A Low Complexity SCMA Decoder Based on List Sphere Decoding

Fan Wei and Wen Chen

Department of Electronic Engineering, Shanghai Jiao Tong University, China
Email: {weifan89; wenchen}@sjtu.edu.cn

Abstract—Non-orthogonal multiple access is one of the key techniques developed for the future 5G communication systems among which, the recent proposed sparse code multiple access (SCMA) has attracted a lot of researchers’ interests. By exploring the shaping gain of the multi-dimensional complex codewords, SCMA is shown to have a better performance compared with some other non-orthogonal schemes such as low density signature (LDS). However, although the sparsity of the codewords makes the near optimal message passing algorithm (MPA) possible, the decoding complexity is still very high. In this paper, we propose a low complexity decoding algorithm based on list sphere decoding. Complexity analysis and simulation results show that the proposed algorithm can reduce the computational complexity substantially while achieve the near maximum likelihood (ML) performance.

Index Terms—5G, Non-orthogonal multiple access, SCMA, message passing algorithm, list sphere decoder.

I. INTRODUCTION

The non-orthogonal multiple access is regarded as an essential technique to cope with the massive connectivity requirement in the next generation wireless network. As a candidate technique for non-orthogonal multiple access, sparse code multiple access (SCMA) [1], [2] is shown to have a superior performance due to the shaping gain of the multi-dimensional complex codewords. By multiplexing the $K$ layers complex sparse codewords over $N$ orthogonal resources with $K > N$, non-orthogonal multiple access is achieved in SCMA.

SCMA can be viewed as an improved version of low density signature (LDS) [3]. In LDS, QAM symbols are repeated over several chips. To reduce the multiuser interference, the chips are sparse so that only a few users are collided on each chip. Benefited from this sparsity feature, both SCMA and LDS can use message passing algorithm (MPA) to detect the receive signals [4]. While the sparsity of the codewords or chips have reduced the decoding complexity effectively, MPA still needs exponential times in decoding. Thus, for some large constellation sizes or highly overloaded systems, the complexity is time consuming.

In [5], log domain MPA is introduced to reduce the decoding complexity of SCMA. It is shown that by this transformation, one can convert considerable number of multiplications into summations. Further, partial marginalized (PM) based MPA is proposed in [6]. In this scheme, after some iterations in MPA, part of the symbols are chosen by the receiver and judged in advance. Then those symbols are never computed in the left iterations. Although the decoding complexity is reduced, the above methods are still of high complexity.

In this paper, motivated by the lattice structure of SCMA codewords, we propose a low complexity decoding algorithm based on list sphere decoding (LSD) algorithm [7], [9]. In general, sphere decoder avoids the exhaustive search for all possible hypotheses, instead it considers only the signal points within a hypersphere. Thus, the log likelihood ratio (LLR) can be approximated by using the candidate list searched by LSD. Simulation shows that a small enough candidate set can achieve near ML performance and thus reduce the computational complexity in MPA detection.

In this paper, $\mathbb{B}$ and $\mathbb{C}$ are used to denote the binary and complex numbers. Lowercase letters $x$, bold lowercase letters $\mathbf{x}$ and bold uppercase letters $\mathbf{X}$ denote scalars, column vectors and matrices, respectively. $x_i$ representing the $i$-th component of vector $\mathbf{x}$ while $x_{ij}$ representing the $(i, j)$-th component of matrix $\mathbf{X}$. We use $(\cdot)^*$, $(\cdot)^T$ and $(\cdot)^\dagger$ to denote complex conjugate, matrix transpose and conjugate matrix transpose. $\text{diag}(\mathbf{x})$ is the diagonal matrix with the diagonal entries being vector $\mathbf{x}$. $\xi \backslash k$ means the set $\xi$ with element $k$ being excluded.

II. SYSTEM MODEL

Fig. 1 shows an uplink multiple access SCMA system. Typically, an uplink SCMA system consists of $K$ layers which map the encoded bits to an $N$-dimensional complex lattice signal point. In SCMA, one or more layers can be allocated to each user and in this paper, we assume each user occupy only one SCMA layer. To reduce the multiuser interference, the $N$-dimensional complex codewords are sparse such that only $P < N$ dimensions are used to convey data while the left $N - P$ dimensions are set to zeros. The data in each dimension are modulated to an OFDMA subcarrier (corresponds to the resource node in Fig. 1) and transmitted through the wireless channels.

