Sphere Constraint based Enumeration Methods to Analyze the Minimum Weight Distribution of Polar Codes

Jinnan Piao, Student Member, IEEE, Kai Niu, Member, IEEE, Jincheng Dai, Member, IEEE, and Chao Dong, Member, IEEE

Abstract—In this paper, the minimum weight distributions (MWDs) of polar codes and concatenated polar codes are exactly enumerated according to the distance property of codewords. We first propose a sphere constraint based enumeration method (SCEM) to analyze the MWD of polar codes with moderate complexity. The SCEM exploits the distance property that all the codewords with the identical Hamming weight are distributed on a spherical shell. Then, based on the SCEM and the Plotkin’s construction of polar codes, a sphere constraint based recursive enumeration method (SCREM) is proposed to recursively calculate the MWD with a lower complexity. Finally, we propose a parity-check SCEM (PC-SCEM) to analyze the MWD of concatenated polar codes by introducing the parity-check equations of outer codes. Moreover, due to the distance property of codewords, the proposed three methods can exactly enumerate all the codewords belonging to the MWD. The enumeration results show that the SCEM can enumerate the MWD of polar codes with code length up to $2^{14}$ and the PC-SCEM can be used to optimize CRC-polar concatenated codes.

Index Terms—polar codes, concatenated polar codes, sphere constraint based enumeration method, distance spectrum, minimum weight distribution.

I. INTRODUCTION

POLAR codes have been proved to achieve the capacity by the successive cancellation (SC) decoding as the code length goes to infinity [1]. However, when the code length is small or medium, the performance is unsatisfying. Thus, successive cancellation list (SCL) decoding [2], [3] and successive cancellation stack decoding [4] are introduced to improve the performance of polar codes. Furthermore, the performance is improved by the CRC-aided SCL (CA-SCL) decoding [5] which introduces the CRC detector into the SCL decoding. Thanks to its excellent performance, polar codes have been adopted as the coding scheme for the control channel of the enhanced Mobile Broadband (eMBB) service category in the fifth generation wireless communication systems (5G) [6], [7].

The weight distribution of codewords is the distance spectrum of polar codes, which can be used to evaluate the maximum-likelihood (ML) performance [8]. However, enumerating the distance spectrum has exponential complexity and it is almost impossible for long code length. In the high signal-to-noise ratio (SNR) region, the minimum weight distribution (MWD) is the main factor influencing the ML performance [9]. Thus, the ML performance can be evaluated by MWD instead of the distance spectrum.

To analyze the MWD of polar codes, Li et al. [8] propose an SCL method with excessively large list size to enumerate codewords and analyze MWD. However, due to the large consumption of memory and high complexity, implementing this method on a memory-constrained computer is difficult. Thus, the hard disk is used in [10] to reduce the number of survival paths and decrease the consumption of memory and a multi-level SCL method is proposed to reduce the list size in [11]. Nevertheless, these SCL based methods still have high complexities.

Besides, concatenated polar codes [12], especially CRC-polar concatenated codes [5], have better error performance than polar codes, since the distance spectrum of polar codes is improved by resorting to concatenated schemes. An uniform interleaver approach [13] is proposed to analyze the distance properties of concatenated polar code ensembles, but it cannot obtain the distance spectrum of CRC-polar concatenated codes with definite CRC polynomial. In addition, since this approach enumerates all the codewords, its complexity is extremely high.

In this paper, we exploit the distance property of codewords to exactly evaluate the MWDs of polar codes and concatenated polar codes. The distance property is that the codewords with the identical Hamming weight are distributed on a spherical shell. Hence, a sphere constraint with the minimum Hamming weight can early prune a large amount of unnecessary codewords for analyzing MWD. In addition, the sphere constraint ensures that all the codewords with the minimum weight could be enumerated exactly.

A sphere constraint based enumeration method (SCEM) is first proposed to analyze the MWD of polar codes with moderate complexity. The process of SCEM similar to that of the sphere decoding (SD) algorithm [14]–[17] is regarded as a depth-first tree search, which has negligible memory overhead compared with the SCL based methods. In the SCEM, the sphere constraint with the minimum Hamming weight is used to evaluate the MWD. Thus, the paths satisfying the sphere constraint in the search tree are reserved and the MWD is evaluated exactly. In comparison, the SCL based methods cannot guarantee to enumerate all the codewords with the minimum Hamming weight when the list size is small or medium. Then, although the paths violating the constraint are early pruned to reduce the redundant search, the complexity of SCEM is still too high to evaluate the MWD of long polar codes. Therefore, a sphere constraint based recursive enumeration method (SCREM) is proposed to analyze the MWD with lower complexity on the basis of the SCEM and
the Plotkin’s construction. Additionally, inspired by the CRC-aided SD (CA-SD) algorithm [17], a parity-check SCEM (PC-SCEM) is proposed to analyze the MWD of concatenated polar codes by introducing the parity-check equations of outer codes.

The main contributions of this paper are summarized as follows:

1) We first propose the SCEM to exactly enumerate all the codewords belonging to the MWD by exploits the distance property that all the codewords with the identical Hamming weight are distributed on a spherical shell.

2) The SCREM is proposed to recursively analyze the MWD of polar codes with lower complexity compared with the SCEM. In the SCREM, we first prove the property that the MWD of a polar code is related with the MWD of the two component polar codes in terms of the Plotkin’s construction. Based on the property, the MWD of the polar code is directly decided without search when the minimum Hamming weight of the two component codes is identical and the complexity of enumerating the MWD of the polar code can be efficiently reduced when the two component codes have different minimum Hamming weight.

3) The PC-SCEM is proposed to exactly analyze the MWD of concatenated polar codes by introducing the parity-check equations of outer codes. The parity-check equations are utilized to ensure that all the codewords enumerated are valid codewords. Then, to match the search order of PC-SCEM, Gaussian elimination is used to transform the parity-check equations into new forms. Due to the newly parity-check equations and the sphere constraint, all the codewords belonging to the MWD of concatenated polar codes are exactly enumerated.

The experimental results show that the proposed SCEM and SCREM with code length 128 have up to 10^4 and 10^8 times lower complexity compared with the SCL methods, respectively. The MWD analysis results show that the SCREM can enumerate the MWD of polar codes with code length up to 2^{14} and the PC-SCEM can analyze the MWD of CRC-polar concatenated codes to optimize the CRC polynomial.

The remainder of the paper is organized as follows. Section II describes the preliminaries of polar codes, SD algorithm and distance spectrum. In Section III, the distance property of codewords and the SCEM are described. The SCREM is provided to recursively analyze the MWD in terms of the Plotkin’s construction in Section IV. Section V presents the PC-SCEM to evaluate the MWD of concatenated polar codes. The MWD and the complexity evaluation are provided in Section VI. Section VII concludes this paper.

II. NOTATIONS AND PRELIMINARIES

A. Notation Conventions

In this paper, the lowercase letters, e.g., x, are used to denote scalars. The bold lowercase letters (e.g., x) are used to denote vectors. Notation x^j denotes the subvector \((x_1, \cdots, x_j)\) and \(x_i\) denotes the \(i\)-th element of \(x\). The sets are denoted by calligraphic characters, e.g., \(X\), and the notation \(|X|\) denotes the cardinality of \(X\). In addition, \(X \backslash x\) denotes the set with element \(x\) excluded. The bold capital letters, such as \(X\), are used to denote matrices. The element in the \(i\)-th row and the \(j\)-th column of matrix \(X\) and the \(i\)-th row of matrix \(X\) are written as \(x_{i,j}\) and \(X_i\), respectively. Furthermore, we write \(F^\otimes n\) to denote the \(n\)-th Kronecker power of \(F\) and the bit-reversal permutation is denoted by \(\pi(\cdot)\). Throughout this paper, 0 and 1 mean an all-zero vector and an all-one vector, respectively.

