Reduced Complexity Closest Point Decoding Algorithms for Random Lattices*

Wanlun Zhao and *Georgios B. Giannakis*† (`contact author`)

Abstract

Closest point algorithms find wide applications in decoding block transmissions encountered with single- or multi-user communication links relying on a single or multiple antennas. In this work, such algorithms are considered for random lattices generated by a known statistical model. The conventional sphere decoding algorithm (SDA) and related reduced-complexity techniques are examined from a probabilistic perspective. Combining SDA with optimal detection ordering is justified based on this random model. A novel algorithm that checks codeword candidates according to a probability order is derived and analyzed. Based on a probabilistic search and a suitable stopping criterion, a computationally efficient hybrid decoding algorithm is developed with near-SDA error performance. Simulations confirm that considerable complexity reduction is achieved by this hybrid algorithm as the signal to noise ratio increases.

Index Terms – Closest point algorithm, sphere decoding, space-time, multiuser detection, random lattice decoding, LLL lattice reduction.

Submission Date: August 16, 2003

Suggested Areas: Space-time processing
Multiuser detection
Detection and estimation

* The work in this paper was supported by the ARL/CTA grant no. DAAD19-01-2-0011

† The authors are with the Dept. of ECE, Univ. of Minnesota, 200 Union Str. SE, Minneapolis, MN 55455, USA; Tel/Fax: (612) 626-7781/625-4583; Emails: {`wlzhao, georgios`}`@ece.umn.edu`

I. INTRODUCTION

Consider the following generic model:

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

where $\mathbf{y}, \mathbf{v} \in \mathbb{C}^M$, $\mathbf{s} \in \mathbb{Z}^N[j]$, $\mathbf{H} \in \mathbb{C}^{M\times N}$ has full rank, $M \geq N$, $\mathbb{Z}$ and $\mathbb{C}$ denote the sets of integers and complex numbers, respectively; and $\mathbb{Z}[j] := \{a + jb \,|\, a, b \in \mathbb{Z}\}$ is the set of Gaussian integers. Operating on $\mathbf{s}$, the matrix $\mathbf{H}$ generates a lattice that we denote as: $\Lambda(\mathbf{H}) := \{\mathbf{x} = \mathbf{H}\,\mathbf{s} \,|\, \mathbf{s} \in \mathbb{Z}^N[j]\}$. The *closest point problem* is: Given $\mathbf{y} \in \mathbb{C}^M$ and a lattice Λ with known *generator* $\mathbf{H}$, find the lattice vector $\hat{\mathbf{x}} \in \Lambda$ that minimizes the Euclidean distance from $\mathbf{y}$ to $\hat{\mathbf{x}}$; that is, $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\Lambda} \|\mathbf{y} - \mathbf{x}\|^2$, where $\|\cdot\|$ represents the Frobenius norm.

In a wireless communication context, $\mathbf{s}$, $\mathbf{y}$, and $\mathbf{v}$ are the transmitted, received, and the additive white Gaussian noise (AWGN) vectors, whereas $\mathbf{H}$ contains the channel coefficients. The distribution of $\mathbf{v}$ is $\mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$, where $\mathcal{CN}(\cdot,\cdot)$ represents the complex Gaussian distribution; and $\mathbf{H}$ is a random matrix often with known statistical properties. Furthermore, instead of the whole integer lattice $\mathbb{Z}^N[j]$, $\mathbf{s}$ is usually drawn from a finite alphabet (subset) $\mathcal{S}^N \subset \mathbb{Z}^N[j]$. In block decoding, we are interested in determining the maximum likelihood (ML) estimate of $\mathbf{s}$, subject to finite-alphabet (FA) constraints:

$$\hat{\mathbf{s}}_{ML} = \arg\min_{\mathbf{s}\in\mathcal{S}^N} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2.$$

Under FA constraints, closest point algorithms can be employed to find $\hat{\mathbf{s}}_{ML}$ in various applications including space-time decoding and multiuser detection. The sphere decoding algorithm (SDA) was originally introduced to determine vectors with small norms in an arbitrary lattice [8], but has gained popularity in decoding lattice codes [17], CDMA [3], and space-time transmissions [19], [15]. Its popularity stems from the fact that SDA offers near-ML decoding at polynomial complexity – a major reduction of the exponential (in codeword length N and $|\mathcal{S}|$) complexity incurred by exhaustive search. A variate of SDA, first utilized by Schnorr and Euchner (SE), appeared recently in both [1] and [5], where an ordering mechanism was introduced to improve search efficiency. Under common random channel and noise models, the average complexity of the SDA was derived in [11] along with an effective method to determine the initial search radius.

In this work, we improve the computational efficiency of closest point algorithms by exploiting the underlying random models of both noise and channel. In Section II, we briefly review the SDA and related techniques. In Section III, we examine the algorithms of Section II from a probabilistic perspective, and provide a statistical justification for combining optimal detection ordering with the SDA. In Section IV, we develop a novel closest point algorithm that relies on a probabilistic search to examine candidates of $\mathbf{s}$ in a descending probabilistic order. In Section V, we introduce an efficient hybrid algorithm of our probabilistic search and SDA. We present simulation results in Section VI.

Notation: Upper (lower) bold face letters denote matrices (column vectors); $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^\dagger$ denote matrix transpose, Hermitian, and pseudo-inverse, respectively; $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts; and $E[\cdot]$ stands for expectation. Finally, $\mathcal{G}(n, \lambda)$ denotes the Gamma distribution with density function $f(x) = 1/(\lambda\Gamma(n))(x/\lambda)^{n-1}\exp(-x/\lambda)$, whereas $\Gamma(n)$ denotes the Gamma function. For brevity, $\mathcal{G}(n)$ denotes the standard Gamma distribution where $\lambda = 1$.

II. SDA FOR DETERMINISTIC LATTICES

Even though the closest point problem is in general NP-complete in the worst case [7], there are known relatively efficient algorithms to solve it. These include the SDA of Fincke and Pohst [8], and Kannan's algorithm [12]. A slightly modified version of the SDA will be overviewed here.

The basic idea behind SDA is to search for $\hat{\mathbf{s}}_{ML}$ in a hypersphere of radius r centered at $\mathbf{y}$. Even though points in this hypersphere are searched exhaustively, the calculation is performed recursively so that intermediate computations are efficiently reused. Specifically, it is easy to see that the Euclidean distance between $\mathbf{y}$ and an arbitrary lattice point $\mathbf{x} = \mathbf{H}\mathbf{s}$ within radius r is:

$$\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 = \|\mathbf{H}(\mathbf{s} - \hat{\mathbf{s}})\|^2 = \|\mathbf{R}(\mathbf{s} - \hat{\mathbf{s}})\|^2 \leq r^2, \tag{2}$$

where $\hat{\mathbf{s}} := \mathbf{H}^{\dagger}\mathbf{y}$ is the unconstrained least-squares solution, and $\mathbf{R}$ is the upper triangular matrix in the QR decomposition of $\mathbf{H}$ defined as $\mathbf{R} := [r_{i,j}]$ with $r_{i,i} > 0$ and $r_{i,j} = 0$ for $i > j$. Rewriting (2) componentwise starting with $r_{N,N}$, it follows that

$$\begin{aligned}\|\mathbf{R}(\mathbf{s}-\hat{\mathbf{s}})\|^2 &= r_{N,N}^2\big|s_N - \hat{s}_N\big|^2 + r_{N-1,N-1}^2\Big|s_{N-1} - \hat{s}_{N-1} + \frac{r_{N-1,N}}{r_{N-1,N-1}}(s_N - \hat{s}_N)\Big|^2 + \cdots \\ &= r_{N,N}^2|s_N - \rho_N|^2 + r_{N-1,N-1}^2|s_{N-1} - \rho_{N-1}|^2 + \ldots \qquad (3)\\ &= r_{N,N}^2[\Re(s_N - \rho_N)]^2 + r_{N,N}^2[\Im(s_N - \rho_N)]^2 + \ldots, \qquad (4)\end{aligned}$$

where

$$\rho_N := \hat{s}_N, \quad \rho_k := \hat{s}_k - \sum_{j=k+1}^{N} \frac{r_{k,j}}{r_{k,k}}(s_j - \hat{s}_j), \quad k = N-1, \ldots, 1. \qquad (5)$$

