

# Sphere decoder for polar codes concatenated with cyclic redundancy check

Yongrun YU, Zhiwen PAN\*, Nan LIU & Xiaosi TAN

*National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China*

Received 2 July 2018/Revised 15 September 2018/Accepted 6 December 2018/Published online 30 May 2019

**Abstract** The existing cyclic redundancy check (CRC)-aided successive cancellation list (CA-SCL) decoder partitions the decoding process into two steps, where an SCL is followed by a CRC check. An SCL decoder can approach the maximum-likelihood (ML) decoding performance of the inner polar codes using a sufficiently large list; however, in this case, CRC is only used for performing error detection. Therefore, the decoding performance of the outer CRC is different from that of ML because the errors are not rectified, which degrades the performance of the entire concatenated codes. In this study, we propose a sphere decoder (SD) that can achieve the ML performance of polar codes concatenated with CRC to address the suboptimality of CA-SCL decoding. The proposed SD performs joint decoding of the CRC-polar codes in a single step, thereby avoiding the polar decoding and CRC detection decoding scheme. Because the proposed SD guarantees the ML decoding performance of the CRC-polar concatenated codes, the simulation results demonstrate that the block error rate (BLER) of the proposed SD acts as the lower bound of the CA-SCL decoding performance. Further, a new initial radius selection method is proposed for the SD to reduce the average decoding complexity. The simulations demonstrate that the proposed initial radius selection method reduces more amount of decoding complexity when compared with that reduced using sequential step size methods.

**Keywords** polar codes, sphere decoder, maximum-likelihood decoding, optimal decoding, radius search

**Citation** Yu Y R, Pan Z W, Liu N, et al. Sphere decoder for polar codes concatenated with cyclic redundancy check. *Sci China Inf Sci*, 2019, 62(8): 082303, <https://doi.org/10.1007/s11432-018-9743-0>

## 1 Introduction

Polar codes are the first error control code class that can achieve symmetric memoryless channel capacity using a successive-cancellation (SC) decoder having a decoding complexity of  $O(N\log N)$  [1], where  $N$  denotes the codeword length. By employing a cyclic redundancy check (CRC)-aided successive cancellation list (CA-SCL) decoder [2,3], the block error rate (BLER) performance of the polar codes improves significantly when compared to that of an SC decoder and can be compared with the state-of-art low-density parity check (LDPC) codes [4]. Further, low-latency SCL decoding algorithms [5,6] are proposed for improving the throughput of the polar decoder. An SCL decoder can approach the maximum likelihood (ML) decoding performance of the inner polar codes using CRC-polar concatenated codes and a sufficiently large list [2]. However, in this case CRC is only used for error detection. Despite the high rate of the CRC, the CRC is a shortened cyclic code [7] (Subsection 3.2) that can rectify several error bits; however, the error correction ability of the CRC is weak due to the high rate of the CRC when CRC is used in modern digital systems. The representative examples are provided in Table 1, where the CRC codeword weight distributions are also presented. Two CRC codes have minimum Hamming

\* Corresponding author (email: pzw@seu.edu.com)

**Table 1** CRC codeword weight distributions<sup>a)b)c)</sup>

| $g(x)$                                  | $d_{\min}$ | $d_{\min 2}$ | $d_{\min 3}$ | $A_{d_{\min}}$ | $A_{d_{\min 2}}$ | $A_{d_{\min 3}}$ |
|-----------------------------------------|------------|--------------|--------------|----------------|------------------|------------------|
| $x^6 + x + 1$                           | 3          | 4            | 5            | 53             | 329              | 1541             |
| $x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$ | 3          | 4            | 5            | 26             | 347              | 2673             |

a) The information lengths of 6- and 8-bit CRCs are  $K = 22$  and  $K = 32$ , respectively.

b)  $d_{\min}$ ,  $d_{\min 2}$ , and  $d_{\min 3}$  denote MHW, second MHW, and third MHW, respectively.

c)  $A_{d_{\min}}$ ,  $A_{d_{\min 2}}$ , and  $A_{d_{\min 3}}$  represent the number of codewords of weights  $d_{\min}$ ,  $d_{\min 2}$ , and  $d_{\min 3}$ , respectively.

weight (MHW)  $d_{\min} = 3$ ; thus, they can rectify a minimum of one error when a hard-decision method is employed. A detailed description of the structure of a CRC codeword can be found in [8, 9]. However, the error correction ability of CRC cannot contribute to the CA-SCL scheme after SCL has produced  $L$  candidate codewords because SC decoding inherently demonstrates an error propagation phenomenon after the occurrence of an error. Thus, simply correcting several error bits is meaningless. Because an SCL decoder is suboptimal with a finite list size and because CRC detects but does not correct the errors, the BLER performance of the CRC-polar concatenated codes can be improved when an optimal decoder is employed. A sphere decoder that achieves suitable ML decoding performance and considers the error correction ability of the outer CRC codes can be used as an optimal decoder for the CRC-polar codes.