B. Polar Codes and Concatenated Polar Codes

Polar codes depend on the polarization effect \([1]\) of the matrix \(F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}\). For an \((N, K)\) polar code with code length \(N = 2^n\) and code rate \(R = K/N\), the polarization effect generates \(N\) polarization subchannels. Each subchannel has different reliability and the information bits are transmitted in the \(K\) most reliable subchannels. Therefore, the information set of polar codes defined by \(A\) with cardinality \(|A| = K\) is composed of the indices of the \(K\) most reliable subchannels and it is a subset of the index set \(\{1, 2, \cdots, N\}\). Then, the frozen set \(A^c\) with cardinality \(|A^c| = N - K\) is a complementary set of \(A\). The codeword \(c\) of polar codes is calculated by \(c = uBG = vG\), where \(u\) is an \(N\)-length information sequence, \(B\) is a bit-reversal permutation matrix, \(G\) is \(F^\otimes n\) and \(v = uB\).

The information sequence \(u\) is generated by assigning \(u_i\) to information bit if \(i \in A\), and assigning \(u_i\) to 0 if \(i \in A^c\). Then, according to \(v = uB\), an another information set \(B = \{j \mid j = \pi(i - 1) + 1, i \in A\}\) is obtained, which means \(v_j\) is an information bit if \(j \in B\). Here, \(\pi(\cdot)\) is a bit-reversal permutation.

For an \((N, K_i)\) concatenated polar code, the inner code is an \((N, K)\) polar code and the outer code is a \((K, K_i)\) binary linear block code. The message sequence \(b\) is first encoded by the binary linear block code to obtain the encoded sequence \(s\). Then, \(s\) is treated as the information bits of an \((N, K)\) polar code and it is inserted into the information sequence \(u\) in terms of the information set \(A\). Furthermore, a codeword of the concatenated polar code is calculated as \(c = uBG = vG\).

Without loss of generality, the binary-input additive white Gaussian noise (BI-AWGN) channel and BPSK modulation are considered in this paper. Thus, each coded bit \(c_j \in \{0, 1\}\) is modulated into the transmitted signal by \(x_j = 1 - 2c_j\). Then, the received sequence is \(y = x + n\), where \(n_i\) is i.i.d. AWGN with zero mean and variance \(\sigma^2\).

C. Sphere Decoding Algorithm

ML decoding of polar codes is equivalent to the following minimization problem

\[
\hat{v} = \arg \min_x \|y - x\|^2 = \arg \min_v \|y - (1 - 2vG)\|^2,
\]

(1)

where \(1\) is an all-one vector of length \(N\). SD algorithm can solve the problem by enumerating the possible sequence \(v\) satisfying the sphere constraint

\[
m(v^N) = \|y - (1 - 2vG)\|^2 \leq r^2,
\]

(2)

where \(r\) denotes the radius for the SD search and \(m(v^N)\) is the squared Euclidean distance along with the sequence \(v^N\).
Noting that $\mathbf{G}$ is a lower triangular matrix, we can define the partial squared Euclidean distance as

$$m(\mathbf{v}_i^N) = \sum_{k=1}^{N} |y_k - \left(1 - 2 \cdot \oplus_{j=k}^{N} (v_j g_{j,k})\right)|^2,$$

(3)

which can be computed recursively as

$$m(\mathbf{v}_i^N) = m(\mathbf{v}_{i+1}^N) + |y_i - \left(1 - 2 \cdot \oplus_{j=i}^{N} (v_j g_{j,i})\right)|^2,$$

(4)

where $\mathbf{v}_i^N$ denotes the bit decisions from the $i$-th bit to the $N$-th bit, and ‘$\oplus$’ denotes summation over $GF(2)$. According to (4), the SD algorithm can be regarded as a depth-first search on the tree and the search order is from the $N$-th bit $v_N$ to the first bit $v_1$. Once a valid sequence $\mathbf{v}$ satisfying the sphere constraint is found, the radius is updated by $\sqrt{m(\mathbf{v}_N)}$. To find the ML decoding sequence, SD adaptively updates the radius $r$. In this process, $r$ decreases rapidly so that the ML solution is efficiently captured.

### D. Distance Spectrum

The distance spectrum of an $(N, K)$ binary linear block code, designated by $A_d$, is the number of codewords of the code with the Hamming weight $d$. The pairwise error probability between two codewords modulated by BPSK differing in $d$ positions and coherently detected in the AWGN channel is $Q\left(\sqrt{\frac{2dRE_b}{N_0}}\right)$, where $E_b$ is the energy of the transmitted bit, $N_0$ is the one-sided power spectral density of AWGN and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\frac{t^2}{2}} dt$$

(5)

is the probability that a random Gaussian variable with zero mean and unit variance exceeds the value $x$. We assume that an all-zero codeword $\mathbf{0}$ is transmitted to analyze the ML performance. The union bound of ML decoding performance can be written as

$$P_e \leq \sum_{d=d_{\min}}^{N} A_d Q\left(\sqrt{\frac{2dRE_b}{N_0}}\right).$$

(6)

Then, since the MWD (i.e., $d_{\min}$ and $A_{d_{\min}}$) is the main factor influencing the ML performance when the $E_b/N_0$ is large, (6) can be approximated as

$$P_e \approx A_{d_{\min}} Q\left(\sqrt{\frac{2d_{\min}RE_b}{N_0}}\right),$$

(7)

where $d_{\min}$ is the minimum Hamming weight of the linear block code and $A_{d_{\min}}$ is the number of the codewords with $d_{\min}$. In this paper, the approximate union bound (AUB) calculated by (7) is used to evaluate the performance of polar codes.

## III. Sphere Constraint based Enumeration Method

In this section, we first illustrate the codewords distribution of polar codes and the idea of SCEM. Then, the detailed description of SCEM is provided on the basis of the codewords distribution.

### A. An Outline of SCEM

The codewords distribution of polar codes is illustrated in Fig. 1. For a polar code, the codewords with the identical Hamming weight are distributed on a spherical shell. Then, in order to analyze the MWD, the number of all the codewords with the minimum Hamming weight needs to be counted. Based on the codewords distribution, these codewords are covered by a sphere whose radius is the minimum Hamming weight. Thus, a method enumerating all these codewords constrained by the sphere can evaluate the MWD exactly.

SD can find the closest decoded sequence from the received sequence in codeword space under the radius constraint. Inspired by this idea, we propose SCEM to enumerate the codewords under sphere constraint and analyze the MWD. Similar to the SD, SCEM is regarded as a depth-first tree search as well. However, since SCEM is just used to enumerate the codewords, the noise is unnecessary.