Separating the real and imaginary parts as in (4), the N-dimensional complex problem of finding $\arg\min_{\mathbf{s}}\|\mathbf{R}(\mathbf{s}-\hat{\mathbf{s}})\|^2$ is converted into a $2N$-dimensional real one. From (5), it follows that ρ_k depends on $s_{k+1}, \ldots, s_N$. Hence, in the RHS of (3), the second term depends on s_N, the third term depends on both s_N and s_{N-1}, etc. This recursive dependency is a direct consequence of the upper triangular structure of $\mathbf{R}$. Since only those lattice points inside a hypersphere around $\mathbf{y}$ are checked, a set of necessary conditions for the candidate s_k can be derived based on (4) for $k = N, N-1, \ldots, 1$

$$\lceil \Re\rho_k - \zeta_k \rceil \le \Re s_k \le \lfloor \Re\rho_k + \zeta_k \rfloor \quad \text{and} \quad \lceil \Im\rho_k - \eta_k \rceil \le \Im s_k \le \lfloor \Im\rho_k + \eta_k \rfloor, \qquad (6)$$

where $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ denote the ceiling and floor operations, $\zeta_k := [\, r^2 - \sum_{i=k+1}^{N} r_{i,i}^2|s_i - \rho_i|^2 \,]^{1/2}/r_{k,k}$ and $\eta_k := [\, r^2 - \sum_{i=k+1}^{N} r_{i,i}^2|s_i - \rho_i|^2 - r_{k,k}^2[\Re(s_i - \rho_i)]^2 \,]^{1/2}/r_{k,k}$. These bounds reduce the search for candidate codewords, thus enabling search complexity reduction.

The SDA proceeds as follows: the candidates for $\Re s_N$ are determined first, say $\Re s_N(1)$, …, $\Re s_N(n)$, which decomposes the original problem into n sub-problems. For each candidate $\Re s_N(k)$, there is a corresponding $2N-1$ dimensional search problem. Accordingly, the square radius for each $(2N-1)$-dimensional sphere can be reduced to:

$$r_k^2 = r^2 - r_{N,N}^2[\Re(s_N(k) - \rho_N)]^2, \quad k = 1, \ldots, n.$$

SDA can be recursively applied to these sub-problems. In fact, this algorithm creates a search tree as depicted in Fig. 1, where the children of the same parent node share the path from their parent to the root. Hence, intermediate calculations are efficiently used. Furthermore, whenever a candidate of $\mathbf{s}$ is found with a distance less than r, the radius of the search sphere is reduced to the new distance.

A. Nulling-Canceling Decoding

Nulling-canceling (NC) finds an *approximately nearest* solution in polynomial time [9]. Also known as the Babai nearest plane algorithm [2], the NC yields:

$$\mathbf{s}_{NC} := [\,\lceil \rho_1 \rfloor, \lceil \rho_2 \rfloor, \ldots, \lceil \rho_{N-1} \rfloor, \lceil \rho_N \rfloor\,]^T, \tag{7}$$

where $\lceil x \rfloor$ denotes the closest Gaussian integer to $x \in \mathbb{C}$. Starting from index N and working downwards to index 1, each s_k is chosen to be $\lceil \rho_k \rfloor$, where ρ_k depends on $s_N, s_{N-1}, \ldots, s_{k+1}$ as in (5). At step k, the distance contribution of the term involving s_k in (3) is minimized without considering its effect on subsequent terms. Therefore, NC is a greedy algorithm in nature.

B. Schnorr-Euchner (SE) Variate of SDA

Recently, a variant of the SDA appeared in both [1] and [5]. The key difference of this algorithm from the conventional SDA lies in a simple ordering of the candidates determined by the necessary conditions in (6). Specifically, if $\Re\rho_k \leq \lceil \Re\rho_k \rfloor$, the candidates for $\Re s_k$ are checked in the following order:

$$\Re s_k = \lceil \Re\rho_k \rfloor,\ \lceil \Re\rho_k \rfloor - 1,\ \lceil \Re\rho_k \rfloor + 1,\ \lceil \Re\rho_k \rfloor - 2, \ldots$$

until upper and lower bounds are reached. With this ordering mechanism, the first lattice point checked by SE-SDA is the NC solution.

C. LLL Lattice Reduction

Without FA constraints, lattice reduction is an effective technique to reduce complexity of SDA; see [1] and the references therein. Since (1) can be easily converted to a real model, we consider the real case here. A lattice Λ can be generated by many matrices: two matrices $\mathbf{H}$ and $\mathbf{H}_r$ generate the same lattice if and only if $\mathbf{H} = \mathbf{H}_r\mathbf{W}$, where $\mathbf{W}$ has integer entries with $\det(\mathbf{W}) = \pm 1$. The basis of Λ are the column vectors of its generator matrix. Among different bases, those with smaller norms are often preferred in terms of complexity [6, Page 83]. Given a generator matrix $\mathbf{H}$, the process of finding an $\mathbf{H}_r$ with basis vectors having small norms is called lattice reduction (LR). A well known polynomial-time LR algorithm is the LLL algorithm [13]. In the following, we will briefly explain how the closest point search can benefit from the LLL algorithm.

Following [8] and [6], we apply the LLL algorithm to the row vectors of $\mathbf{R}^{-1}$ instead of $\mathbf{R}$, (the reason will become clear soon). With LR, we obtain $\mathbf{H}_r^{-1} = \mathbf{W}\mathbf{R}^{-1}$, or equivalently, $\mathbf{R} = \mathbf{H}_r\mathbf{W}$. Plugging to (2), we have

$$\|\mathbf{R}(\mathbf{s} - \hat{\mathbf{s}})\|^2 = \|\mathbf{H}_r\mathbf{W}(\mathbf{s} - \hat{\mathbf{s}})\|^2 = \|\mathbf{H}_r(\mathbf{x} - \hat{\mathbf{x}})\|^2 = \|\mathbf{R}_r(\mathbf{x} - \hat{\mathbf{x}})\|^2, \tag{8}$$

where $\mathbf{x} := \mathbf{W}\mathbf{s}$ and $\hat{\mathbf{x}} := \mathbf{W}\hat{\mathbf{s}}$. Clearly, without FA constraints, $\mathbf{s} \in \mathbb{Z}^N$ and $\mathbf{x} \in \mathbb{Z}^N$. Since $\mathbf{x} = \mathbf{R}_r^{-1}\mathbf{R}_r\mathbf{x}$, we have $x_i = \mathbf{r}_i^T\mathbf{R}_r\mathbf{x}$, where x_i is the ith element of $\mathbf{x}$ and $\mathbf{r}_i^T$ denotes the ith row vector of $\mathbf{R}_r^{-1}$. Applying the Cauchy-Schwarz inequality, we have

$$x_i^2 \leq \|\mathbf{r}_i\|^2\|\mathbf{R}_r\mathbf{x}\|^2 \leq \|\mathbf{r}_i\|^2\, r^2. \tag{9}$$

Due to the LR on the row vectors of $\mathbf{R}^{-1}$, the rows of $\mathbf{H}_r^{-1}$ have small norm; i.e., the norm of $\mathbf{r}_i$ is small, which implies the upper bound in (9) is reduced, and thus the number of candidates for x_i are reduced accordingly.

Unfortunately, LR does not facilitate incorporating FA constraints. To be specific, let $\mathbf{s} \in \mathcal{S}^N$, where $\mathcal{S}$ is the P-ary PAM constellation $\mathcal{S} := \{-P/2+1, \ldots, P/2\} \subset \mathbb{Z}$, and P is an even integer. Although checking the FA constraint on $\mathbf{s}$ is easy, the corresponding $\mathbf{s} = \mathbf{W}^{-1}\mathbf{x}$ for the candidate $\mathbf{x}$ after LR, may not be in $\mathcal{S}^N$. Even though $\mathbf{W}^{-1}$ is an integer matrix, checking the FA constraint for $\mathbf{x}$ is much more difficult. The benefit of LR is further limited for certain constellations; for example, in the widely used 4-QAM, binary symbols are transmitted per dimension. Hence, we have a natural constraint from the constellation, which eliminates many nearby lattice points that are not in $\mathcal{S}^N$. On the other hand, any point in the integer lattice is initially considered to be valid by SDA, and many of them are checked unnecessarily. Nonetheless, when the constellation size is large, LR offers an effective means of reducing complexity.