Sphere decoding for polar codes was initially introduced in [10]. A binary tree can be established for a sphere decoder by exploring the lower triangular structure of the generator matrix of polar codes. A parallel sphere decoding scheme that can reduce the decoding latency to half of that of the original scheme has been proposed in [10]. In addition, an optimal path metric has been proposed to determine the most promising current decoding path [11], where the decoding complexity can be reduced by assigning the highest priority to the most promising path. Fixed and dynamic lower bounds of a distance metric have also been proposed [12] for reducing the size of the current decoding search space by pruning the unlikely candidate paths. Other schemes [12] reduce the sphere decoding complexity without degrading the ML performance. Further, list sphere decoding has been introduced [13, 14] to address the fact that sphere decoding exhibits unfixed complexity. Using the path metric [13] and a predetermined list size, the list sphere decoder can guarantee fixed time complexity [13]. A matrix reordering strategy that further reduces the complexity of the list sphere decoder [13] has been proposed [14]. In addition, an early terminating criterion has been proposed for a polar sphere decoder [15]. This early terminating criterion performs well when the signal-to-noise ratio (SNR) is high, i.e., greater than 5 dB. In addition, an initial decoding radius selection algorithm has also been proposed [15]. However, the aforementioned studies have only dealt with single polar codes using a sphere decoder. To the best of our knowledge, no study has considered a sphere decoder for CRC-polar concatenated codes.

The existing sphere decoder for a single polar code relies on the lower triangular structure of a polar code generator matrix. However, to achieve the ML decoding performance of CRC-polar concatenated codes, the conventional two-step decoding scheme that initially decodes the polar codes and that subsequently performs CRC detection should be avoided. It is expected that a different generator matrix structure will be required for decoding the CRC-polar concatenated codes. In this study, a nonsystematic CRC is employed to obtain a generator matrix with a stair structure (defined in Subsection 3.2). We calculate the generator matrix  $\mathbf{G}$  of the CRC-polar codes and use the stair structure of  $\mathbf{G}$  for implementing sphere decoding. The existing CA-SCL decoding comprises two steps, i.e., SCL is followed by CRC [3], where CRC is only used for error detection. However, the proposed sphere decoder performs joint decoding of the CRC-polar codes. Decoding of the CRC-polar codes is completed in a single step, and the code structure, i.e., the generator matrix, of the CRC-polar concatenated code is completely explored. In addition, a new initial radius selection method is proposed for reducing the decoding complexity because the selection of an initial decoding radius is important in sphere decoding. The simulation results demonstrate that the BLER performance of the proposed decoder acts as the lower bound of the CA-SCL decoding performance; further, when compared to the existing methods, the proposed initial radius selection algorithm can reduce the decoding complexity.

This remainder of this study is organized as follows. In Section 2, preliminaries relative to polar codes

and a sphere decoder are introduced. The proposed sphere decoder is described in Section 3. In Section 4, our initial radius selection method is proposed. The simulation results are provided in Section 5, and the conclusion is provided in Section 6.

## 2 Preliminaries

### 2.1 Polar codes

Polar codes are linear block codes with the generator matrix  $\mathbf{G}_p = \mathbf{F}^{\otimes n}$ , where  $\mathbf{G}_p$  is an  $N \times N$  matrix and  $N$  denotes the codeword length,  $n = \log_2 N$ .  $\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ . The encoding process of polar codes can be expressed as follows:

$$\mathbf{x}_1^N = \tilde{\mathbf{u}}_1^N \mathbf{G}_p, \quad (1)$$

where  $\mathbf{x}_1^N = (x_1, \dots, x_N)$  denotes the coded bits and  $\tilde{\mathbf{u}}_1^N = (\tilde{u}_1, \dots, \tilde{u}_N)$  denotes the source bit sequence. Note that  $\tilde{\mathbf{u}}_1^N$  includes both information bits and frozen bits.

In symmetric memoryless channels, the value of frozen bits does not influence the BLER performance of polar codes [1]; therefore, for simplicity, the frozen bits are set to zero bits. Here, let  $\mathcal{A}$  denote the index set of information bits.  $\mathbf{G}_{\mathcal{A},p}$  consists of the rows in  $\mathbf{G}_p$  indicated by the set  $\mathcal{A}$ . Further, the encoding process can be rewritten as follows:

$$\mathbf{x}_1^N = \mathbf{u}_1^K \mathbf{G}_{\mathcal{A},p}, \quad (2)$$

where  $\mathbf{u}_1^K = (u_1, \dots, u_K)$  only represents  $K$  information bits. Note that the selection of  $\mathcal{A}$  is referred to as polar code construction and is beyond the scope of this study. The readers who are interested in this topic can refer to [16–18].