Fig. 2 is a toy example to illustrate the process of SCEM. A binary search tree with code length 4 and sphere constraint $d_{\min}$ is shown in Fig. 2. The branches of the $i$-th level in the tree are associated with $c_{N-i+1}$. Each path from the root node to a leaf node represents a codeword. In Fig. 2, path 1 and path 2 satisfy the sphere constraint and the two paths are reserved in the search tree. On the contrary, path 3 violates the sphere constraint. Thus, all the paths attached to path 3 are pruned from the search tree, since the corresponding codewords are out of the sphere. Therefore, the proposed SCEM reserves all the codewords satisfying the sphere constraint to analyze the MWD and prunes unnecessary codewords to reduce the redundant search.

In comparison, since the SCL based methods are breadth-
Algorithm 1: The SCEM method: \( (T, A_{d_{\text{min}}}) = \text{SCEM}(N, K, B) \)

**Input:** The code length \( N \), the information bit length \( K \) and the information set \( B \);

**Output:** \( T \) is a set composed of the codewords with \( d_{\text{min}} \) and \( A_{d_{\text{min}}} \) is the number of the codewords with \( d_{\text{min}} \);

1. Initialize \( d_{\text{min}} \leftarrow \min_{i \in B} wt(G_i) \), \( T \leftarrow \emptyset \) and \( A_{d_{\text{min}}} \leftarrow 0; \)
2. Initialize the index of searching bit \( i \leftarrow N; \)
3. Initialize \( v \leftarrow 0 \) and \( c \leftarrow 0; \)
4. while \( i \leq N \) do
5.   if \( i \in B \) then   // information bit
6.     \[ v_i \leftarrow \arg \min (c_i) \text{ and } c_i \leftarrow 0; \]
7.   else   // frozen bit
8.     \[ v_i \leftarrow 0 \text{ and } c_i \leftarrow \left( \sum_{j=i}^{N} (v_j g_{ji}) \right); \]
9.   if \( d(v^N_i) \leq d_{\text{min}} \) then   // satisfy the sphere constraint
10.      if \( i > 1 \) then
11.        \[ i \leftarrow i - 1; \]
12.      else
13.        \[ T \leftarrow T \cup \{c\} \text{ and } A_{d_{\text{min}}} \leftarrow A_{d_{\text{min}}} + 1; \]
14.        Go to Step 16;
15.   else   // prune the search tree
16.     while \( i \leq N \) do
17.       if \( i \in B \) and \( c_i = 0 \) then
18.         \[ v_i \leftarrow v_i \oplus 1 \text{ and } c_i \leftarrow 1; \]
19.         Go to Step 9;
20.       else
21.         \[ i \leftarrow i + 1; \]
22.     \[ T \leftarrow T - \{0\} \text{ and } A_{d_{\text{min}}} \leftarrow A_{d_{\text{min}}} - 1; \]

According to (3), the sphere constraint used in the SCEM is decided, which is
\[ wt(c) \leq d_{\text{min}}. \tag{9} \]
Thus, the codewords satisfying (9) is on the surface of the sphere constraint \( c = 0 \).

Then, since \( G \) is a lower triangular matrix, \( c_i \) is only related to the subvector \( v^N_i \), i.e.,
\[ c_i = \bigoplus_{j=i}^{N} (v_j g_{ji}). \tag{10} \]
Thus, the partial Hamming weight of \( c \) is defined as
\[ d(v^N_i) \triangleq wt(v^N_i) = \sum_{k=i}^{N} \left( \bigoplus_{j=k}^{N} (v_j g_{jk}) \right), \tag{11} \]
which can be calculated recursively as
\[ d(v^N_i) = d(v^N_{i+1}) + \left( \bigoplus_{j=i}^{N} (v_j g_{ji}) \right). \tag{12} \]

According to (12), the process of enumerating all the codewords on the surface of the sphere can be treated as a depth-first tree search and the search order is from \( v_N \) to \( v_1 \). Then, when \( v_i \) is decided, \( d(v^N_i) \) is decided as well. Therefore, the sphere constraint (3) can be simplified as
\[ d(v^N_i) \leq d_{\text{min}}, \tag{13} \]
which means that the Hamming weight of the codewords attached to \( v^N_i \) is larger than \( d_{\text{min}} \) when (13) is false. Hence, these codewords need to be pruned from the search tree to avoid the useless search.

Algorithm 1 describes the entire procedure of SCEM, where \( N \) is the code length, \( K \) is the information bit length and \( B \) is the information set about \( v \). Without loss of generality, for describing the SCEM easily, we set the branch with \( c_i = 0 \) as the first searching branch during deciding information bit \( v_i \). Then, when the search of the branch with \( c_i = 0 \) is completed, SCEM continues to search the branch with \( c_i = 1 \).

In Algorithm 1, the search order is from \( v_N \) to \( v_1 \). Thus, in terms of (12), when a bit \( v_i \) is decided, whether \( d(v^N_i) \) satisfies the Hamming weight constraint (13) or not is judged. If satisfying, the search continues to decide next bit \( v_{i-1} \) until that a codeword with Hamming weight \( d_{\text{min}} \) is enumerated. If not, the nodes attached to the path \( v^N_i \) are pruned from the search tree and the search goes on from a new branch of the tree. By repeating the search process, all the \( A_{d_{\text{min}}} \) codewords with Hamming weight \( d_{\text{min}} \) is enumerated and these codewords are recorded into a codeword set \( T \). Thus, the MWD \( T \) is obtained by Algorithm 1, i.e.,
\[ T = \{c \mid wt(c) = d_{\text{min}}, c \in C \}. \tag{14} \]

IV. Sphere Constraint Based Recursive Enumeration Method

In this section, we first prove the MWD relationship between a polar code and two component polar codes based on the Plotkin’s construction. Then, according to the MWD relationship and the SCEM, we design the SCREM.
A. MWD Relationship Based on the Plotkin’s Construction

According to the Plotkin’s construction of polar codes [18], a polar code can be divided into two component polar codes. Then, we find that the MWD of the polar code is related with the MWDs of the two component codes and prove the MWD relationship. Based on this, the MWD can be enumerated recursively.

To prove the MWD relationship, we first describe the Plotkin’s construction of polar codes as follows. For an \((N, K)\) polar code \(C\) with information set \(B\), the encoding process can be expressed as

\[
c = vG = (v', v'') \begin{bmatrix} G' & 0 \\ G' & G' \end{bmatrix} = (c' \oplus c'', c'') = (c' \oplus v'G', v'G'), \tag{15}
\]

where \(G' = F^{\oplus(n-1)}\), \(c' = v'G'\), and \(c'' = v''G'\). Then, \(c'\) and \(c''\) are the codewords of an \((N, K')\) polar code \(C'\) and an \((N, K'')\) polar code \(C''\), respectively. The information set of \(C''\) is denoted by \(B''\) and

\[
B'' = \left\{ i \in B : 1 \leq i \leq N, \frac{N}{2} + 1 \leq i \leq N \right\}. \tag{16}
\]

Similarly, \(B''\) is the information set of \(C''\) and

\[
B'' = \left\{ i - \frac{N}{2} \in B : \frac{N}{2} + 1 \leq i \leq N \right\}. \tag{17}
\]

In addition, \(K' = |B'|\) and \(K'' = |B''|\). The minimum Hamming weight of \(C'\) and \(C''\) is denoted by \(d'_{\text{min}}\) and \(d''_{\text{min}}\), respectively.

Then, in order to prove the MWD relationship among \(C, C'\), and \(C''\), we first prove Lemma 1 and Lemma 2 as follows.