A. Factor Graph Representation

As $K$ layers are multiplexing over $N$ OFDMA subcarriers, the SCMA system can be represented by a Forney factor graph. Fig. 2 illustrates a factor graph with $K = 6$, $N = 4$ and $P = 2$. For an equivalent representation of this structure, we can use an indicator matrix

$$
\mathbf{F} = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1
\end{bmatrix},
$$

978-1-5090-1328-9/16/$31.00 ©2016 IEEE
B. SCMA Codewords

The SCMA encoder can be defined as a mapping of binary encoded bits to a multidimensional complex signal point, i.e., \( f_k : V_k g \) with \( g : \mathbb{B}^{\log_2 M} \rightarrow C, C \subset \mathbb{C}^P \) where \( M \) is the size of the constellation point and \( V_k \in \mathbb{B}^{N \times P} \) is the mapping matrix that converts the \( P \)-dimensional non-zero codewords to an \( N \)-dimensional sparse codewords. In general, \( V_k \) can be obtained by inserting \( N - P \) zero rows to an identity matrix \( I_P \).

Since there are \( K = C(N, P) \) different mapping matrices, we can have at most \( K \) layers in an SCMA system with a given parameter \( N \) and \( P \).

For the design of \( P \)-dimensional non-zero codewords \( x_P \) for each layer, [1], [2] has introduced a multi-step suboptimal method which can be formulated as

\[
G^+, |\Delta_k^+|_{k=1}^K = \arg \max_{G, \Delta_k} m(S(V^+, G) = [(\Delta_k)G_{k=1}^K; K, M, P, N]),
\]

where \( G^+ \) is the mother constellation and \( |\Delta_k^+|_{k=1}^K \) is the constellation operators for \( K \) layers. Therefore, we can design one mother constellation for all and \( K \) constellation operators separately for each layer.

Typically, the \( P \)-dimensional mother constellation can be formed by the Cartesian product of \( P \) orthogonal QAM symbols. In addition, to increase the product distance of the constellation and create the power variation in different dimension, the unitary rotation would be applied to the mother constellation.

Let \( u_{2P} = (u_1, u_2, ..., u_{2P})^T \) be the equivalent \( 2P \)-dimensional QAM constellation, where \( u_i = \pm 1, \pm 3, ..., M_{2P \times 2P} \) be the unitary rotation matrix. Then the \( P \)-dimensional non-zero codewords can be written as,

\[
x_P = (E_r + i \cdot E_i) \cdot M \cdot u_{2P},
\]

where \( E_r \) and \( E_i \) are \( P \times 2P \) matrix that select components from \( u_{2P} \) corresponding to the real part and imaginary part of the QAM symbols, respectively. Detailed construction of rotation matrix can be found in [10], [11].

After that, \( K \) different constellation operators are applied to the above mother constellation to construct the codebook for \( K \) layers. Three different operators were introduced in [1], [2], i.e.,

1) Phase rotation: \( \theta_k = (k - 1) \frac{2\pi}{M_{2P}}, \forall k = 1, ..., d_c; \)
2) Dimensional permutation;
3) Complex conjugate.

Denote the operator for layer \( k \) as \( \Delta_k \). Then we can write the \( P \)-dimensional codewords for layer \( k \) as,

\[
x_{P,k} = \Delta_k \cdot (E_r + i \cdot E_i) \cdot M \cdot u_{2P,k}
\]

where \( G_k = \Delta_k \cdot (E_r + i \cdot E_i) \cdot M \). Indeed, the SCMA codewords are essentially multi-dimensional complex lattice constellation signal points.

At the receiver, the received signal can be written as,

\[
y = \sum_{k=1}^{K} \text{diag}(h_k)x_k + z,
\]

where \( x_k \) and \( h_k \) are the SCMA codewords and channel coefficient for layer \( k \), respectively, and \( z \) is the additive complex Gaussian white noise with distribution \( \mathcal{CN}(0, \sigma^2 \mathbf{I}) \).

