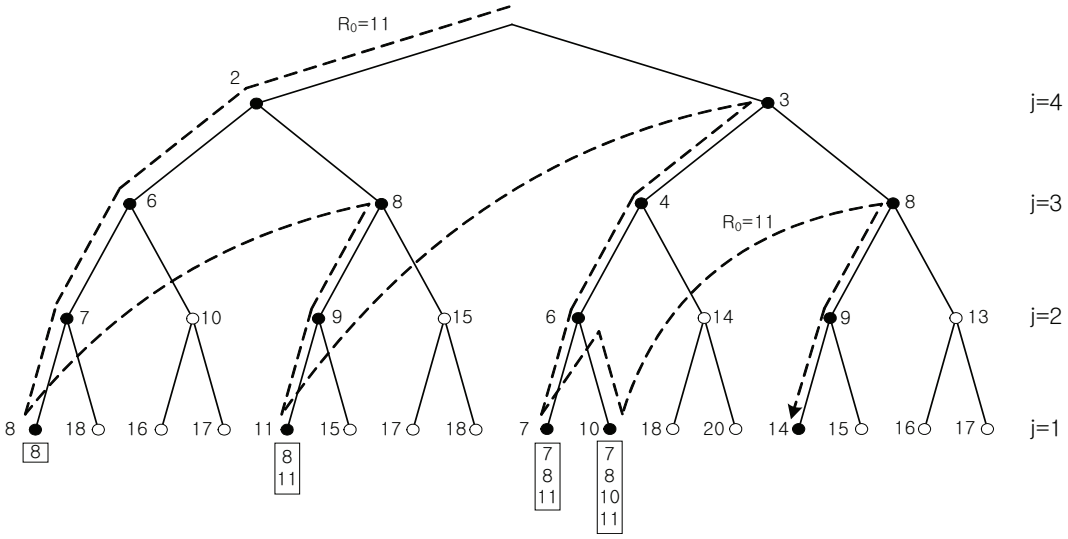


Fig. 3. Illustration of the proposed PRT-LSD search in a tree structure with $N = 4$ and BPSK modulation. Numbers inside the bottom rectangle denote the cost functions associated with lattice points in $\mathcal{L}$.

$N = 50$ is used for the N-LSD algorithm and its variations. Although there is a slight difference in performance between the conventional LSD (N-LSD and IRA-LSD) and the proposed PRT-LSD in low SNR regime (BER $> 10^{-2}$), the difference becomes negligible in the waterfall regime. Interestingly, the performance curves of the LSD and PRT-LSD for $P_\epsilon = 0.01, 0.05$ lie on top of each other for most SNR regime under test so that there is no virtual loss in performance. Even with $P_\epsilon = 0.1$, the performance loss of the PRT-LSD over the conventional LSD is only about 0.1 dB.

Fig. 4(b) plots the average number of visited nodes per search. Except for M-algorithm having constant complexity, all techniques based on the modification of sphere search show a tendency that the complexity is getting smaller as the SNR increases. Due to the overhead to find N-best lattice points without proper control of the hypersphere radius, the N-LSD shows worst complexity among sphere search techniques. Since the IRA-LSD is being operated with relatively small initial radius, its complexity is smaller than the N-LSD, achieving moderate complexity reduction (e.g., 16% and 20% reduction at 9 dB and 14 dB of SNR respectively). Owing to the inclusion of the optimal initial radius together with the probabilistic tree pruning mechanism, the complexity reduction of the PRT-LSD is noticeably better so that it achieves 43% and 80% reduction over the N-LSD for $P_\epsilon = 0.1$.

In Fig. 5, we plot the performance of the N-LSD and the proposed PRT-LSD for $N = 10$, $N_{\mathbf{d}} = 2,000$ and maximum 2 iterations. For both algorithms, we see the reduction in complexity and degradation in performance, mainly due to the small list size and the code block length. We can, however, see that computational benefit of the PRT-LSD over the LSD is maintained.