**Lemma 1.** \(C'\) is a subcode of \(C''\).

**Proof:** According to the partial order [19], if \(v_i\) is an information bit, \(v_{i+\frac{N}{2}}\) is also an information bit. Thus, if \(i \in B'\), we have \(i \in B''\). Therefore, \(C'\) is a subcode of \(C''\).

**Lemma 2.** \(d'_{\text{min}}\) and \(d''_{\text{min}}\) have three combinations:

1) \(d'_{\text{min}} = d_{\text{min}}\) and \(d''_{\text{min}} = d_{\text{min}}\).
2) \(d'_{\text{min}} = d_{\text{min}}\) and \(d''_{\text{min}} = \frac{d_{\text{min}}}{2}\).
3) \(d'_{\text{min}} > d_{\text{min}}\) and \(d''_{\text{min}} = \frac{d_{\text{min}}}{2}\).

**Proof:** Due to Lemma 1, we have

\[
d'_{\text{min}} \geq d''_{\text{min}}. \tag{18}
\]

According to [16, Sec 4.4], we can obtain

\[
d_{\text{min}} = \min (2d''_{\text{min}}, d_{\text{min}}). \tag{19}
\]

Supposing \(2d''_{\text{min}} \geq d'_{\text{min}}\), according to [18] and [19], we have

\[
\left\{ \begin{array}{l}
2d''_{\text{min}} \geq d'_{\text{min}} \geq d''_{\text{min}} \\
d'_{\text{min}} = d_{\text{min}}
\end{array} \right. \tag{20}
\]

Then, we obtain

\[
d_{\text{min}} = \frac{d'_{\text{min}} + d''_{\text{min}}}{2} \leq d''_{\text{min}} \leq d_{\text{min}}. \tag{21}
\]

Furthermore, since

\[
wt(G_i) = 2^j, \quad \exists j \in \{1, 2, \cdots, n\}, \tag{22}
\]

we have

\[
\left\{ \begin{array}{l}
d'_{\text{min}} = d_{\text{min}} \\
2d''_{\text{min}} = d_{\text{min}}
\end{array} \right. \quad \text{or} \quad d_{\text{min}}. \tag{23}
\]

Then, supposing \(2d''_{\text{min}} < d'_{\text{min}}\), similarly, according to [18] and [19], we can obtain that \(d''_{\text{min}} = \frac{d_{\text{min}}}{2}\) and \(d'_{\text{min}} > d_{\text{min}}\).

From the above, Lemma 2 has been proved. According to Lemma 1 and Lemma 2, the MWD relationship among \(C, C'\), and \(C''\) is provided as Lemma 3.

**Lemma 3.**

Given \(T, T'\) and \(T''\) are the codeword sets of \(C, C'\) and \(C''\) with Hamming weights \(d_{\text{min}}, d'_{\text{min}}\) and \(d''_{\text{min}}\), respectively.

1) When \(d'_{\text{min}} = d_{\text{min}}\) and \(d''_{\text{min}} = d_{\text{min}}\), we have \(T = T_1 \cup T_2\). \tag{24}
2) When \(d'_{\text{min}} = d_{\text{min}}\) and \(d''_{\text{min}} = \frac{d_{\text{min}}}{2}\), we have \(T = T_1 \cup T_2 \cup T_3 \cup T_4\). \tag{25}
3) When \(d'_{\text{min}} > d_{\text{min}}\) and \(d''_{\text{min}} = \frac{d_{\text{min}}}{2}\), we have \(T = T_3\). \tag{26}

In (24), (25) and (26).

\[
\begin{align*}
T_1 &= \{(c', 0) | c' \in T'\}, \tag{27} \\
T_2 &= \{(0, c') | c' \in T'\}, \tag{28} \\
T_3 &= \{(c'', c') | c' \in T''\}, \tag{29} \\
T_4 &= \{(c' \oplus c'', c'') | c' \in T', c'' \in T'', \text{wt}(c' \oplus c'') = \frac{d_{\text{min}}}{2}\}. \tag{30}
\end{align*}
\]

**Proof:** See the Appendix.

**Lemma 3** describes the relationship among \(T, T'\), and \(T''\). Based on this, \(T\) can be directly decided by \(T'\) and \(T''\).

B. Detailed Description of SCREM

In order to exploit Lemma 3 to analyze the MWD of \(C\), the MWD of \(C'\) and \(C''\) need to be evaluated first. Fortunately, the SCREM can be used to exactly enumerate the MWD of \(C'\) and \(C''\). Then, due to the recursive structure of the generator matrix of polar codes, the MWD of \(C\) can be enumerated recursively by using the SCREM and Lemma 3.

The SCREM is proposed to enumerate the MWD of polar codes recursively and the method is described as Algorithm 2. In Algorithm 2, polar code \(C\) is first divided into two component polar codes \(C'\) and \(C''\) in terms of the Plotkin’s construction (step 1 to 4). Then, \(T\) can be obtained by \(T'\) and \(T''\) on the basis of Lemma 3 (step 10 to 16). Also, SCREM is used to enumerate \(T'\) and \(T''\) (step 8 and 9). Hence, the MWD of \(C\) can be analyzed recursively. In addition, when \(C\) cannot be divided into two component codes, i.e., \(N = 2\) or \(K' = 0\), or the division cannot reduce the search complexity, i.e., \(K'' = \frac{N}{2}\), the recursion is stop and the SCREM is used to enumerate the MWD of \(C\) (step 5 and 6). Thus, according to Lemma 3 and the SCREM, we can obtain \(T\) recursively.
V. PARITY-CHECK SCEM

In this section, we first describe the parity-check equations and transform them into new forms to match the search order of the PC-SCEM. Then, the detailed description of PC-SCEM is provided.

A. Parity-Check Equations

For an \((N, K_f)\) concatenated polar code, the inner code is an \((N, K)\) polar code and the outer code is a \((K, K_f)\) binary linear block code. The parity-check matrix and the codeword of the binary linear block code are denoted by \(H\) and \(s\), respectively. Each row of \(H\) represents a parity-check equation, i.e.,

\[
\bigoplus_{j=1}^{K_f} h_{i,j} s_j = 0, \quad i = 1, 2, \cdots, K_P, \tag{31}
\]

where \(K_P = K - K_f\) is the number of the parity-check equations.

Then, parity-check sets are used in this paper to represent the parity-check equations.

Algorithm 2: The SCREM: \((T, A_{d_{\min}}) = \text{SCREM}(N, K, B, d_{\min})\)

**Input:** The code length \(N\), the information bit length \(K\), the information set \(B\) and the minimum Hamming weight \(d_{\min}\).

**Output:** \(T\) is the codeword set with \(d_{\min}\) and \(A_{d_{\min}}\) is the number of the codewords with \(d_{\min}\).