On the \( n \)th subcarrier, the received signal can be written as,

\[
y_n = \sum_{k \in \xi_n} h_{n,k}x_{n,k} + z
\]

\[
= h_n^T x_n + z
\]

\[
= h_n^T \text{diag}(g_{n,\kappa(1)}, ..., g_{n,\kappa(d_c)}) u_n + z
\]

\[
= H_n u_n + z,
\]
where \( \mathbf{x}_n = [x_{n,k(1)}, \ldots, x_{n,k(d_e)}]^T \) is the set of users collided on the \( n \)th subcarrier, \( \mathbf{h}_n = [h_{n,k(1)}, \ldots, h_{n,k(d_e)}]^T \) is the channel vector in subcarrier \( n \). Since \( \mathbf{x}_{P,k} = \mathbf{G}_k \mathbf{u}_{2P,k} \), the \( n \)th component of \( \mathbf{x}_{P,k} \) can be written as \( x_{n,k} = g_{n,k}^T \mathbf{u}_{2P,k} \), where the row vector \( g_{n,k}^T \) corresponds to the \( n \)th row of matrix \( \mathbf{G}_k \). In this case, we can formulate \( \mathbf{x}_n = \text{diag}(g_{n,k(1)}^T, \ldots, g_{n,k(d_e)}^T) \mathbf{u}_n \) with \( \mathbf{u}_n = [u_{2P,n(1)}, \ldots, u_{2P,n(d_e)}]^T \) being the set of lattice points of collision users in subcarrier \( n \).

### III. Decoding of SCMA Codewords

#### A. MPA Detection

Inspired by LDPC codes, the SCMA codewords can also be decoded using message passing algorithm (MPA). During the decoding, soft information of the codewords are exchanged between layer nodes (LNs) and resource nodes (RNs) in a iterative way until a maximum number of iteration is reached.

In log domain, the LLR information sent from resource node \( n \) to layer node \( k \) can be written as,

\[
I_{g_n \rightarrow v_k}(x_k) = \max_{x_k \in \xi_k \setminus k} \{ -f_n(x) + \sum_{u \in \xi_k \setminus k} I_{v_u \rightarrow g_n}(x_u) \},
\]

where \( f_n(x) = \frac{1}{2\pi} \| y_n - \sum_{k \in \xi_n} h_{n,k} x_{k,n} \|^2 \). The \( \max \) operation is given by

\[
\max(a, b) = \log(e^a + e^b) = \max(a, b) + \log(1 + e^{-|a-b|}).
\]

For Max-log-MPA, the approximation \( \max(a, b) \approx \max(a, b) \) is applied.

The information sent from layer node \( k \) to resource node \( n \) can be written as,

\[
I_{v_k \rightarrow g_n}(x_k) = L(x_k) + \sum_{l \in \xi_k \setminus n} I_{g_l \rightarrow v_k}(x_k),
\]

where \( L(x) = \log p(x) \) is a priori probability of \( x \).

When the algorithm is converged or a maximum number of iterations \( IT \) is reached, a posteriori probability of codeword \( x_k \) is given by

\[
I(x_k) = L(x_k) + \sum_{l \in \xi_k} I_{g_l \rightarrow v_k}(x_k).
\]

Clearly, the operations dominating the process of decoding is equation (6), where the marginalization \( \max \) need to computed \( M^{d_e-1} \) items. When the constellation size is \( M \), the total complexity is \( M^{d_e} \). However, it can be observed that the massive computation is useless since the \( f_n(x) \) with large Euclidean distance has tiny contributions to (6). In the next subsection, we will introduced a new method based on list sphere decoding (LSD), which avoids the exhaustive search for all the possible hypotheses and only considers the signals within a given hypersphere.

#### B. LSD-based MPA Detection

Let us consider the following maximum likelihood detection,

\[
\hat{u}_n = \arg \min_{u_n \in \Lambda} \| y_n - \mathbf{H}_n u_n \|^2,
\]

where \( y_n \in \mathbb{C} \) is the receive signal in subcarrier \( n \), \( \mathbf{H}_n \) and \( u_n \) are row vector and column vector defined in (5), respectively. The ML detector searches the optimal \( \hat{u}_n \) in a brute-force way, which is clearly inefficient especially in high SNR, when the possible transmit signal is not far from the receive value \( y_n \).