Since sorting is heuristic yet effective way to improve the computational complexity of sphere search [27] and list sphere search [24], it is of interest to investigate the performance and complexity when the sorted QR decomposition is applied. Towards this end, we combine the permutation matrix $\mathbf{P}$ and the channel matrix $\mathbf{H}$ so that the main diagonal entries of $\mathbf{R}$ after the QR-decomposition ($\mathbf{HP} = \mathbf{QR}$) are sorted in an ascending order. Additionally, since poorly conditioned channel realizations might incur an increase in complexity due to the severe drop in the effective SNR, the regularization can be applied on top of the sorted QR-decomposition (readers are referred to [25] for more details) as

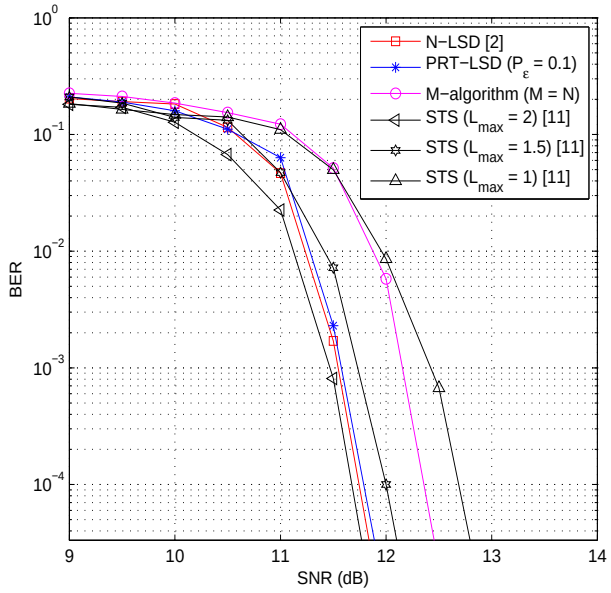
$$\begin{bmatrix} \mathbf{H} \\ \alpha \mathbf{I} \end{bmatrix} \mathbf{P} = \mathbf{QR}, \tag{37}$$

where α is a suitably chosen regularization parameter [25]. Fig. 6 compares the performance and the complexity of the algorithms with the conventional QR decomposition and those with the MMSE-SQRD for 16-QAM transmission. We see that by employing the MMSE-SQRD preprocessing, considerable reduction in complexity can be achieved for all algorithms. Although the complexity gain of the MMSE-SQRD is slightly favorable to the N-LSD, the overall gain of the PRT-LSD over the N-LSD remains almost unchanged. To be specific, complexity gain of the PRT-LSD over the N-LSD at 9 dB and 14 dB is 34% and 78%, respectively.

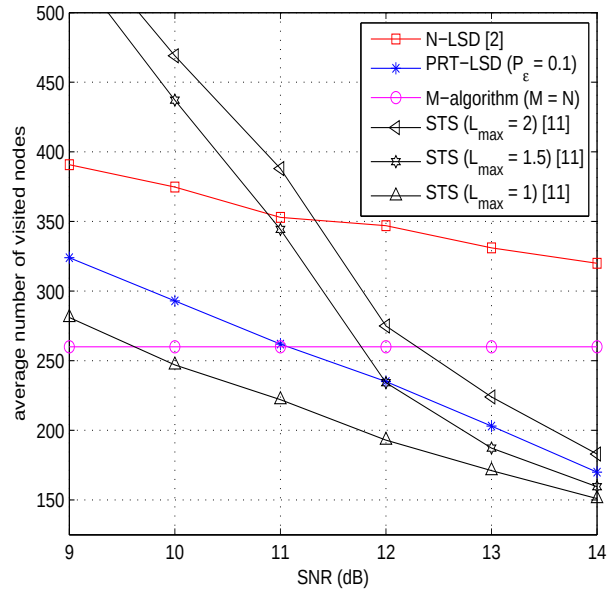
C. EXIT Chart Analysis

We finally check the near optimality of our method using an EXIT chart analysis. EXIT chart is a useful tool to analyze convergence behaviors of soft-input/soft-output (SISO) based iterative decoding schemes [22], [30]. By comparing the EXIT charts of the conventional LSD and proposed method, we can observe the similarity in convergence behavior.