1. Initialize \(T \leftarrow \emptyset \) and \(A_{d_{\min}} \leftarrow 0\);
2. Initialize \(B'\) and \(B''\) by \(16\) and \(17\), respectively;
3. Initialize \(K'\) and \(K''\) by \(|B'|\) and \(|B''|\), respectively;
4. \(d'_{\min}\) and \(d''_{\min}\) are the minimum Hamming weight of \(C'\) and \(C''\), respectively;
5. if \(N = 2\) or \(K' = 0\) or \(K'' = \frac{N}{2}\) then
6. \(\langle T, A_{d_{\min}} \rangle \leftarrow \text{SCREM}(N, K, B, d_{\min});\)
else
7. \(\langle T', A_{d_{\min}} \rangle \leftarrow \text{SCREM}(\frac{N}{2}, K', B', d'_{\min});\)
8. \(\langle T'', A_{d_{\min}} \rangle \leftarrow \text{SCREM}(\frac{N}{2}, K'', B'', d''_{\min});\)
9. if \(d'_{\min} = d_{\min}\) and \(d''_{\min} = d_{\min}\) then
10. // case 1
11. \(T \leftarrow T_1 \cup T_2\) and \(A_{d_{\min}} \leftarrow 2A_{d_{\min}};\)
else if \(d'_{\min} = d_{\min}\) and \(d''_{\min} = \frac{d_{\min}}{2}\) then
12. // case 2
13. Obtain \(T_1\) by enumerating all the combinations of \(c'\) and \(c''\) which satisfy \(c' \in T',\ c'' \in T''\) and \(wt(c' \oplus c'') = \frac{d_{\min}}{2}\);
14. \(T \leftarrow T_1 \cup T_2 \cup T_3 \cup T_4\) and \(A_{d_{\min}} \leftarrow 2A_{d_{\min}} + A_{d''_{\min}} + |T_4|;\)
else if \(d'_{\min} > d_{\min}\) and \(d''_{\min} = \frac{d_{\min}}{2}\) then
16. // case 3
17. \(T \leftarrow T_3\) and \(A_{d_{\min}} \leftarrow A_{d''_{\min}};\)

Algorithm 3: \(\{Q_i(v)\} = \text{Transform}(\{R_i(v)\})\)

**Input:** The parity-check sets \(R_i(v), \ i = 1, 2, \cdots, K_P;\)

**Output:** Total \(K_P\) transformed parity-check sets \(Q_i(v);\)

1. Initialize \(D\) as a \(K_P \times N\) matrix and the \(i\)-th row of \(D\) represents the parity-check equation obtained by \(R_i(v);\)
2. Employ GE on the rows of \(D\) and obtain a row echelon form matrix \(E;\)
3. Each row of \(E\) represents the newly parity-check equation and the corresponding parity-check set is \(Q_i(v);\)

Definition 1. The parity-check sets corresponding to the parity-check equations of \(s\) are given as

\[
R_i(s) \triangleq \{j|h_{i,j} = 1\}, \ i = 1, 2, \cdots, K_P. \tag{32}
\]

Since \(s\) is inserted into \(u\) in terms of the information set \(A\), the parity-check sets corresponding to \(u\) are defined as

\[
R_i(u) = \{t|t = f(j), j \in R_i(s)\}, \ i = 1, 2, \cdots, K_P \tag{33}
\]

where the function \(f(t)\) is the index mapping from \(s\) to \(u\) and \(f(t)\) is different for various concatenated polar code schemes. Then, the parity-check sets \(R_i(v)\) corresponding to \(v\) are derived by performing the bit-reversal permutation to all the elements in \(R_i(u)\), i.e.,

\[
R_i(v) = \{k|k = \pi(t - 1) + 1, t \in R_i(u)\}, \ i = 1, 2, \cdots, K_P \tag{34}
\]

In terms of the definition of parity-check sets, for any \(i = 1, 2, \cdots, K_P\), we have

\[
\bigoplus_{j \in R_i(s)} s_j = \bigoplus_{t \in R_i(u)} u_t = \bigoplus_{k \in R_i(v)} v_k = 0. \tag{35}
\]

Since the search order of the PC-SCEM method is from \(v_N\) to \(v_1\), the bit with the least index in each parity-check set \(R_i(v)\) can be directly judged by the previous searched bits. Followed by this, the definition of parity-check bit index of the parity-check set is given.

Definition 2. The index \(k_i\) of parity-check bit corresponding to \(R_i(v)\) is defined as \(k_i = \min(R_i(v)).\)

Similar to the CA-SD [17], if two more parity-check sets have the same index of parity-check bit, the colliding decision phenomenon will happen where this parity-check bit cannot be uniquely judged. This severe problem leads to the search error. However, the above colliding decision problem can be solved by the linear combination of multiple parity-check equations. Thus, to avoid the colliding decision, Gaussian elimination (GE) is used to transform the parity-check equations into new forms to ensure the indices of parity-check bits are different with each other. The process is described in Algorithm 3.

In Algorithm 3 the \(i\)-th row of a \(K_P \times N\) matrix \(D\) is first initialized in terms of the parity-check equation obtained by \(R_i(v);\) Then, GE is used on the rows of \(D\) to obtain a row echelon form matrix \(E;\) Finally, the newly parity-check set \(Q_i(v)\) is obtained by the \(i\)-th row of \(E, \ i = 1, 2, \cdots, K_P\) and
Algorithm 4: The PC-SCEM

Input: $N$, $B$ and $(Q_i(v))$;  
Output: $T$, $d_{\text{min}}$ and $A_{d_{\text{min}}}$;  
1 Initialize $T \leftarrow \emptyset$, $A_{d_{\text{min}}} \leftarrow 0$ and $r \leftarrow \min_{i \in B}(\text{wt}(G_i))$;  
2 Initialize $k \leftarrow N$, $v \leftarrow 0$ and $c \leftarrow 0$;  
3 Initialize $P = \{ k_i \mid k_i = \min(Q_i(v)), i = 1, 2, \ldots, K_P \}$;  
4 while $A_{d_{\text{min}}} = 0$ do  
5 
6 
7 
8 
9 
10 
11 
12 
13 
14 
15 
16 
17 
18 
19 
20 
21 
22 
23 
24 
25 
26 
27 
28 
29 
30 
31 

while $k \leq N$ do  

if $k \in B - P$ then // information bit  

$v_k \leftarrow \arg \min_{v_k \in \{0,1\}} (c_k)$ and $c_k \leftarrow 0$;  

else if $k \in P$ then // parity-check bit  

Find $i$ making $k_i = k$;  

$v_k \leftarrow \bigoplus_{i \in (Q_i(v) \backslash k_i)} v_i$;  

$c_k \leftarrow \left( \sum_{j = k}^{N} (v_j g_{j,k}) \right)$;  

else // frozen bit  

$v_k \leftarrow 0$ and $c_k \leftarrow \left( \sum_{j = k}^{N} (v_j g_{j,k}) \right)$;  

if $d(v_k^N) \leq r$ then // satisfy the sphere constraint  

if $k > 1$ then  

$k \leftarrow k - 1$;  

else  

$T \leftarrow T \cup \{c\}$ and $A_{d_{\text{min}}} \leftarrow A_{d_{\text{min}}} + 1$;  

Go to Step 21;  

else // prune the search tree  

while $k \leq N$ do  

if $k \in B - P$ and $c_k = 0$ then  

$v_k \leftarrow v_k + 1$ and $c_k \leftarrow 1$;  

Go to Step 14;  

else  

$k \leftarrow k + 1$;  

end if  

end while  

$T \leftarrow T - \{0\}$ and $A_{d_{\text{min}}} \leftarrow A_{d_{\text{min}}} - 1$;  

if $A_{d_{\text{min}}} = 0$ then  

$r \leftarrow r + 2$;  

else  

$d_{\text{min}} \leftarrow r$;  

end if  

end while  

end if  

end if  

end if

The parity-check bits of these sets are different due to the row echelon form. Thus, the parity-check bit indices are

$$k_i = \min(Q_i(v)), i = 1, 2, \ldots, K_P. \tag{36}$$

and these bits can be decided by

$$v_{k_i} = \bigoplus_{k \in (Q_i(v) \backslash k_i)} v_k, i = 1, 2, \ldots, K_P. \tag{37}$$

B. Detailed Description of PC-SCEM

Since all the codewords of concatenated polar codes with the Hamming weight $d_{\text{min}}$ in the codeword space is on the surface of a sphere, a sphere constraint can also be used to enumerate these codewords with $d_{\text{min}}$. Based on this, an PC-SCEM is proposed to analyze the MWD of concatenated polar codes.