In sphere decoder, we avoid this exhaustive search and only consider signals within a given hypersphere.

Let the dimension of \( \mathbf{H}_n \) and \( u_n \) be \( L \) and assume that the entries of \( u_n \) are of constant modulus (e.g., BPSK modulation). Equation (10) can be formulated as [12],

\[
\hat{u}_n = \arg \min_{u_n \in \Lambda} \| y_n - \mathbf{H}_n u_n \|^2 + \alpha \| u_n \|^2,
\]

where \( \alpha > 0 \), \( \mathbf{y}_n = [y_n] \) and \( \mathbf{H}_n = \begin{pmatrix} \mathbf{H}_n & \alpha \mathbf{I} \end{pmatrix} \) are \((L + 1) \times 1\) and \((L + 1) \times L\) matrices, respectively.

Now consider the following generalized sphere decoder,

\[
\| \mathbf{y}_n - \tilde{\mathbf{H}}_n u_n \|^2 \leq C,
\]

where \( C \) is the search radius. Applying the QR factorization of \( \mathbf{H}_n \), we have,

\[
\tilde{\mathbf{H}}_n = [Q_1, Q_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix},
\]

where \( Q_1, Q_2 \) and \( \mathbf{R} \) are \((L + 1) \times L\), \((L + 1) \times 1\) and \( L \times L\) matrices, respectively. Since the multiplication of a unitary matrix to a vector does not change the norm of the vector, we have,

\[
\| \mathbf{y}_n - \tilde{\mathbf{H}}_n u_n \|^2 \leq \| \mathbf{y}_n' - \mathbf{R} u_n \|^2 \leq C',
\]

where \( y_n' = Q_1 \mathbf{y}_n \) and \( C' = C - \| Q_2 \mathbf{y}_n \|^2 \). For notation simplification, we will drop the prime in the following.

Note that matrix \( \mathbf{R} \) is an upper triangle matrix. Hence we can expand (14) as,

\[
\sum_{i=1}^{L} |y_i - \sum_{j=i}^{L} r_{ij} u_j|^2 \leq C.
\]

The sphere decoder works in a backward recursive way. For \( i = l \), i.e., considering the last \( L - l + 1 \) terms, we have,

\[
|y_l - \sum_{j=l+1}^{L} r_{lj} u_j|^2 + \sum_{i=l+1}^{L} |y_i - \sum_{j=i}^{L} r_{ij} u_j|^2 \leq C.
\]

Define \( \hat{y}_l = y_l - \sum_{j=l+1}^{L} r_{lj} u_j \) and \( T_l = C - \sum_{i=l+1}^{L} |\hat{y}_i - r_{il} u_i| \), (16) can be further simplified as,

\[
|\hat{y}_l - r_{il} u_i|^2 \leq T_l.
\]
The upper bound $UB(u_i)$ and lower bound $LB(u_i)$ can be obtained by solving the above inequality such that,

$$UB(u_i) = \frac{|\tilde{y}_i| \cos(\tilde{\theta}_i) + \sqrt{T_i - |\tilde{y}_i|^2 \sin^2(\tilde{\theta}_i)}}{r_{HL}}, \quad (18)$$

and

$$LB(u_i) = \frac{|\tilde{y}_i| \cos(\tilde{\theta}_i) - \sqrt{T_i - |\tilde{y}_i|^2 \sin^2(\tilde{\theta}_i)}}{r_{HL}}, \quad (19)$$

where the complex $\tilde{y}_i = |\tilde{y}_i| \exp(j\tilde{\theta}_i)$.

In general, we can search the signal points by a lexicographic order, i.e., let $u_i = \lfloor LB(u_i) \rfloor - 1$ and search in an ascending order until it reaches $\lceil UB(u_i) \rceil$, which is the original version of sphere decoder [7]. Alternatively, the search can also be proceeded in an ascending order of distance increment (DI) $e(u_i) = |\tilde{y}_i - r_i u_i|^2$, i.e., search in an order

$$u_{i,1}, u_{i,2}, u_{i,3}, \ldots,$$  

such that $e(u_{i,i}) \leq e(u_{i,j})$ for $i < j$. This is referred as Schnorr-Euchner enumeration (SEE) [8]. The SEE starts with the point that minimize the partial Euclidean distance (PED) $d(u_{i}^T) = \sum_{i=1}^{L} |y_i - \sum_{j=1}^{L} r_{ij} u_j|^2$, so that we can find the right path earlier than the original sphere decoder. Further, in SEE the search radius can be initialized as $C = +\infty$, i.e., we are free of the choice of initial radius.