III. SDA FOR RANDOM LATTICES

Modeling $\mathbf{H}$ as a random matrix with independent and identically distributed (i.i.d.) zero-mean unit-variance complex Gaussian $\mathcal{CN}(0,1)$ entries is widely applicable in wireless links [11]. In this section, we will exploit this statistical channel model to improve computational efficiency of the SDA. For the random matrix $\mathbf{H} = \mathbf{QR}$, the elements of the upper triangular matrix $\mathbf{R}$ are distributed independently according to: $r_{i,i}^2 \sim \mathcal{G}(M+1-i)$, and $|r_{i,j}|^2 \sim \mathcal{G}(1)$ for $i < j$ [16]. Multiplying both sides of (1) with the unitary matrix $\mathbf{Q}^H$, we obtain

$$\tilde{\mathbf{y}} = \mathbf{R}\mathbf{s} + \tilde{\mathbf{n}}, \tag{10}$$

where $\tilde{\mathbf{y}} := \mathbf{Q}^H\mathbf{y}$ and $\tilde{\mathbf{n}} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$. To decode s from $\tilde{\mathbf{y}}$, a series of scalar decisions has to be made on the following set of equations [c.f. (10)]:

$$\check{y}_N = r_{N,N}s_N + \tilde{n}_N, \ldots, \check{y}_1 = r_{1,1}s_1 + \tilde{n}_1, \tag{11}$$

where $\check{y}_k := \tilde{y}_k - \sum_{i=k+1}^{N} r_{k,i}s_k$. Since $r_{i,i}$ is real and positive, a set of $2N$ equations can be obtained from (11):

$$\Re\check{y}_N = r_{N,N}\Re s_N + \Re\tilde{n}_N, \quad \Im\check{y}_N = r_{N,N}\Im s_N + \Im\tilde{n}_N, \quad \ldots \tag{12}$$

Each decision in (12) corresponds to a branch in the search tree depicted in Fig. 1. Whenever an error occurs, the corresponding sub-tree is searched in vain. An early error corresponds to a high dimensional sub-tree, which leads to a major waste of computations. Since $r_{i,i}^2 \sim \mathcal{G}(M+1-i)$, the probability that $r_{i,i}$ takes small values increases with i. Hence, it is more likely to make an error early in the decision sequence of (12). Unfortunately, with the single goal of efficiently reusing intermediate calculations, SDA adopts a *depth-first* search strategy, which does not account for this statistical property of $\mathbf{R}$.

With the random model, it follows from (11) that the SE-SDA examines the candidates of s_k in a decreasing probability order. Hence, the ordering mechanism of the SE-SDA improves search efficiency. Nonetheless, the distributions of $r_{N,N}, \ldots, r_{1,1}$ remain unaffected by this ordering, which indicates that further computational savings are possible.

Under our random model, the reason that LR improves the efficiency of SDA is as follows: Geometrically, $\det(\mathbf{H})$ is the volume of the fundamental region for the lattice. This determinant is invariant under different bases, which implies that the set of reduced basis vectors is not far from being orthogonal. Hence, the off diagonal elements of $\mathbf{R}_r$ in (8) have relatively small magnitudes. This guarantees that the decision on one equation in (11) has less effect on other equations with LR. Furthermore, the diagonal entries of $\mathbf{R}_r$ are relatively large in magnitude, which enhances SNR (and thus lowers the probability of error) for all the decisions in (11).

A. *Optimal Detection Ordering*

In this section, optimal detection ordering is examined as a method to improve the statistical properties of $\mathbf{R}$, which will in turn reduce the complexity of SDA. For a deterministic $\mathbf{H}$, this ordering mechanism was mentioned in [8] based on a heuristic argument (see also [14]). Here, we provide a theoretical justification based on our random model of $\mathbf{H}$.

Optimal detection ordering proceeds as follows: Rearranging the columns of $\mathbf{H} := [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]$ in an ascending order of their squared norm, i.e., $\|\mathbf{h}_{o(1)}\|^2 \leq \|\mathbf{h}_{o(2)}\|^2 \leq \ldots \leq \|\mathbf{h}_{o(N)}\|^2$, we obtain the ordered generator matrix as $\mathbf{H}_o := [\mathbf{h}_{o(1)}, \mathbf{h}_{o(2)}, \ldots, \mathbf{h}_{o(N)}]$. With the QR decomposition of $\mathbf{H}_o = \mathbf{Q}_o \mathbf{R}_o$, we are now ready to characterize the distribution of the entries of $\mathbf{R}_o$ denoted by $r_{o,i,j}$. Let $\mathbf{h}_{o(i)} := \sqrt{X_i}\boldsymbol{\theta}_i$, where $\boldsymbol{\theta}_i^H \boldsymbol{\theta}_i = 1$. Since $\{\mathbf{h}_i\}$s are i.i.d. with $\mathbf{h}_i \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$, we deduce that X_i is the ith order statistic of N independent $\mathcal{G}(M)$ random variables such that

$X_1 \leq X_2 \leq \ldots \leq X_N$. Because of this ordering, X_i's are necessarily dependent. However, $\{\boldsymbol{\theta}_i\}$s are i.i.d. uniformly distributed on the unit sphere in $\mathbb{R}^{2M}$ denoted by S^{2M-1}, where we have identified $\mathbb{C}^M$ with $\mathbb{R}^{2M}$ for simplicity. With this geometric interpretation, it is clear that the distribution of $\{\boldsymbol{\theta}_i\}$s remains invariant under orthogonal transformations. That is, for any unitary matrix $\mathbf{U}$, $p(\mathbf{U}^H\boldsymbol{\theta}) = p(\boldsymbol{\theta})$, where the p.d.f. of $\boldsymbol{\theta}$ is: $p(\boldsymbol{\theta}) = 1/A_{2M-1}$, $\boldsymbol{\theta}^H\boldsymbol{\theta} = 1$, and A_{2M-1} is the surface area of S^{2M-1} determined by

$$A_{2M-1} = \frac{2\pi^M}{\Gamma(M)}.$$

More generally, $\mathbf{Q}$ in the QR decomposition of $\mathbf{H}$ is uniformly (a.k.a. isotropically) distributed over the Stiefel manifold $\{\mathbf{Q} : \mathbf{Q}^H\mathbf{Q} = \mathbf{I}\}$ [10]. The p.d.f. of X_i is

$$f_{X_i}(x_i) = \frac{N!}{(i-1)!(N-i)!}\big[F(x_i)\big]^{i-1}\big[1 - F(x_i)\big]^{N-i} f(x_i),$$

where $F(x)$ and $f(x)$ are the c.d.f. and p.d.f. of a standard $\mathcal{G}(M)$ random variable, respectively. Let $\mathbf{Q}_o := [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N]$. Based on the QR decomposition of $\mathbf{H}_o$, we obtain

$$r_{o,i,i}^2 = X_i\big[1 - \sum_{k=1}^{i-1}(\mathbf{q}_k^H\boldsymbol{\theta}_i)^2\big] \equiv X_i\big[1 - \sum_{k=1}^{i-1}\boldsymbol{\theta}_i^2(k)\big],$$

where $\boldsymbol{\theta}_i(k)$ is the kth element of $\boldsymbol{\theta}_i$ and the equivalence follows from the fact that the distribution of $\boldsymbol{\theta}_i$ is invariant under the orthogonal transformation $\mathbf{Q}_o$. To determine the expectation of $r_{o,i,i}^2$, we first calculate $E[\boldsymbol{\theta}_i^2(k)]$. Since $\boldsymbol{\theta}_i^H\boldsymbol{\theta}_i = 1$, it follows that

$$E[\boldsymbol{\theta}_i^H\boldsymbol{\theta}_i] = E\left[\sum_{k=1}^{M}\boldsymbol{\theta}_i^2(k)\right] = ME[\boldsymbol{\theta}_i^2(k)] = 1,$$

where the second equality follows from the symmetry of the distribution. Hence, $E[\boldsymbol{\theta}_i^2(k)] = 1/M$, for $k = 1, \ldots, M$. Due to the independence between X_i and $\boldsymbol{\theta}_i$, we have $E[r_{o,i,i}^2] = E[X_i][1-(i-1)/M]$. The moments of the order statistics of i.i.d. Gamma random variables are available in closed form [4, Page 43]. Because the expressions are lengthy, exact formulas are omitted for brevity. Instead, we provide an illustrative example.