The mutual information I_E between the information bit vector $\mathbf{d}$ and extrinsic output of the SISO operators (i.e., APP

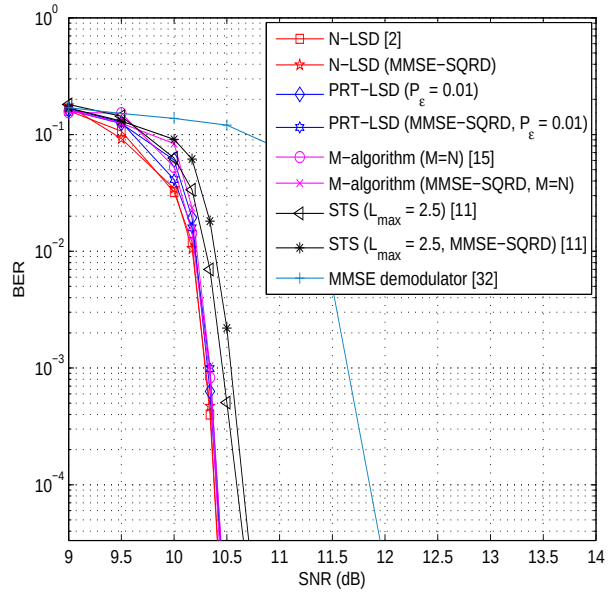


(a) BER Performance

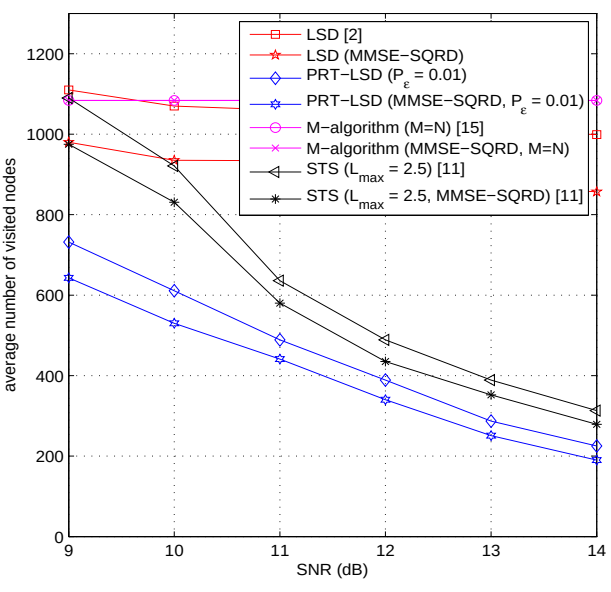


(b) Average number of visited nodes

Fig. 5. Performance and complexity of PRT-LSD when $N = 10$ using 16-QAM transmission.



(a) BER performance



(b) Average number of visited nodes

Fig. 6. Performance and complexity of 4×4 system with 16-QAM transmission with MMSE-SQRD employed ($N = 50$).

detector and Turbo decoder) is defined as [26], [30]

$$I_E = \frac{1}{2}\sum_{d=-1,1}\int_{-\infty}^{\infty} p_E\left(\xi|d\right)\cdot \log_2\frac{2p_E\left(\xi|d\right)}{p_E\left(\xi|d=-1\right)+p_E\left(\xi|d=1\right)}d\xi \qquad (38)$$

where $p_E\left(\xi|d\right) = \frac{\exp\left(-(\xi-(\sigma_E^2/2)\cdot d)^2/2\sigma_E^2\right)}{\sqrt{2\pi}\sigma_E}$ $(0 \le \sigma_E \le 1)$. Applying the symmetry condition $p_E\left(\xi|d\right) = p_E\left(-\xi|-d\right)$ [31] and consistency condition $p_E\left(\xi|d\right) = p_E\left(-\xi|d\right)\cdot\exp\left(d\xi\right)$ [30], (38) can be simplified to [32]