In the above, $u_n$ is assumed to be constant modulus. For general QAM symbols, $u_i = \pm 1, \pm 3, \ldots, \text{to make } (11)\text{ feasible, a simple decomposition can be applied},

$$u = \sum_{i=0}^{\log M - 1} 2^i \cdot u'_i \gamma^T \cdot u', \quad (21)$$

where $u = \pm 1, \pm 3, \ldots$ is an M-PAM symbol and $u'_i$ is a BPSK modulation, and vector $\gamma^T = [2^{\log M - 1}, \ldots, 2, 1]$. In this way, the $u_{2P,k}$ in (3) can rewritten as,

$$u_{2P,k} = \begin{bmatrix} \gamma_{1,k} \\ \gamma_{2,k} \\ \vdots \\ \gamma_{2P,k} \end{bmatrix} \begin{bmatrix} u_k^1 \\ u_k^2 \\ \vdots \\ u_k^{2P} \end{bmatrix} = \Gamma_k \cdot u'_{2P,k}, \quad (22)$$

where the entries in $u'_{2P,k}$ are BPSK. Further, the collision signals in the $n$ subcarrier can be written as,

$$u_n = \begin{bmatrix} u_{2P,n(1)} \\ u_{2P,n(2)} \\ \vdots \\ u_{2P,n(d_c)} \end{bmatrix} = \begin{bmatrix} \Gamma_{n(1)} \\ \Gamma_{n(2)} \\ \vdots \\ \Gamma_{n(d_c)} \end{bmatrix} \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ \vdots \\ u_{i,2P} \end{bmatrix}, \quad (23)$$

Algorithm 1 LSD based MPA Detection

**Initialization:** $Q = [Q_1, Q_2], R, y_n = Q_1\tilde{y}_n, C = +\infty, IT$

**Iteration:** (LSD Iteration)

1: Set $i = L, y_L = y_{L}, T_L = 0$
2: (SE Enumeration) Set $u_i = \text{sign}(|\tilde{y}_i| \cos(\tilde{\theta}_i))$ and search step $\Delta_i = -2 \cdot \text{sign}(|\tilde{y}_i| \cos(\tilde{\theta}_i))$.
3: (Node Pruning) If $T_i + |\tilde{y}_i - r_i u_i|^2 > C$ or $u_i \notin \{-1, 1\}$, go to 4. else go to 5.
4: If $i = L$, terminate and output $\Phi_n$; else $i = i + 1, u_i = u_i + \Delta_i$, go to 3.
5: (PED Computation) If $i = 1$, go to 6; else set $T_{i-1} = T_i + |\tilde{y}_i - r_i u_i|^2, y_i = y_{i-1} - \sum_{j=1}^{L} r_{i-1,j} u_{j}$ and $i = i - 1$, go to 2.
6: A lattice point $u_n$ within the hypersphere has been found. Let $\|y_n - Ru_n\|^2 = T_1 + |y_1 - r_{11} u_1|^2$. Add $u_n$ and the corresponding Euclidean distance $d(u_n)$ to the candidate list set. If the candidate list set is full (the size reaches $T_{max}$), find $u_n$, with maximum $d(u_n)$, set $C = d(u_n), u_i = u_i + \Delta_i$, go to 3.

**Iteration:** (MPA Iteration)

7: Set $i = 1$
8: while $i \leq IT$ do
9: (LN Updating) Compute $I_{g_n \rightarrow v_n}(x_k)$ using (25).
10: (RN Updating) Compute $I_{u_n \rightarrow g_n}(x_k)$ using (8).
11: $i = i + 1$.
12: end while
13: (LLR of $x_k$) Compute $I(x_k)$ using (9).