Example 1: Here, we consider $E[r_{i,i}^2]$ and $E[r_{o,i,i}^2]$ for $M = N = 16$. The mean values for the two cases are compared in Table I. Without ordering, it is clear that $E[r_{i,i}^2] = M + 1 - i$, since $r_{i,i}^2 \sim \mathcal{G}(M+1-i)$. With ordering, we observe from this table that $E[r_{o,i,i}^2]$ increases considerably as i approaches M. Hence, the probability of an early error in the sequence of decisions in (11) is lower, which explains why complexity is reduced with ordering.

Optimal detection ordering can be conveniently combined with SE-SDA. An attractive feature of the resulting algorithm is that the first lattice point checked is the NC solution with optimal detection ordering [18], whereas the first vector examined by the original SE-SDA is the NC solution. NC with optimal detection ordering is known to exhibit improved error performance as compared to conventional NC.

IV. A NEW SEARCH STRATEGY

The SDA follows a depth-first search strategy, where the order in which the lattice points are examined, is determined by the recursive search tree structure. On the other hand, given the realizations of $r_{i,i}$, a reliability order for candidates of $\mathbf{s}$ is suggested by the set of equations in (11). Taking into account the statistics of $\mathbf{v}$ only, is it possible to check lattice points using a probabilistic ordering? The following algorithm provides such an approach.

A. Algorithm

Let us define $u_i := \lceil \Re(s_i - \rho_i) \rfloor$ and $\mu_i = \Re(s_i - \rho_i) - u_i$, where $u_i \in \mathbb{Z}$ and $\mu_i \in (-1/2,\, 1/2)$. Similarly, let $v_i := \lceil \Im(s_i - \rho_i) \rfloor$ and $\nu_i = \Im(s_i - \rho_i) - v_i$, where $v_i \in \mathbb{Z}$ and $\nu_i \in (-1/2,\, 1/2)$. With these definitions, (4) becomes

$$\|\mathbf{R}(\mathbf{s} - \hat{\mathbf{s}})\|^2 = \sum_{i=1}^{N} r_{i,i}^2 \big[(u_i + \mu_i)^2 + (v_i + \nu_i)^2\big],$$

where the vector $\mathbf{c} := [u_1, v_1, u_2, v_2, \ldots, u_N, v_N]^T$ uniquely determines a candidate for $\mathbf{s}$. The possible $\{\mathbf{c}\}$s are determined next. Since $|\mu_i| < 1/2$ and $u_i \in \mathbb{Z}$, it follows that

$$\sum_{i=1}^{N} r_{i,i}^2 \mu_i^2 \leq \sum_{i=1}^{N} r_{i,i}^2 (u_i + \mu_i)^2. \qquad (13)$$

Based on (13) and the triangle inequality

$$\sqrt{\sum_{i=1}^{N} r_{i,i}^2 u_i^2} \leq \sqrt{\sum_{i=1}^{N} r_{i,i}^2 \mu_i^2} + \sqrt{\sum_{i=1}^{N} r_{i,i}^2 (u_i + \mu_i)^2},$$

we conclude that

$$\sum_{i=1}^{N} r_{i,i}^2 u_i^2 \leq 4 \sum_{i=1}^{N} r_{i,i}^2 (u_i + \mu_i)^2.$$

Similar inequalities hold for v_i and ν_i. Hence, all the $\{\mathbf{c}\}$s that must be checked satisfy

$$Q(\mathbf{c}) := \sum_{i=1}^{N} r_{i,i}^2 [u_i^2 + v_i^2] \leq 4\, e_{\min}^2, \tag{14}$$

where $e_{\min}$ is the Euclidean distance from the received vector $\mathbf{y}$ to its closest lattice point. That is, the candidates depend on the true minimum distance albeit through a bound. Furthermore, prior knowledge of $e_{\min}$ is not necessary.

The new algorithm breaks the search tree structure of SDA, and follows these steps:

S1: Calculate the distance of the NC solution $\mathbf{s}_{NC}$ defined in (7). Let $e_{\min}^2 = \|\mathbf{R}(\mathbf{s}_{NC} - \hat{\mathbf{s}})\|^2$;

S2: Generate a vector $\mathbf{c}$ such that $Q(\mathbf{c})$ is the next smallest;

S3: If $Q(\mathbf{c}) > 4\, e_{\min}^2$, goto **S6**;

S4: Check the FA constraint on the $\mathbf{s}$ corresponding to $\mathbf{c}$. If the $\mathbf{s}$ is invalid, goto **S2**;

S5: Calculate $\|\mathbf{R}(\mathbf{s} - \hat{\mathbf{s}})\|^2$, update $e_{\min}^2 = \min\{e_{\min}^2, \|\mathbf{R}(\mathbf{s} - \hat{\mathbf{s}})\|^2\}$, and goto **S2**;

S6: Return the best candidate for $\mathbf{s}$ and $e_{\min}$.

Several remarks are now in order:

Remark 1: The list of $\mathbf{c}$ vectors is generated in an ascending order of $Q(\mathbf{c})$. This list determines the order in which the candidates of $\mathbf{s}$ are examined. Furthermore, the $\mathbf{c}$ list is shared by all the decodings based on the same $\mathbf{H}$ realization, which is often encountered in coherent communications;

Remark 2: The NC solution always corresponds to $\mathbf{c} = \mathbf{0}$, since $Q(\mathbf{0}) = 0$;

Remark 3: Initially, $e_{\min}$ may not be the distance of $\mathbf{y}$ from its closest lattice point. Due to the update of $e_{\min}$ in S5, $e_{\min}$ is guaranteed to be the true shortest distance, when the algorithm terminates.

B. Probabilistic Ordering

In this section, we justify the new algorithm from a probabilistic perspective. Since each $\mathbf{c} \in \mathbb{Z}^{2N}$ corresponds to a unique candidate for $\mathbf{s}$, we will determine the probability that $\mathbf{c}$

corresponds to the transmitted vector $\mathbf{s}$. We start with the scalar case. Based on the definition of ρ_i in (5) and the ith equation from (11), it follows that $\rho_i = \check{y}_i / r_{i,i}$, and hence $\rho_i = s_i + \check{n}_i$, where $\check{n}_i := \tilde{n}_i / r_{i,i}$ and $\check{n}_i \sim \mathcal{CN}(0, \sigma^2/r_{i,i}^2)$. Equivalently, $s_i = \rho_i + \check{n}_i$, since $-\check{n}_i$ has the same distribution as $\check{n}_i$. Separating real and imaginary parts, the probability of u_i is the probability that $\Re s_i \in [\Re\rho_i + u_i - 0.5,\ \Re\rho_i + u_i + 0.5]$ as illustrated in Fig. 2. The latter is calculated as:

$$p(u_i) = 2\exp\Big(-\frac{r_{i,i}^2 u_i^2}{\sigma^2}\Big)\int_0^{\frac{1}{2}} \frac{1}{\sqrt{\pi\sigma^2/r_{i,i}^2}} \exp\Big(-\frac{r_{i,i}^2 x^2}{\sigma^2}\Big)\cosh\Big(\frac{2u_i}{\sigma^2/r_{i,i}^2}x\Big)dx \qquad (15)$$

$$\approx 2\exp\Big(-\frac{r_{i,i}^2 u_i^2}{\sigma^2}\Big)\int_0^{\frac{r_{i,i}}{\sqrt{2}\sigma}} \frac{1}{\sqrt{2\pi}}\exp\Big(-\frac{x^2}{2}\Big)dx := k_i \exp\Big(-\frac{r_{i,i}^2 u_i^2}{\sigma^2}\Big),$$

where we have applied a first order approximation for $\cosh(x) = 1 + x^2/2 + x^4/4! + \cdots$. This approximation is reasonable, since if u_i is approximately zero, $\cosh(2u_i r_{i,i}^2 x/\sigma^2)$ is close to 1; otherwise, $\exp(-r_{i,i}^2 u_i^2/\sigma^2)$ becomes dominant. With this approximation, we observe that $p(u_i)$ depends on $r_{i,i}^2 u_i^2$ only. Since $\check{n}_i$'s are independent, the probability of $\mathbf{c}$ can be approximated by

$$p(\mathbf{c}) \approx \exp\Big(-\frac{\sum_{i=1}^N r_{i,i}^2[u_i^2 + v_i^2]}{\sigma^2}\Big)\prod_{i=1}^N k_i^2. \qquad (16)$$

Hence, $p(\mathbf{c})$ is approximately determined by $Q(\mathbf{c})$. Since in our algorithm $\mathbf{c}$'s are examined with increasing order of $Q(\mathbf{c})$, it follows that the candidates of $\mathbf{s}$ are examined according to a decreasing probability order. Based on this observation, the new closest point scheme is a probabilistic search algorithm.