### 2.2 Sphere decoding

Throughout this study, we assume an additive white Gaussian noise (AWGN) channel and a binary-phase shift keying (BPSK) modulation scheme. Therefore, the sphere decoding of polar codes can be interpreted as the following optimization problem:

$$\hat{\mathbf{u}}_1^N = \arg \min_{\tilde{\mathbf{u}}_1^N \in \{0,1\}^N} \|\mathbf{y}_1^N - \mathbf{s}_1^N\|_2^2 = \arg \min_{\tilde{\mathbf{u}}_1^N \in \{0,1\}^N} \|\mathbf{y}_1^N - (\mathbf{1}_1^N - 2\tilde{\mathbf{u}}_1^N \mathbf{G}_p)\|_2^2 = \arg \min_{\tilde{\mathbf{u}}_1^N \in \{0,1\}^N} \left\| \frac{\mathbf{1}_1^N - \mathbf{y}_1^N}{2} - \tilde{\mathbf{u}}_1^N \mathbf{G}_p \right\|_2^2, \quad (3)$$

where  $\hat{\mathbf{u}}_1^N = (\hat{u}_1, \dots, \hat{u}_N)$  denotes the decoding output and  $\mathbf{y}_1^N = (y_1, \dots, y_N)$  represents the received signal. Further,  $\mathbf{s}_1^N = (s_1, \dots, s_N)$  denotes the BPSK modulation symbol, and  $\mathbf{1}_1^N$  is an all-one vector of length  $N$ . Vector-matrix multiplication is performed on the binary Galois field (GF(2)), and the remaining computations are performed on a real number field.  $\mathbf{y}_1^N$  is considered to be the sphere center, and the distance between  $\mathbf{y}_1^N$  and  $\mathbf{s}_1^N$  is considered to be the radius. Further, the objective of sphere decoding is to search among all  $\tilde{\mathbf{u}}_1^N$  to find the one exhibiting the minimum Euclidean distance to  $\mathbf{y}_1^N$ .

Here,  $\mathbf{G}_p$  denotes a lower triangular matrix. By utilizing the structure of  $\mathbf{G}_p$ , the problem (3) can be represented as a depth-first binary tree search with a pruning strategy. The sphere decoder is assigned an initial radius  $r_0 = +\infty$  and begins by guessing the value of  $\tilde{u}_N, \tilde{u}_{N-1}, \tilde{u}_{N-2}, \dots$ , and  $\tilde{u}_1$ . When  $\tilde{u}_i, 1 \leq i \leq N$  is obtained, the distance  $d_i$  between  $\mathbf{y}_i^N$  and  $\tilde{\mathbf{u}}_i^N$  is compared to the current sphere radius  $r$ . If  $d_i < r$ , the current result  $\tilde{\mathbf{u}}_i^N$  is considered to be promising; further, decoding is continued. If  $d_i \geq r$ , the current search at level  $i$  terminates, and the value of  $\tilde{u}_{i+1}$  is flipped. This process continues until  $\tilde{u}_1$  is obtained. The distance  $d_1$  between  $\mathbf{y}_1^N$  and  $\tilde{\mathbf{u}}_1^N$  is compared to the current sphere radius  $r$ . If  $d_1 < r$ , the current radius is updated to  $r = d_1$ ; further, the current optimal estimate is updated to  $\hat{\mathbf{u}}_1^N = \tilde{\mathbf{u}}_1^N$ . If  $d_1 \geq r$ , no update is performed; further, the decoder proceeds to flip bits at high levels. The sphere decoding terminates when no codeword has a smaller distance to  $\mathbf{y}_1^N$  than the current radius  $r$ . Further, the bit sequence  $\hat{\mathbf{u}}_1^N$  corresponding to the current radius  $r$  is selected as the decoding output. Note that sphere decoding achieves ML performance because the search space includes all  $2^K$  codewords.

### 3 Proposed sphere decoder

In this section, we describe the proposed decoder for the CRC-polar concatenated codes. Specifically, the generator matrix  $\mathbf{G}$  of the CRC-polar code is calculated, and the structure of  $\mathbf{G}$  is analyzed. The proposed sphere decoder relies on the special structure of  $\mathbf{G}$ . To facilitate clear understanding, examples are provided in the following subsections.