$$I_{E_{\theta,\tau}} \approx 1 - \frac{1}{N_{\mathbf{d}}}\sum_{i=1}^{N_{\mathbf{d}}}\log_2\left(1+\exp\left(-d_i\cdot L_{E_{\theta,\tau}}(d_i)\right)\right) \qquad (39)$$

where d_i is the i-th bit of the information bit vector $\mathbf{d}$, $L_{E_{\theta,\tau}}$ is the extrinsic LLR value (τ denotes the number of iterations and $\theta = 1$ and 2 denote the information associated with APP detector and Turbo decoder, respectively). By computing (39) for each iteration and generating two-dimensional plot, we

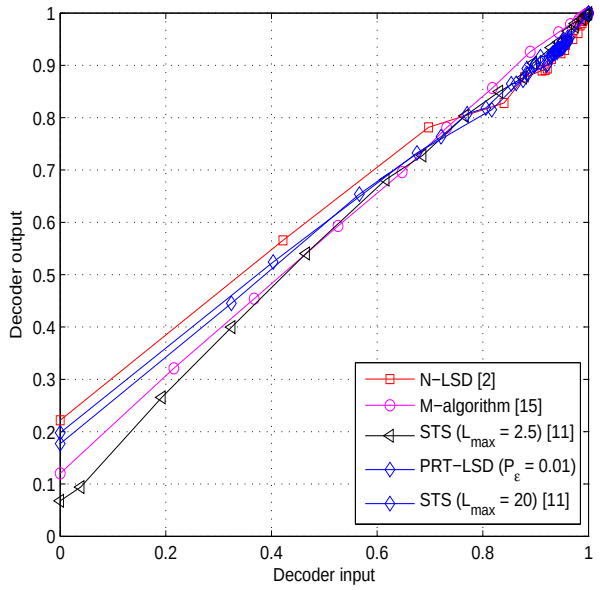


Fig. 7. The decoder trajectory for 16-QAM transmission, $N = 50$, SNR = 11.5 (dB).

obtain the EXIT chart.

In our simulation, we plot the EXIT charts of algorithms under test for 16-QAM transmission in the waterfall regime (SNR = 11.5 dB). Fig. 7 shows the trajectory of $I_{E_{\theta,\tau}}$ for the inner iteration of the Turbo decoder with 16-QAM transmission. Although there is a slight gap in trajectory between the N-LSD and PRT-LSD for the 1st and 2nd iteration, the trajectories become almost identical after the 3rd iteration. Similar behavior can be observed for the STS and M-algorithm as well.

V. Conclusions

In this paper, we proposed a low-complexity list sphere decoding referred to as the LSD with a probabilistic radius tightening (PRT-LSD). Key features of the PRT-LSD are 1) the selection of the hypersphere radius R_0 for pruning lattice points with vanishing likelihood and 2) the LSD with the probabilistic tree pruning. Our derivation of the hypersphere radius is based on the observation that contribution of the lattice points with small likelihood on the extrinsic information is negligible. We observed from simulations of the MIMO channel that the tightened sphere radius, in conjunction with the probabilistic tree pruning, is very effective in reducing the computational complexity of IDD. Further, we showed from the EXIT chart analysis that the characteristics of the conventional N-LSD and proposed method are quite similar, in particular for two or more iterations. Although the validity of the proposed method was primarily investigated in the LSD, we expect that the effectiveness of the proposed method will be maintained for other depth-first tree search algorithms. We also expect that complexity gain of the proposed method will be more significant for future wireless systems with large transmit/receive antennas.

References

[1] J. Lee, B. Shim, and I. Kang "List Sphere Decoding with a Probabilistic Radius Tightening," *IEEE GLOBECOM*, Dec. 2010.