C. Complexity Analysis

In this section, we assess the average complexity of our probabilistic search algorithm without detection ordering, which also upper bounds the case with ordering. The complexity of decoding is averaged over the noise and channel realizations. To facilitate our analysis, we first give the p.d.f. of the linear combination of K independent Gamma variables in closed form.

Lemma 1: If $\{X_i\}_{i=1}^K$ are independently distributed according to $\mathcal{G}(n_i)$, and $l_i > 0$, $n_i \in \mathbb{N}$, and

$l_i \neq l_j$, $\forall\ i \neq j$, then the p.d.f. of $Y = \sum_{i=1}^K l_i X_i$ is given by

$$f_Y(y) = \sum_{k=1}^{K} \sum_{\sum_{q=1}^K m_q = n_k - 1} C(k, \mathbf{m}) \frac{1}{l_k} \frac{1}{\Gamma(m_k+1)} \left(\frac{y}{l_k}\right)^{m_k} e^{-y/l_k},$$

where $\mathbf{m} := [m_1, \ldots, m_K]^T$ and

$$C(k, \mathbf{m}) := (-1)^{n_k - 1 - m_k} \prod_{\substack{q=1 \\ q \neq k}}^{K} \binom{n_q + m_q - 1}{m_q} \left(\frac{l_q}{l_k}\right)^{m_q} \left(1 - \frac{l_q}{l_k}\right)^{-(n_q + m_q)}.$$

Proof: Using the Laplace transform of Y

$$\mathcal{L}_r(f(y)) = \frac{1}{(1+l_1 r)^{n_1}} \frac{1}{(1+l_2 r)^{n_2}} \cdots \frac{1}{(1+l_K r)^{n_K}},$$

$f_Y(y)$ can be uniquely determined by the inverse Laplace transform as:

$$f(y) = \frac{1}{2\pi i} \int_{c-i\infty}^{c_i\infty} \mathcal{L}_r(f(y)) e^{ry} dr = \sum_{k=1}^{K} R_k,$$

where R_k's are the K residues of $\mathcal{L}_r(f(y))$, calculated using:

$$R_k = \frac{1}{\Gamma(n_k)} \frac{d^{n_k-1}}{dr^{n_k-1}} \Big[(1+l_k r)^{n_k} \mathcal{L}_r(f(y))\Big]\Big|_{r=-1/l_k}.$$

The lemma follows by applying the multi-nomial formula to the derivatives. ■

In the following, we calculate the average probability that a particular $\mathbf{c}$ is checked by the probabilistic search algorithm. Letting $V := \|\mathbf{v}\|^2$, the p.d.f. of V is:

$$p_V(v) = \frac{1}{\sigma^2 \Gamma(M)} \left(\frac{v}{\sigma^2}\right)^{M-1} e^{-v/\sigma^2}, \quad v \geq 0.$$

The probability that $\mathbf{c} := [u_1, v_1, u_2, v_2, \ldots, u_N, v_N]^T$ needs to be checked conditioned on the noise realization $V = v$, is: $P(\sum_{i=1}^N r_{i,i}^2 (u_i^2 + v_i^2) \leq 4v)$. We know combine those $r_{i,i}^2$'s with equal coefficients into one random variable. Let the resulting K variables and their coefficients be X_i and l_i, respectively, where the X_i's are independent, $X_i \sim \mathcal{G}(n_i)$, and $l_i > 0$. The conditional probability can be calculated as

$$P_{\mathbf{c}|V} = P\left(\sum_{i=1}^{K} l_i\, X_i \leq 4v\right) = \sum_{k=1}^{K} \sum_{\mathbf{m}} C(k, \mathbf{m}) \left[1 - e^{-4v/l_k} \sum_{p=0}^{m_k} \frac{(4v/l_k)^{m_k - p}}{\Gamma(m_k + 1 - p)}\right],$$

where we have applied Lemma 1 and the following identity:

$$\frac{1}{\Gamma(n+1)}\int_0^x t^n e^{-t}\, dt = 1 - e^{-x}\sum_{k=0}^{n}\frac{x^{n-k}}{\Gamma(n+1-k)}.$$

The average probability $P_{\mathbf{c}}$ that $\mathbf{c}$ needs to be checked is:

$$P_{\mathbf{c}} = \int P_{\mathbf{c}|V}(v)P_V(v)dv = c_1 - \sum_{k=1}^{K}\sum_{\mathbf{m}}\sum_{p=0}^{m_k} D(k,\mathbf{m},p)\left[\frac{4}{l_k}+\frac{1}{\sigma^2}\right]^{-(M+m_k-p)},$$

where

$$c_1 := \sum_{k=1}^{K}\sum_{\mathbf{m}} C(k,\mathbf{m}) \quad \text{and} \quad D(k,\mathbf{m},p) := C(k,\mathbf{m})\binom{M+m_k-p-1}{m_k-p}\frac{4^{m_k-p}}{l_k^{m_k-p}\sigma^{2M}}.$$

Hence, the average number of $\{\mathbf{c}\}$s checked is:

$$N_{\text{av}} = \sum_{\mathbf{c}} P_{\mathbf{c}},$$

where the list of $\mathbf{c}$ can be generated in an ascending order of $\sum_{i=1}^{N}(M+1-i)(u_i^2+v_i^2)$, and N_{av} can be well approximated by a finite number of $\{\mathbf{c}\}$s.

V. Hybrid Algorithm

The new closest point algorithm examines the candidates of $\mathbf{s}$ in a probabilistic order. At high SNR, we can observe from (16) that the probability of each candidate decreases considerably. Hence, only a few $\mathbf{c}$ candidates need to be checked. Since it is more likely that candidates differ at positions near the end of $\mathbf{c}$, the reusable intermediate calculations are limited. On the other hand, at low and medium SNR, the probability of each $\mathbf{c}$ decreases slowly, many $\{\mathbf{c}\}$s have to be checked, and efficient reuse of intermediate calculations becomes a necessity.

To this end, we introduce a hybrid algorithm combining probabilistic search with SDA, yielding approximate solutions to the closest point problem. By trading off performance for complexity at high SNR, it achieves error performance comparable to SDA with reduced complexity, and thereby fills in the gap both in terms of performance and complexity between two extremes: the low complexity NC that is clearly suboptimum and the high complexity SDA with near-ML performance.

A. Exploiting FA and Probabilistic Ordering

Without LR, FA property of the constellation can lower complexity, when the alphabet size per dimension is small. Without reordering the columns of $\mathbf{H}$, we have $r_{i,i}^2 \sim \mathcal{G}(M+1-i),\ 1 \leq i \leq N$. Since some realizations of $r_{i,i}$ are likely to be small, ordering can bring improvement. But still, for a small $r_{i,i}$, the variance of $\check{n}_k$ in $\rho_k = s_k + \check{n}_k$ is considerably magnified, since $\check{n}_k \sim \mathcal{CN}(0, \sigma^2/r_{i,i}^2)$. As a result, the number of possible candidates for u_i and v_i is large [c. f. (14)]. However, many of them are not valid constellation points. On the other hand, it is straightforward to determine the probability order for all s_k candidates from a small set. The real question is how to decide when to exploit the FA property, and when to pursue the probabilistic search. We suggest the following simple approach.

Let the constellation size per dimension be an even integer P. If the interval $[\Re\rho_k - P/2, \Re\rho_k + P/2]$ contains the true $\Re s_k$ with high probability (say 0.999), we decide that dimension k participates in calculating $\sum_i r_{i,i}^2(u_i^2 + v_i^2)$; otherwise, we omit that dimension and exploit the FA property. Explicitly, the threshold for $r_{i,i}$ is calculated by $r_{th} := \text{erf}^{-1}(0.999)2\sigma/P$, where $\text{erf}^{-1}(x)$ is the inverse of $\text{erf}(x) := (2/\sqrt{\pi})\int_0^x \exp(-t^2)\, dt$. If $r_{i,i} < r_{th}$, we exploit the FA; otherwise, we pursue probabilistic ordering.