Using (23), we can reformulated (5) as,

$$y_n = \sum_{k \in \xi_n} h_n,k x_{n,k} + z$$

$$= h_n^T \text{diag}(g_{n,(1)}, \ldots, g_{n,(d_c)}) u_n + z$$

$$= h_n^T \text{diag}(g_{n,(1)}(\Gamma_{n(1)}), \ldots, g_{n,(d_c)}(\Gamma_{n(d_c)})) u'_n + z$$

$$= H_n u_n + z, \quad (24)$$

where vector $H_n = h_n^T \text{diag}(g_{n,(1)}(\Gamma_{n(1)}), \ldots, g_{n,(d_c)}(\Gamma_{n(d_c)}))$ and $u_n = [u_{i,1}^{T}, \ldots, u_{i,2P}^{T}].$ Since the entries in $u_n$ are BPSK, we can proceed the sphere decoding as in the constant modulus case.

In general, the sphere decoder finds all the candidate lattice points within a given radius $C$. Denote the set of candidate lattice points as $\Phi_n = \{u_{i,1}^{n}, \ldots, u_{i,2P}^{n,\max}\}$ where $T_{max}$ is the size of this set. In LSD we approximate the equation (6) as [9],

$$I_{g_n \rightarrow v_n}(x_k) = x_k^{a_n}$$

$$\approx \max_{x \in \Phi_n \cap X_{k,n}} \{-f_n(x) + \sum_{u \in \xi_n} I_{v_u \rightarrow g_n}(x_u)\}, \quad (25)$$

where $X_{k,n}^m$ is the collision codewords set on the $n$th subcarrier with $x_k$ being the $n$th constellation point.

Obviously, the equality holds only when $T_{max} = M^{d_c}$ and there is a trade off between the computational complexity.
of (25) and the performance of MPA. However, we will observe in the simulation section that a small enough $T_{\text{max}}$ would usually result in an acceptable performance loss. Hence the complexity to compute (25) is reduced substantially. The LSD based MPA is summarised in Algorithm 1.

C. Complexity Analysis

In this subsection, the complexity of original MPA and LSD-based MPA is analyzed. We assume that the Max-log-MPA is used in the detection and the complexity is evaluated in terms of the number of floating point operations per OFDMA subcarrier in this paper. A flop is regarded as either a complex multiplication or a complex summation.

First consider the LSD-based MPA. For initialization, the LSD needs a QR factorization of matrix $\tilde{H}_n$ and the computation of the column vector $\tilde{y}_n = Q_1^T \tilde{y}_n$. Note that the dimensions of $\tilde{H}_n$ and $\tilde{y}_n$ are $(L+1) \times L$ and $L \times 1$. Since we have the decomposition (23), $L = d_c \cdot \log_2 M$. The QR factorization is implemented based on modified Gram-Schmidt algorithm (MGS) [15], which needs $2(L+1)L^2$ flops. To compute $y_n = Q_1^T \tilde{y}_n$, $L$ multiplications are needed since $\tilde{y}_n$ is a sparse column vector. Therefore, $2L^3 + 2L^2 + L$ flops are required in total before running the LSD.

For LSD, the expected complexity as shown in (26) is evaluated in this paper [13],

$$E_c = \sum_{k=1}^{L} f(k)N_k = \sum_{k=1}^{L} (2k + 7)N_k,$$

where $N_k$ is the averaged number of points inside a $k$-dimensional sphere and $f_k$ is the number of flops needed for searching a point in dimension $k$ (step 2 to step 6).

For the complexity of Max-log-MPA, $(2d_c^2 - d_c)M^{d_c}$ summations and $(d_c^2 + 3d_c)M^{d_c}$ multiplications are required per OFDMA subcarrier to compute equation (6) while for LSD-MPA, since $f_n(x)$ has been computed in Algorithm 1, only $(d_c^2 - d_c)T_{\text{max}}$ summations are needed. Finally, the computations of (8) and (9) need $d_c(P-1)M = (\frac{N}{N}d_c^2 - d_c)M$ summations and $\frac{KPM}{N} = d_cM$ summations, respectively. The computational complexities of the two algorithms are summarised in Table I, where $IT$ is the number of MPA iteration.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$I_{g_n \rightarrow v_k}(x_k)$</th>
<th>$I_{v_k \rightarrow g_n}(x_k)$</th>
<th>$I(x_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere Decoding</td>
<td>$2L^3 + 2L^2 + L + \sum_{k=1}^{L} (2k + 7)N_k$</td>
<td>$IT \cdot (3d_c^2 + 2d_c) \cdot M^{d_c}$</td>
<td>$IT \cdot (d_c^2 - d_c) \cdot M$</td>
</tr>
<tr>
<td>LSD-MPA</td>
<td>$IC \cdot (d_c^2 - d_c) \cdot T_{\text{max}}$</td>
<td>$IC \cdot (\frac{N}{N}d_c^2 - d_c) \cdot M$</td>
<td>$d_cM$</td>
</tr>
</tbody>
</table>