Algorithm 4 describes the entire procedure of the PC-SCEM. In the method, since $d_{\text{min}}$ determines the sphere constraint, deciding $d_{\text{min}}$ is the first thing to analyze the MWD. However, there are no simple methods to calculate $d_{\text{min}}$ of concatenated polar codes. Therefore, a greedy method is used to determine $d_{\text{min}}$. We first set the radius of the sphere constraint with a lower bound of $d_{\text{min}}$. Since concatenated polar codes are the subcode of the corresponding polar codes, the minimum Hamming weight of polar codes is the lower bound of the minimum Hamming weight of concatenated polar code. Thus, the radius is first set as

$$r = \min_{i \in B}(\text{wt}(G_i)). \tag{38}$$

Then, considering that polar code is the subcode of RM code and the Hamming weight of the codewords of RM code is even [16, Sec. 4.3], the Hamming weight of the codewords of concatenated polar codes is even as well. Thus, if no codewords can be enumerated in the sphere constraint (38), $r$ is added 2 until finding codewords in the sphere constraint and $r$ is the $d_{\text{min}}$ of the concatenated polar code.

To enumerate the codewords in the sphere constraint, all the bits are divided into three types: information bits, frozen bits and parity-check bits. For the information bits and frozen bits, the search process is same as that in the SCEM. For the parity-check bits, they are directly calculated by the previous searched bits according to the corresponding $Q_i(v)$ such that the codewords searched by the PC-SCEM belong to the MWD of the concatenated polar code.

VI. MWD AND COMPLEXITY EVALUATION

In this section, we first provide the MWD of polar codes. Then, the optimal CRC polynomial of the CRC-polar concatenated codes and the corresponding MWD are provided. Finally, the complexity comparison between the three proposed methods and the SCL based methods is provided. The improved GA [20] and the polarization weight (PW) [21] are applied to construct polar codes.

A. MWD of Polar Codes

In this subsection, the MWD of polar codes with different code rates is first provided. Then, we provide the MWD of polar codes with different SNR. Finally, the BLER performance of polar codes decoded by SCL with list size 32 and the corresponding A UB are provided.

Table V provides the MWD of polar codes constructed by GA and PW with different code lengths and code rates. Since the MWD of the polar codes constructed by GA changes along with SNR, the polar codes are constructed at $E_b/N_0 = 3$dB. The polar codes marked with “*” are constructed at $E_b/N_0 = 2.5$dB, since $A_{d_{\text{min}}}$ of these polar codes at $E_b/N_0 = 3$dB is so large that the codewords are difficult to be enumerated. In Table V, we can observe that the MWD of polar codes constructed by GA and PW with $N = 256$ is almost the same.
### TABLE I
The MWD of polar codes constructed by GA and PW with different code lengths and code rates.

| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
|---|---|---|---|---|
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |

Then, the difference of the MWD between GA and PW occurs and becomes larger as the code length increases. Specifically, the MWD of polar codes constructed by GA has larger \(d_{\text{min}}\) or less \(A_{\text{dmin}}\). Based on this, we can explain why GA and PW have almost the same performance for short polar codes, but the performance of GA is better for long polar codes generally.

Table III provides the MWD of polar codes with code rate 1/2 and different code lengths at various SNR. Polar codes are constructed by GA. In Table III, we can observe that the MWD of polar codes constructed by GA is variable with the change of SNR. The reason is that the information set obtained in terms of GA is distinct in different SNR regions. In addition, the MWD with fixed code length and code rate is

---

### TABLE II
The MWD of polar codes constructed by GA in different SNR with code rate 1/2 and different code lengths.

| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
|---|---|---|---|---|
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |
| &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. | &d 자체한 내용을 제공하지 않습니다. |

Then, the difference of the MWD between GA and PW occurs and becomes larger as the code length increases. Specifically, the MWD of polar codes constructed by GA has larger \(d_{\text{min}}\) or less \(A_{\text{dmin}}\). Based on this, we can explain why GA and PW have almost the same performance for short polar codes, but the performance of GA is better for long polar codes generally.

Table III provides the MWD of polar codes with code rate 1/2 and different code lengths at various SNR. Polar codes are constructed by GA. In Table III, we can observe that the MWD of polar codes constructed by GA is variable with the change of SNR. The reason is that the information set obtained in terms of GA is distinct in different SNR regions. In addition, the MWD with fixed code length and code rate is

---

**Fig. 3.** The BLER performance of polar codes decoded by SCL with list size 32.
also improved as the SNR increases, i.e., larger $d_{\text{min}}$ or less $A_{d_{\text{min}}}$. Thus, no error floor occurs in the performance curve of GA due to the improved MWD. Moreover, compared with the MWD of polar codes constructed by PW (shown in Table I), the polar codes constructed by GA has better AUB in the high SNR region, which leads the performance is better in Fig. 3.

Fig. 3 shows the BLER performance of polar codes with code rate $1/2$ and code length 1024 and 2048. According to Table I and Table III the AUB calculated by (7) is provided. In Fig. 3 we can observe that the BLER performance coincides with the corresponding AUB. Thus, the AUB calculated by the MWD can be used to evaluate the BLER performance of polar codes. Then, in the low SNR region, the performance of polar codes constructed by GA is almost the same as that constructed by PW. As the SNR increases, the performance gap between them occurs and becomes larger. The reason is that the MWD of polar codes constructed by GA is improved with the increase of SNR and better than that constructed by PW. Therefore, from the viewpoint of MWD, GA is appropriate for constructing long polar codes rather than PW.

### B. MWD of CRC-Polar Concatenated Codes

In this subsection, we first provide the MWD of CRC-polar concatenated codes with optimal CRC polynomial. Then, the corresponding BLER performance are provided.

Table III provides the MWD of CRC-polar concatenated codes for different code lengths and code rates with optimal CRC and standard CRC. PW is used to construct CRC-polar concatenated codes. The standard CRC polynomials are provided in [22]. By exhausting all the CRC polynomial and analyzing the corresponding MWD of CRC-polar concatenated codes, the optimal CRC is obtained. The optimization principles are 1) maximizing $d_{\text{min}}$ and 2) minimizing $A_{d_{\text{min}}}$ when $d_{\text{min}}$ is identical.

Fig. 4 shows the BLER performance of CRC-polar concatenated codes with the optimal CRC and the standard CRC, where code length is 128 and CRC length is 6. As shown in Table III, the optimal CRC polynomials for codes rates $1/4, 1/2$ and $3/4$ are 0x5B, 0x73 and 0x73, respectively, and the standard CRC polynomial is 0x59. In Fig. 4 the BLER performance is close to the AUB in the high SNR region. Then, since the CRC-polar concatenated codes with the optimal CRC has better MWD, the performance is better in the medium to high SNR regions.