B. Performance-Oriented Fast Stopping Criterion

In this section, the first key difference between the exact probabilistic search and the hybrid algorithm is clarified. In the exact search, the candidates s are examined in a descending probability order, which maximizes the chance to find the closest point early. However, even when the solution has been reached, the algorithm continues until the bound in (14) is achieved, thus carrying out unnecessary calculations. Certainly, an efficient algorithm should find the closest point as quickly as possible, and stop as fast as possible. Hence, a fast stopping criterion with controllable performance degradation is indispensable for an efficient approximate algorithm.

We develop such a criterion based on binary hypothesis testing next. Whenever a candidate s' for the transmitted vector s is found, we test the corresponding distance V. Let the null hypothesis, H_0, correspond to having $V = \|\mathbf{v}\|^2$, where $\mathbf{v}$ is the true noise vector in (1). This is the case when $\mathbf{s}' = \mathbf{s}$. Let the alternative hypothesis, H_a, be $V = \|\mathbf{v}+\mathbf{h}\|^2$, where $\mathbf{h} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$. The alternative occurs when $\mathbf{s}'$ differs from $\mathbf{s}$ in only one position, and the difference between the two corresponding elements may be either ± 1 or $\pm j$. Only such an error pattern is considered, because it is the dominant one at high SNR. Let the testing threshold be V_{th}. Whenever $V < V_{th}$, we decide that H_0 is true and stop searching; otherwise, we continue to search. Since $\mathbf{v}$ and $\mathbf{h}$ are independent, it follows that $V \sim \mathcal{G}(M, \sigma^2)$ under H_0, while $V \sim \mathcal{G}(M, \sigma^2 + 1)$ under H_a. We illustrate how we determine V_{th} through the following example.

Example 2: Let us consider a communication system where $M = N = 32$. It could correspond to either a BLAST-type space-time system, or to a symbol synchronous CDMA system. The modulation employed is 16-QAM, and the system operates at receive SNR $= 24$ dB, which corresponds to 5.9382 dB in terms of transmit SNR. Receive SNR seems to be preferred in space-time settings, whereas transmit SNR is often used in multiuser scenario. In a space-time setting, 24dB belongs to the high SNR regime, whereas in multiuser detection, 5.9382dB is at best in the medium SNR regime. Since our work has potential applications in both areas, we will provide both values. The density functions for V under H_0 and H_a are depicted in Fig. 3. Since $V \sim \mathcal{G}(M, \sigma^2 + 1)$ under H_a, the probability of missing with the threshold V_{th} can be calculated in closed form as:

$$P_M(V_{th}) = 1 - e^{-V_{th}/(1+\sigma^2)} \sum_{k=0}^{M-1} \frac{1}{k!} \left(\frac{V_{th}}{1+\sigma^2} \right)^k,$$

which equals the probability that a pairwise error happens, but remains undetected. Continuing with our example, let the target P_M be 10^{-6}. Usually, P_M is determined by either the target symbol error rate (SER), or, the actual SER. For a certain P_M, the corresponding V_{th} can be calculated by standard toolboxes, e.g., *Mathematica*. In this case, $V_{th} = 29.6099$. Now that we have obtained V_{th}, is this testing effective? The answer becomes clear when we calculate the probability that $\|\mathbf{v}\|^2 < V_{th}$ denoted by P_{in}. For this example, $P_{in} = 0.999971$. Hence, the fast stopping criterion is effective for almost all noise realizations.

Our next task is to choose a proper P_M based on the actual SER performance. Letting P_e be the actual SER, we choose $P_M = P_e$. In the worst case, the resulting SER will be doubled. When plotting SER in a log scale, this causes the SER curve to shift upwards for about $\log_{10}(2) = 0.3$. It turns out that the performance degradation is less than this, because of all the pairwise errors missed by the fast stopping criterion (a fraction remains undetectable under the ML decision rule). Hence, a considerable portion of P_M and P_e is overlapping, and the SER is between P_e and $2P_e$.

C. Algorithm Description

Since our algorithm is a hybrid of probabilistic search with an improved SDA, we discuss this variate of SDA first. The SDA with increasing radius search (IRS) proceeds as follows: Letting $r_1 < r_2 < \ldots < r_n$ be a set of sphere radii, start the SDA with search radius r_1. If a candidate is found, then stop. Otherwise, run the SDA again with the next radius r_2, and so on. This algorithm was initially mentioned in [17] without giving the set of radii. The set of radii can be determined as $r_{0.9}, r_{0.99}, \ldots$, where $r_{0.9}$ is determined by $P(\|\mathbf{v}\|^2 \leq r_{0.9}) = 0.9$, etc [11]. Here, we employ a modified version of the SE-SDA with IRS and our stopping criterion. After increasing the radius for a few times without success, the NC solution is returned.

We first provide a brief description of our hybrid algorithm. Differences from its exact counterpart will be further elaborated afterwards. The hybrid algorithm without LR works as follows:

$\langle 1 \rangle$ Order the columns of $\mathbf{H}$ as in Section III-A, and order the entries of $\hat{\mathbf{s}}$ accordingly;

$\langle 2 \rangle$ Determine r_{th} and those dimensions exploiting the FA as in Section V-A;

$\langle 3 \rangle$ Calculate V_{th} as in Section V-B;

$\langle 4 \rangle$ Generate a $\mathbf{c}$ list with increasing $Q(\mathbf{c})$ until $Q(\mathbf{c}) > B$, where B will be specified later;

$\langle 5 \rangle$ Calculate a list of sphere radii $r_1, r_2, \ldots, r_n$ to be used by the IRS;

$\langle 6 \rangle$ Examine $\{\mathbf{c}\}$s in the ordered list sequentially with the sphere radius upper bound $r_U > r_n$. Whenever a candidate for $\mathbf{s}$ is found to be within distance V_{th}, goto $\langle 8 \rangle$;

$\langle 7 \rangle$ Execute the SE-SDA with IRS;

$\langle 8 \rangle$ Order the entries of the best candidate with the inverse ordering in $\langle 1 \rangle$ and return.

Besides our stopping rule, another key feature lies in $\langle 4 \rangle$, where instead of achieving the upper bound as in (14), the hybrid algorithm stops when $Q(\mathbf{c}) > B$. The argument is that the complexity of the initial probabilistic search increases linearly with the number of $\{\mathbf{c}\}$s checked. However, the probability of a $\mathbf{c}$ decreases exponentially with $Q(\mathbf{c})$. After a certain threshold, it is not advantageous to check additional $\{\mathbf{c}\}$s in this manner, since it is highly likely that the individual search is in vain and a lot of reusable intermediate calculations are discarded. A heuristic method to determine B is described next. Since $\mathbf{c} = \mathbf{0}$ corresponds to the NC solution, its probability can be calculated based on (15) and (16) as $\prod_i k_i^2$. Using this probability as a reference, we define the relative probability of a certain $\mathbf{c}$ as: $P_r(\mathbf{c}) := P(\mathbf{c}) / \prod_i k_i^2 = \exp(-\sum_i r_{i,i}^2 [u_i^2 + v_i^2]/\sigma^2)$. For high SNR, whenever the relative probability is less than a threshold, say 10^{-5}, we stop checking candidates one by one, hence, $B = 5 \ln(10) \sigma^2$. Albeit intuitively appealing, this method is not sufficient to determine the optimal B value. Clearly, the parameter B controls when to stop the probabilistic search and switch on the SDA.

It can be observed from (16) that the decreasing rate of the sequence of probabilities, $P(\mathbf{c}_0)$, $P(\mathbf{c}_1)$, $\ldots$, increases with SNR. Exploiting this ordering, the efficiency of our hybrid algorithm improves as SNR increases. If no candidate is found by the initial probabilistic search, SE-SDA with IRS is employed as in $\langle 7 \rangle$ to guarantee error performance. At high SNR, the probability of an unsuccessful initial search is small.

The upper bound r_U on the search radius in $\langle 6 \rangle$ can be either initialized to infinity, or determined as follows: $r_U := r_{1-P_e/100}$, where P_e is the actual SER. We suggest the latter, since it causes negligible performance loss, but saves some calculations.

VI. Simulations

The complexity reduction brought by various techniques and algorithms described in this paper are tested in this section. Since different algorithms may incur considerably different overhead, the comparison is based on the actual CPU time instead of the number of flops.