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

[3] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429-445, March 1996.

[4] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Computat.*, vol. 44, pp. 463-471, April 1985.

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

[6] 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.

[7] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 2299-2311, Nov. 2004.

[8] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *IEEE Trans. Commun.*, vol. 54, pp. 587-590, April 2006.

[9] C. Studer, A. Burg, and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. on Inform. Theory*, vol. 56, pp. 4827-4842, Oct. 2010. (see also http://www.ece.rice.edu/∼cs32/index.html)

[10] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Commun.*, vol. 24, pp. 491-503, March 2006.

[11] J. B. Anderson and S. Mohan, "Sequential coding algorithms: a survey and cost analysis," *IEEE Trans. Commun.*, vol. COM-32, pp. 169-176, Feb. 1984.

[12] Yvo L. C. de Jong and T. J. Willink, "Iterative tree search detection for MIMO wireless systems," *IEEE Trans. Commun.*, vol. 53, pp. 930-935, June 2005.

[13] L. G. Overlinebero, T. Ratnarajah, and C. Cowan, "A low-complexity soft-MIMO detector based on the fixed-complexity sphere decoder," *IEEE Acoustic, Speech and Signal Process. (ICASSP)*, pp. 2669-2672, April 2008.

[14] E. G. Larsson and J. Jaldén, "Fixed-complexity soft MIMO detection via partial marginalization," *IEEE Trans. Signal Process.*, vol. 56, pp. 3397-3407, Aug. 2008.

[15] J. Boutros, N. Gresset, L. Brunel and M. Fossorier, "Soft-input soft-output lattice sphere decoder for linear channels," *IEEE GLOBECOM*, Dec. 2003.

[16] B. Shim and I. Kang, "Sphere decoding with a probabilistic tree pruning," *IEEE Trans. Signal Process.*, vol. 56, pp. 4867-4878, Oct. 2008.

[17] J. Clausen, "Branch and Bound Algorithms. Principles and Examples," Dept. Comput. Sci., Univ. Copenhagen, [Online]. Available:http://www.imm.dtu.dk/jha/.

[18] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. expected complexity," *IEEE Trans. Signal Process.*, vol. 53, pp. 2806-2818, Aug. 2005.

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

[20] P. B. Patnaik, "The noncentral χ^2 and F-distributions and their applications," *Biometrika*, vol. 36, pp. 202-232, 1949.

[21] B. Shim and I. Kang, "On further reduction of complexity in tree pruning based sphere search," *IEEE Trans. Commun.*, vol. 58, pp. 417-422, Feb. 2010.

[22] S. ten Brink, "Convergence of iterative decoding," *Electron. Lett.*, vol. 35, pp. 806-808, May 1999.

[23] R. Gowaikar and B. Hassibi, "Statistical pruning for near-maximum likelihood decoding," *IEEE Trans. Signal Process.*, vol. 55, pp. 2661-2675, June 2007.

[24] D. Wübben, R. Böhnke, J. Rinnas, V. Kühn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electron. Lett.*, vol. 37, pp. 1348-1350, Oct. 2001.

[25] D. Wübben, R. Böhnke, J. Rinnas, V. Kühn, and K. D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," *Proc. IEEE Vehic. Tech. Conf.*, vol. 1, pp. 508-512, Oct. 2003.

[26] T. M. Cover and J. A. Thomas, *Elements of Inform. Theory*, New York: Wiley, 1991.

[27] 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.

[28] S. Schwandter, P. Fertl, C. Novak, and G. Matz, "Log-likelihood ratio clipping in MIMO-BICM systems: information geometric analysis and impact on system capacity," *Proc. ICASSP*, Apr. 2003

[29] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using parallel interference cancellation," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 1754-1765, July 2011.

[30] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727-1737, Oct. 2001.

[31] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, Feb. 2001.

[32] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Trans. Commun.*, vol. 52, pp. 209-218, Feb. 2004.