Fig. 5 illustrates the BLER performance of CRC-polar concatenated codes with code length 512 and 11-bit CRC. As shown in Table III, the optimal CRC polynomials for code rates $1/4, 1/2$ and $3/4$ are 0x9A7, 0xC23 and 0xE81, respectively, and the standard CRC polynomial is 0xCBB. Similarly, in Fig. 5 the BLER performance is also close to the AUB and the performance of the optimal CRC is better in the high SNR region.

### C. Complexity Evaluation

In this subsection, we provide the complexity comparison between the proposed methods and the SCL based methods.

| $N$ | $K_L$ | $K_P$ | Optimal CRC | Standard CRC |
|-----|-------|-------|-------------|--------------|
| 6   | 32    | $g(x)$ | $d_{\text{min}}$ | $A_{d_{\text{min}}}$ |
| 8   | 0x1E7 | 24    | 128         | 0x59         | 16 | 12 |
| 11  | 0xD11 | 24    | 34          | 0xCBB        | 16 | 3  |
| 128 | 64    | $g(x)$ | $d_{\text{min}}$ | $A_{d_{\text{min}}}$ |
| 8   | 0xD7  | 16    | 5853        | 0x59         | 12 | 25 |
| 11  | 0xC31 | 16    | 120         | 0x59         | 12 | 16 |
| 192 | 8     | $g(x)$ | $d_{\text{min}}$ | $A_{d_{\text{min}}}$ |
| 11  | 0xCB9 | 8     | 335         | 0xCBB        | 12 | 98 |
| 6   | 0x43  | 32    | 498         | 0x59         | 32 | 1036 |
| 128 | 8     | $g(x)$ | $d_{\text{min}}$ | $A_{d_{\text{min}}}$ |
| 11  | 0x9A7 | 32    | 3           | 0xCBB        | 32 | 32 |
| 512 | 256   | $g(x)$ | $d_{\text{min}}$ | $A_{d_{\text{min}}}$ |
| 8   | 0xC3  | 16    | 1912        | 0x59         | 16 | 4344 |
| 384 | 6     | $g(x)$ | $d_{\text{min}}$ | $A_{d_{\text{min}}}$ |
| 8   | 0xE81 | 8     | 6           | 0xCBB        | 8  | 708 |

Fig. 4. The BLER performance of CRC-polar concatenated codes with code length 128 and CRC length 6.

Fig. 6 illustrates the complexities of the SCEM, the SCREM and the SCL based methods with code length 128 and different code rates. Considering the MWD of polar codes constructed by GA changes along with SNR, PW is used in Fig. 6. The complexity of SCEM and SCREM is counted by the average visited nodes (AVN). The AVN of enumerating all the codewords is $2^L N \log N$, which is the upper bound of the complexity of enumerating MWD. The AVN of SCL method [10] with list size $L_1$ are $\min (2^K N \log N, L_1 N \log N)$. 
In this paper, we propose three methods to analyze the MWD of polar codes. The SCEM is first proposed to exactly enumerate all the codewords belonging to the MWD, which exploits the distance property that all the codewords with the identical Hamming weight are distributed on a spherical shell. Then, based on the SCEM and the Plotkin’s construction, we propose SCREM to recursively analyze the MWD with lower complexity. Finally, the PC-SCEM is proposed by introducing the parity-check equations and the sphere constraint to analyze the MWD of concatenated polar codes. The experimental results illustrate that the complexities of the proposed SCEM and SCREM with code length 128 are lower than those of the SCL based methods.

APPENDIX

According to (15), a codeword \( \mathbf{c} \) with minimum Hamming weight \( d_{\min} \) can be expressed as

\[
wt(\mathbf{c}) = wt(\mathbf{c'}) + wt(\mathbf{c''}) = d_{\min}.
\]

Then, \( d'_{\min} \) and \( d''_{\min} \) are divided into three cases by Lemma 2. In the proof of each case, classified discussion is used.

1) When \( d'_{\min} = d_{\min} \) and \( d''_{\min} = d_{\min} \), \( T \) is obtained as follows.
   a) Supposing \( wt(\mathbf{c''}) = 0 \), \( wt(\mathbf{c}) \) is simplified as

\[
wt(\mathbf{c}) = wt(\mathbf{c'}) = d_{\min}.
\]

Thus, \( \forall \mathbf{c'} \in T' \) makes \( wt(\mathbf{c}) \) is \( d_{\min} \).
   b) Supposing \( wt(\mathbf{c''}) = d_{\min} \), similarly, \( wt(\mathbf{c}) \) is simplified as

\[
wt(\mathbf{c'}) + wt(\mathbf{c''}) = 0.
\]

Hence, we have \( \mathbf{c''} = \mathbf{c'} \). Furthermore, according to Lemma 1, we have \( T' \subseteq T'' \). Therefore, for \( \forall \mathbf{c'} \in T' \), \( \exists \mathbf{c''} \in T'' \) makes \( wt(\mathbf{c'} + \mathbf{c''}) = 0 \), i.e., \( \mathbf{c''} = \mathbf{c'} \). Thus, \( (0, \mathbf{c'}), \mathbf{c'} \in T', \) is the codeword of \( C \) and its Hamming weight is \( d_{\min} \).
   c) Supposing \( wt(\mathbf{c''}) > d_{\min} \), it is clear that \( wt(\mathbf{c}) > d_{\min} \).

In conclusion, \( T = T_1 \cup T_2 \), where \( T_1 = \{ (\mathbf{c'}, 0) | \mathbf{c'} \in T' \} \) and \( T_2 = \{ (0, \mathbf{c'}) | \mathbf{c'} \in T' \} \).

2) When \( d'_{\min} = d_{\min} \) and \( d''_{\min} = \frac{d_{\min}}{2} \), \( T \) is obtained as follows.
   a) Supposing \( wt(\mathbf{c''}) = 0 \), \( \forall \mathbf{c'} \in T' \) makes \( wt(\mathbf{c}) \) is \( d_{\min} \).
   b) Supposing \( wt(\mathbf{c'}) = \frac{d_{\min}}{2} \) and \( wt(\mathbf{c'}) = 0 \), obviously, \( (\mathbf{c'}, \mathbf{c''}) \) is the codeword of \( C \) and its Hamming weight is \( d_{\min} \).
   c) Supposing \( wt(\mathbf{c'}) = \frac{d_{\min}}{2} \) and \( wt(\mathbf{c'}) = d_{\min} \), to obtain the codeword \( \mathbf{c} \) with \( d_{\min} \), all the \( \mathbf{c'} \in T' \) and \( \mathbf{c''} \in T'' \) are enumerated to satisfy

\[
wt(\mathbf{c'} + \mathbf{c''}) = d_{\min} \frac{2}{2}.
\]

Thus, \( wt(\mathbf{c}) > d_{\min} \).

In conclusion, \( T = T_1 \cup T_2 \), \( T_1 = \{ (\mathbf{c'}, 0) | \mathbf{c'} \in T' \} \) and \( T_2 = \{ (0, \mathbf{c'}) | \mathbf{c'} \in T' \} \).

The AVN of multi-level SCL method [11] with list size \( L_2 \) and level number \( M \) are \( \min(2^K N \log N, ML_2 N \log N) \). \( L_1 \) and \( L_2 \) used in [10] and [11] are 1280000 and 32768, respectively, and \( M \) is the number of row with the row weight \( d_{\min} \) in the generator matrix of polar codes.