Test Case 1: In this example, we show the running time improvement obtained by optimally ordering the columns of $\mathbf{H}$, when $M = N = 16$, and SE-SDA is implemented with IRS. The set of radii is chosen to be $[r_{0.9}, r_{0.99}, \ldots, r_{0.99999}]$. For each channel realization, 100 noise realizations are generated, while the number of channel realizations is determined depending on the SNR. With and without ordering, the algorithm essentially achieves the same SER performance depicted in Fig. 4. However, the running times are considerably different. Let t_o and $t_{w/o}$ be the running time of the SDA with and without ordering, respectively. The ratio $t_{w/o}/t_o$ is shown in Table II along with the number of channel realizations per SNR. It is evident that optimal detection ordering is a simple, yet effective technique to improve the SDA. Furthermore, the detection ordering is performed only once per channel realization.

Test Case 2: To demonstrate the efficiency of the hybrid algorithm at high SNR, we compare its running time and performance with those of the SE-SDA equipped with both detection ordering and IRS at receive SNR $= 24$ dB. The set of initial radii is the same as in Test Case 1. The dimensions under consideration are listed in Table III along with the transmit SNR, the probability of missing P_M, the stopping threshold V_{th}, and P_{in}. We set $P_M = P_e$, where an approximate SER P_e of the SE-SDA is obtained through Monte-Carlo simulations. The threshold V_{th} is determined as described in V-B. In the simulation, $50,000$ channel realizations are generated, and for each realization a short $\mathbf{c}$ list is generated with $B = 5\ln(10)\sigma^2$. This choice of B is by no means optimal. Fine tuning B can further reduce complexity. For each channel, 200 noise instances sharing the same $\mathbf{c}$ list are generated. The running time reduction of our hybrid algorithm is shown in Fig. 5. Let t_{SE} and t_{HA} denote the running time of the SE-SDA and the hybrid algorithm, respectively. The ratio of t_{SE}/t_{HA} is shown in Table IV with the SER performance for each dimension. An intriguing fact is that our hybrid algorithm achieves lower SER than the SE-SDA except for $M = 4$. This is because the SE-SDA stops increasing the search radius after a few times, returning the NC estimate if no candidate is found. On the other hand, the hybrid algorithm first resorts to the probabilistic search. If the initial search fails to find a candidate with distance less than V_{th}, it switches to the SE-SDA. If still no candidate is found, the best estimate determined by the initial search is returned. However, this estimate is guaranteed to have lower SER than the NC estimate.

Test Case 3: In this example, we examine the efficiency of our hybrid algorithm when $M = N = 16$. The receive and transmit SNRs considered are listed in Table V. We can observe from the P_{in} values that the stopping rule becomes less effective with decreasing SNR. Nevertheless, some complexity reduction is achieved with controllable error performance loss.

VII. CONCLUSIONS

In this paper, the conventional SDA was considered under a random channel model commonly encountered in communications. It was observed that the depth-first search strategy of SDA determines the order in which candidates of $\mathbf{s}$ are examined, which is opposite to what the distributions of $r_{i,i}$ suggest. To address this problem, we combined the SDA with optimal detection ordering. With both theoretical analysis and simulations, we asserted that detection ordering offers a simple, but effective means to improve the search efficiency.

A novel probabilistic search algorithm was derived based on the SDA, whose complexity is bounded by the actual minimum Euclidean distance instead of the distance of the NC solution or bounds determined from the random noise model. This algorithm breaks the search tree structure of the SDA, and examines the candidates approximately in a descending probability order. A complexity analysis was also provided. This algorithm is computationally efficient for the relatively high SNR regime, where current CDMA systems operate. To further reduce complexity of our probabilistic search, a hybrid algorithm with bounded performance degradation was developed based on a fast stopping criterion. Simulations confirmed considerable complexity reduction of the hybrid algorithm as the SNR increases.

REFERENCES

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. on Inform. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[2] L. Babai, "On Lovasz' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[3] L. Brunel and J. Boutros, "Euclidean space lattice decoding for joint detection in CDMA systems," in *Proc. of Inform. Theory Workshop*, Kruger National Park, South Africa, June 1999.

[4] N. Balakrishnan and A. C. Cohen, *Order Statistics and Inference Estimation Methods*, Academic Press, Inc., 1991.

[5] A. Chan and I. Lee, "A new reduced-complexity sphere decoder for multiple antenna systems," *Proc. of Intl. Conf. on Communications*, vol. 1, pp. 460-464, New York City, N.Y., April 28-May 2, 2002.

[6] H. Cohen, *A Course in Computational Algebraic Number Theory*, Berlin, Germany: Springer-Verlag, 1993.

[7] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups, 3rd Ed.*, Springer-Verlag New York, Inc., 1999.

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

[9] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multielement antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–49, Autumn 1996.

[10] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*, Chapman & Hall/CRC, 2000.

[11] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," *Proc. of 35th Asilomar Conf. on Signals, Systems, and Computers*, pp. 1051–1055, Pacific Grove, CA, Nov. 2001.

[12] R. Kannan, "Improved algorithms for integer programming and related lattice problems," *Proc. of 15th Annual ACM Symp. on Theory of Computing*, pp. 193–206, Boston, MA, Apr. 1983.

[13] A. K. Lenstra, H. W. Lenstra, and L. Lovasz, "Factoring polynomials with rational coefficients," *Math. Annalen*, vol. 261, pp. 513–534, 1982.

[14] J. Luo, K. Pattipati, P. Willett, and L. Brunel, "Branch-and-bound-based fast optimal algorithm for multiuser detection in synchronous CDMA," *Proc. of Intl. Conf. on Communications*, Anchorage, Alaska, May 11-15, 2003.

[15] X. Ma, and G. B. Giannakis, "Full-Diversity Full-Rate Complex-Field Space-Time Coding," *IEEE Transactions on Signal Processing*, Nov. 2003 (to appear).

[16] R. J. Muirhead *Aspects of Multivariate Statistical Theory*, John Wiley & Sons, New York, 1982.

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

[18] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," *Proc. ISSSE*, Pisa, Italy, Sep. 1998.

[19] Y. Xin, Z. Wang, and G. B. Giannakis, "Space-Time Diversity Systems based on Linear Constellation Precoding," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 294-309, March 2003.

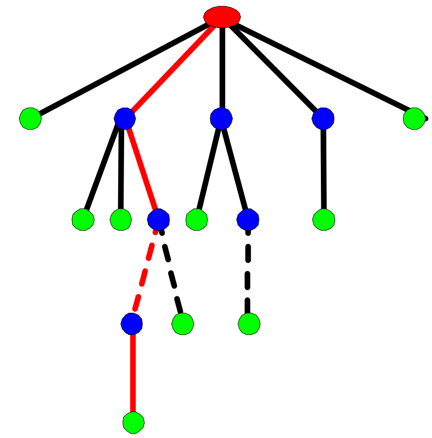

Fig. 1. The search tree of the SDA.

TABLE I

COMPARISON OF EXPECTATIONS FOR THE DIAGONAL ELEMENTS OF $\mathbf{R}$ WHEN $M = N = 16$

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$E[r_{i,i}^2]$	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
$E[r_{o,i,i}^2]$	9.8	10.5	10.6	10.5	10.2	9.8	9.2	8.7	8.0	7.3	6.5	5.6	4.7	3.7	2.7	1.5

TABLE II

RUNNING TIME REDUCTION BY OPTIMAL DETECTION ORDERING WHEN $M = N = 16$

SNR	16	17	18	19	20
Channel	10^3	$5 \cdot 10^3$	10^4	10^5	10^6
$t_{w/o}/t_o$	1.34	1.42	1.48	1.56	1.65