IV. Simulation Results

In this section, we investigate the BER performance and the computational complexity of the proposed LSD-MPA in uplink SCMA system. 16 points SCMA with $N = 4$, $P = 2$ and indicator matrix $F$ introduced in Section II is considered in this paper. The channel is additive white Gaussian noise (AWGN) channel.

In Fig. 3, the BER performance of Max-log-MPA, LSD-MPA and PM-MPA is compared, where $R_s$ is used to determine the number of symbols to be marginalized in the rest of $IT - m$ iterations as in PM-MPA. For channel coding, half rate turbo code is utilized. The maximum iteration number is set to 10 for both MPA and turbo code. From Fig. 3, we can observe that the proposed LSD-MPA has attained the near ML performance. In fact, the raw bits curves for Max-log-MPA and LSD-MPA are overlapped with each other and for encoded BER, the proposed scheme only results in 0.2dB performance.
loss. For comparison, we can easily observe that there is a large performance gap between PM-MPA and Max-log-MPA especially in high SNR region.

The amazing fact reflected in Fig. 3 is that the near ML performance can be achieved through the candidate list set $\Phi_n$ with a small enough size. In simulation, we set $T_{\text{max}} = 96$, which is a small portion of the whole searching space ($96/16^3 \approx 0.023$). Indeed, the brute force search for all possible hypotheses is rather inefficient since the large Euclidean distance has tiny contribution to (6). Thus, by limiting the searching space within a hypersphere, we are able to reduce the computational complexity of (6) effectively.

The complexity of sphere decoder depends mainly on the expected visited points $N_k$ in each $k$-dimensional hypersphere [14]. In this paper, we count the averaged number of visited nodes through Monte-Carlo simulation with 10000 noise samples (i.e., for 10000 receive $y^k_n$). In Table II, we list the total averaged number of visited nodes $\sum_{k=1}^{L} N_k$ during the search in sphere decoder.

<table>
<thead>
<tr>
<th>EbN0 (dB)</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_{k=1}^{L} N_k$</td>
<td>1908</td>
<td>1908</td>
<td>1898</td>
<td>1905</td>
<td>1914</td>
<td>1907</td>
</tr>
</tbody>
</table>

The histogram in Fig. 4 compares the number of operations for additions and multiplications, where the SNR is set to 10 dB. From the figure, we can observe that by utilizing the proposed LSD-MPA, the numbers of summations and multiplications has been reduced by about one order of magnitude compared with the traditional MPA. Note that although some square root operations are required by the MGS QR factorization, they are far less than the other operations. Therefore the decoding complexity can still be reduced by using the proposed algorithm.

V. Conclusion

In this work, we propose a new decoding algorithm based on LSD, which aims to reduce the computational complexity of the original MPA detection. By clarifying the lattice structure of SCMA codewords, we had shown that SCMA can be decoded by using the low complexity list sphere decoder. Instead of the exhaustive search for all possible hypotheses, the proposed LSD-MPA only considers the signals within a hypersphere. Simulation results shown that a candidate list set with a small enough size searched by LSD can achieve the near ML performance. Therefore the proposed algorithm has attained a well trade off between the BER performance and computational complexity.

ACKNOWLEDGMENT

This paper is supported by NSFC #6167101252, by the National 973 Project #2012CB316106, by the National 863 Project #2015AA01A710, by Guangxi NSF #2015GXNSFFA139037, and by SKL on Mobile Communications #2013D11.

REFERENCES