In Fig. 6 the complexity of SCEM is lower than those of the SCL based methods, since the sphere constraint can prune the search tree to reduce the redundant search. Specifically, the AVN of the proposed SCEM achieves 3 to 4 magnitude reduction compared with both the SCL method and the multi-level SCL method. Furthermore, due to the recursive structure of SCREM, its complexity is lower than the SCEM. In comparison to the SCEM, the SCL method and the multi-level SCL method, the SCREM can achieve up to \( 10^9 \), \( 10^8 \), and \( 10^7 \) times complexity reduction.
From the above, Lemma 3 has been proved.

3) When

Then, according to Lemma, \( c' \) is the codeword of \( C'' \). Thus, \( c' + c'' \) is also the codeword of \( C'' \). However, since \( d_{\min}'' \) is \( d_{\min}'' \), no codeword of \( C'' \) can satisfy (43). Therefore, in this case, no codeword of \( C \) with \( d_{\min} \) can be found.

f) Supposing \( wt(c'') = d_{\min} \), \( wt(c) \) is simplified as

According to the 1)-b) of the proof of Lemma, \( (0, c') \), \( c' \in T' \), is the codeword of \( C \) and its Hamming weight is \( d_{\min} \).

g) Supposing \( wt(c'') > d_{\min} \), obviously, \( wt(c) > d_{\min} \).

In conclusion, \( T = T_1 \cup T_2 \cup T_3 \cup T_4 \), where \( T_3 = \{ (c'', c') \mid c'' \in T'' \} \) and \( T_4 = \{ (c' + c'', c') \mid c' \in T', c'' \in T'' \} \).

3) When \( d_{\min}'' > d_{\min} \) and \( d_{\min}' = \frac{d_{\min}''}{2} \), \( T \) is obtained as follows.

a) Supposing \( wt(c'') = 0 \), it is clear that \( wt(c) > d_{\min} \).

b) Supposing \( wt(c'') = \frac{d_{\min}}{2} \) and \( wt(c') = 0 \), obviously, \( (c', c'') \) is the codeword of \( C \) and its Hamming weight is \( d_{\min} \).

c) Supposing \( wt(c'') = \frac{d_{\min}}{2} \) and \( wt(c') > d_{\min} \), we have

Thus, \( wt(c) > d_{\min} \).

d) Supposing \( \frac{d_{\min}}{2} < wt(c'') < d_{\min} \), in order to make \( wt(c) \) is \( d_{\min} \), we have

Then, according to the 2)-e) of the proof of Lemma, no codeword of \( C \) with \( d_{\min} \) can be found in this case.

e) Supposing \( wt(c'') = d_{\min} \), we have \( wt(c' + c'') > 0 \).

Thus, \( wt(c) > d_{\min} \).

f) Supposing \( wt(c'') > d_{\min} \), obviously, \( wt(c) > d_{\min} \).

In conclusion, \( T_3 = \{ (c', c'') \mid c'' \in T'' \} \).

From the above, Lemma 3 has been proved.

References

[1] E. Arıkan, “Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels,” IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] I. Tal and A. Vardy, “List decoding of polar codes,” IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2213–2226, May. 2015.

[3] K. Chen, K. Niu, and J. R. Lin, “List successive cancellation decoding of polar codes,” Electron. Lett., vol. 48, no. 9, pp. 500–501, 2012.

[4] K. Chen, K. Niu and J. Lin, “Improved Successive Cancellation Decoding of Polar Codes,” IEEE Trans. Commun., vol. 61, no. 8, pp. 3100–3107, Aug. 2013.

[5] K. Niu and K. Chen, “CRC-aided decoding of polar codes,” IEEE Commun. Lett., vol. 16, no. 10, pp. 1668–1671, Oct. 2012.

[6] 3rd Generation Partnership Project (3GPP), “Multiplexing and channel coding,” 3GPP TS 38.212 V15.0.0, 2017.

[7] J. Dai, J. Gao and K. Niu, “Learning to Mitigate the FAR in Polar Code Blind Detection,” IEEE Wireless Commun. Lett., early access.

[8] B. Li, H. Shen and D. Tse, “An Adaptive Successive Cancellation List Decoder for Polar Codes with Cyclic Redundancy Check,” IEEE Commun. Lett., vol. 16, no. 12, pp. 2044–2047, December 2012.

[9] S. Lin and D. J. Costello, “Error Control Coding (2nd ed.),” Prentice Hall, Inc., 2004.

[10] Z. Liu, K. Chen, K. Niu and Z. He, “Distance spectrum analysis of polar codes,” in Proc. 2014 IEEE WCNC, Istanbul, 2014, pp. 490–495.

[11] Q. Zhang, A. Liu, X. Pan and K. Pan, “CRC Code Design for List Decoding of Polar Codes,” IEEE Commun. Lett., vol. 21, no. 6, pp. 1229–1232, June 2017.

[12] E. Arıkan, “Serially Concatenated Polar Codes,” IEEE Access, vol. 6, pp. 64549–64555, 2018.

[13] G. Ricciutelli, T. Jerkovits, M. Baldi, F. Chiaraluce and G. Liva, “Analysis of the Block Error Probability of Concatenated Polar Code Ensembles,” IEEE Trans. Commun., vol. 67, no. 9, pp. 5953–5962, Sep. 2019.

[14] S. Kahraman and M. E. Celebi, “Code based efficient maximum-likelihood decoding of short polar codes;” in Proc. 2012 IEEE Int. Symp. Inf. Theory, Cambridge, MA, pp. 1967–1971, 2012.

[15] K. Niu, K. Chen and J. Lin, “Low-Complexity Sphere Decoding of Polar Codes Based on Optimum Path Metric,” IEEE Commun. Lett., vol. 18, no. 2, pp. 332–335, Feb. 2014.

[16] J. Guo and A. Guillen Fbregas, “Efficient sphere decoding of polar codes,” in Proc. 2015 IEEE Int. Symp. Inf. Theory, Hong Kong, pp. 236–240, 2015.

[17] J. Piao, J. Dai and K. Niu, “CRC-Aided Sphere Decoding for Short Polar Codes,” IEEE Commun. Lett., vol. 23, no. 2, pp. 210–213, Feb. 2019.

[18] S. Ejaz, F. Yang, T. H. Soliman, “Network polar coded cooperation with joint sc decoding”, Electron. Lett., vol. 51, no. 9, pp. 695–697, 2015.

[19] C. Schür, “A partial order for the synthesized channels of a polar code,” in Proc. 2016 IEEE Int. Symp. Inf. Theory, Barcelona, pp. 220–224, 2016.

[20] J. Dai, K. Niu, Z. Si, C. Dong and J. Lin, “Does Gaussian Approximation Work Well for the Long-Length Polar Code Construction?”, IEEE Access, vol. 5, pp. 7950–7963, 2017.

[21] G. He et al., “Beta-Expansion: A Theoretical Framework for Fast and Recursive Construction of Polar Codes,” in Proc. 2017 IEEE Global Communications Conference, Singapore, pp. 1–6, 2017.

[22] P. Koopman and T. Chakravarty, “Cyclic redundancy code (CRC) polynomial selection for embedded networks,” in International Conference on Dependable Systems and Networks, Florence, Italy, 2004, pp. 145–154, 2004.