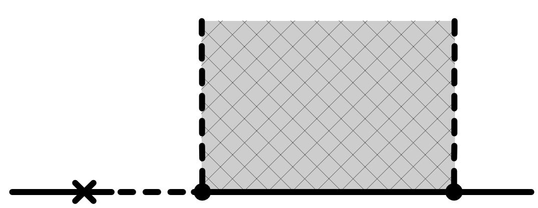

Fig. 2. Approximate probability of u_i given $\hat{s}_i$

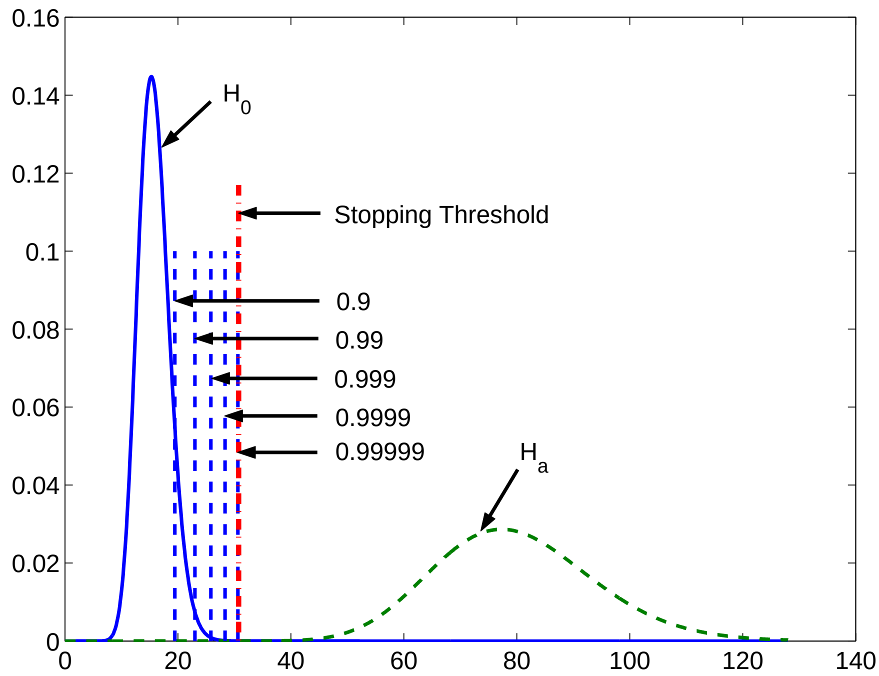

Fig. 3. Hypothesis Testing for $M = N = 32$ at SNR $= 24$ dB

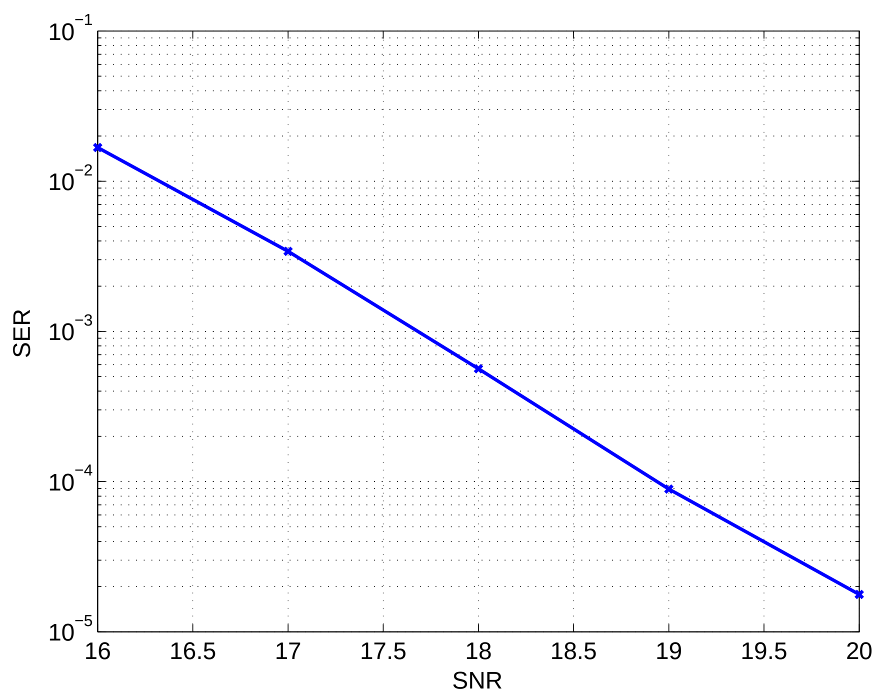

Fig. 4. The SER Performance for $M = N = 16$

TABLE III

FAST STOPPING THRESHOLD VS. DIMENSION ($M = N$) AT RECEIVE SNR = 24 dB

M	4	8	12	16	20	24	28	32
Transmit SNR	14.97	11.96	10.20	8.95	7.98	7.19	6.52	5.94
P_M ($\times 10^{-6}$)	149.9	2.5	2.1	2.1	2.2	2.1	2.8	2.3
V_{th}	0.5315	1.746	4.561	8.396	13.059	18.317	24.607	30.842
P_{in}	0.972184	0.970879	0.998254	0.99979	0.999952	0.999979	0.999992	0.999992

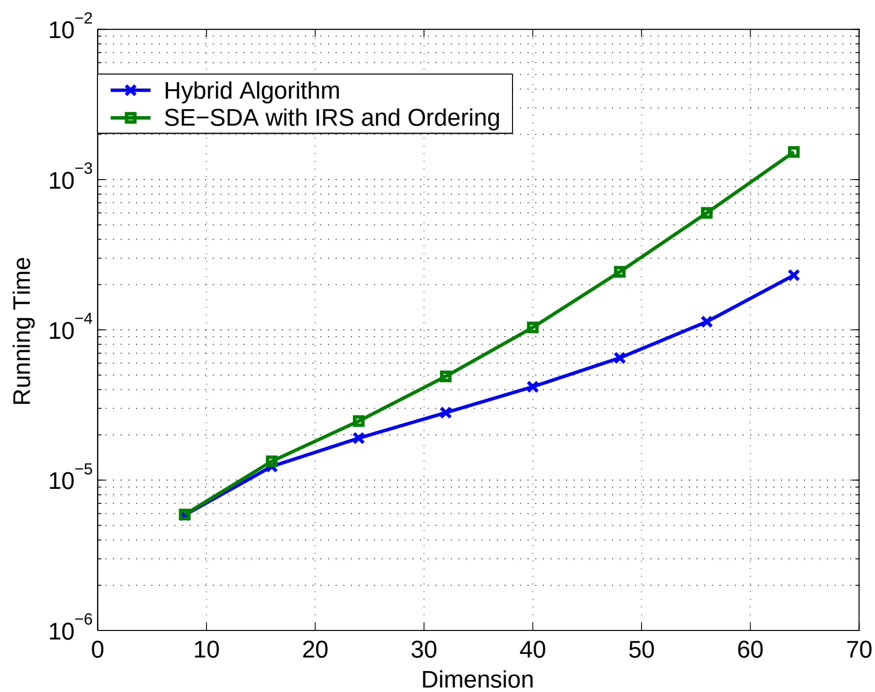

Fig. 5. Running Time Comparison at SNR $= 24$

TABLE IV

COMPARISON BETWEEN THE HYBRID ALGORITHM AND THE SE-SDA IN TERMS OF BOTH COMPLEXITY AND SER AT RECEIVE SNR = 24 dB

M	4	8	12	16	20	24	28	32
t_{SE}/t_{HA}	1.0092	1.0812	1.2968	1.7420	2.4860	3.7414	5.3062	6.5950
SE ($\times 10^{-6}$)	164.0	2.6	2.2	2.1	2.2	1.8	1.8	3.1
HA ($\times 10^{-6}$)	244.3	2.5	1.7	1.5	1.9	1.3	1.4	2.1

TABLE V

COMPARISON OF THE HYBRID ALGORITHM WITH THE SE-SDA WHEN $M = N = 16$

SNR$_R$	18	20	22	24
SNR$_T$	2.95	4.95	6.95	8.95
P_M	$5.8 \cdot 10^{-4}$	$2.0 \cdot 10^{-5}$	$3.7 \cdot 10^{-6}$	$2.1 \cdot 10^{-6}$
V_{th}	18.10035	11.114	9.341	8.396
P_{in}	0.745196	0.707481	0.963778	0.99979
t_{SE}/t_{HA}	1.2482	1.1914	1.9135	1.7420
SE	$5.75 \cdot 10^{-4}$	$1.81 \cdot 10^{-5}$	$3.22 \cdot 10^{-6}$	$2.1 \cdot 10^{-6}$
HA	$6.8 \cdot 10^{-4}$	$1.90 \cdot 10^{-5}$	$3.14 \cdot 10^{-6}$	$1.5 \cdot 10^{-6}$