#### 3.1 Calculation of the CRC generator matrix

Because CRC is a shortened cyclic code, it can be characterized by a generator polynomial  $g(x) = x^r + a_{r-1}x^{r-1} + \dots + a_1x + 1$ , where  $r$  denotes the length of the CRC check bits and  $a_i \in \{0, 1\}$ ,  $1 \leq i \leq r-1$ . Note that  $g(x)$  can be described by vector  $\mathbf{v}(g(x))$  as follows:

$$\mathbf{v}(g(x)) = (1, a_{r-1}, \dots, a_1, 1). \quad (4)$$

Here, assume that  $K$  information bits must be attached to  $r$  bits of CRC. The last row (i.e., the  $K$ -th row) of the CRC generator matrix  $\mathbf{G}_{\text{CRC}}$  can be given as follows:

$$\mathbf{G}_{\text{CRC}}^{(K)} = (0_1, \dots, 0_{K-1}, \mathbf{v}(g(x))) = (0_1, \dots, 0_{K-1}, 1, a_{r-1}, \dots, a_1, 1), \quad (5)$$

where  $0_i$  denotes the  $i$ -th zero bit placed before  $\mathbf{v}(g(x))$ . The remaining rows of  $\mathbf{G}_{\text{CRC}}$  can be obtained by the left circulating shift of  $\mathbf{G}_{\text{CRC}}^{(K)}$ <sup>1)</sup>. An example of  $\mathbf{G}_{\text{CRC}}$  with  $g(x) = x^3 + x + 1$  [19], i.e.,  $\mathbf{v}(g(x)) = (1011)$ , and  $K = 3$  is provided in the following equation:

$$\mathbf{G}_{\text{CRC}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (6)$$

Note that  $\mathbf{G}_{\text{CRC}}$  in (6) is not in systematic form. Note that CRC is typically used as a systematic code in digital communication systems; however, in the proposed sphere decoder, the systematic form of  $\mathbf{G}_{\text{CRC}}$  is not required. The only difference between systematic and nonsystematic forms is the mapping between the source bit sequence and coded word, i.e., systematic and nonsystematic forms represent the same code; thus, there is no difference in the BLER.

We use a nonsystematic CRC generator matrix because the nonsystematic form can be partitioned as follows:

$$\mathbf{G}_{\text{CRC}} = [\mathbf{U}_{K \times r} | \mathbf{D}_{K \times K}], \quad (7)$$

where  $\mathbf{D}_{K \times K}$  is an invertible lower triangular matrix that can be obtained naturally by the left circulating shift of  $\mathbf{v}(g(x))$ . The characteristic of  $\mathbf{D}_{K \times K}$  is used to obtain a CRC-polar code generator matrix that can be directly used in the proposed sphere decoding method.

#### 3.2 Calculation of the CRC-polar generator matrix

To obtain the CRC-polar code generator matrix, polar codes with  $K + r$  unfrozen bits should be constructed, where  $K$  denotes the length of the data and  $r$  denotes the length of the CRC bits. The index set of  $K + r$  unfrozen bits is denoted as  $\mathcal{A}$ . In  $\mathbf{G}_p = \mathbf{F}^{\otimes n}$ , further, the rows that correspond to the indices in  $\mathcal{A}$  are selected to form  $\mathbf{G}_{\mathcal{A},p}$ , where  $\mathbf{G}_{\mathcal{A},p}$  has  $K + r$  rows and  $N$  columns. The generator matrix  $\mathbf{G}$  of the CRC-polar concatenated code can be obtained as follows:

$$\mathbf{G} = \mathbf{G}_{\text{CRC}} \mathbf{G}_{\mathcal{A},p}. \quad (8)$$

Examples of  $\mathbf{G}$  are provided in (9) and (10). Here,  $\mathbf{G}_{\text{CRC}}$  is the same as that in a previous study (6), and polar codes have length  $N = 8$ .

1) This process is equivalent to using  $K$  linearly independent CRC codewords to form the generator matrix.

$$\mathbf{G}_{\mathcal{A},p} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (9)$$

$$\mathbf{G} = \mathbf{G}_{\text{CRC}} \mathbf{G}_{\mathcal{A},p} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \quad (10)$$

A generator matrix with a stair structure is used in the proposed sphere decoder. A stair matrix can be defined as follows.

**Definition 1.** Matrix  $\mathbf{A}$  is a stair matrix with the following characteristics.

$\mathbf{A}$  is an  $m \times t$  full row rank matrix. The last nonzero element in the  $i$ -th ( $1 \leq i \leq m-1$ ) row is in the  $p$ -th ( $1 \leq p \leq t-1$ ) column, and the last nonzero element in the  $(i+1)$ -th row is in the  $q$ -th ( $2 \leq q \leq t$ ) column. If  $q > p$  for any  $1 \leq i \leq m-1$ ,  $\mathbf{A}$  is a stair matrix with  $m$  stairs.

**Example 1.** For matrices  $\mathbf{A}$  and  $\mathbf{B}$  that are shown below,  $\mathbf{A}$  is a stair matrix with four stairs, and  $\mathbf{B}$  is not a stair matrix even though it can be converted into  $\mathbf{A}$  by adding the third row to the second row. Note that an invertible lower triangular matrix will always be a stair matrix.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \quad (11)$$

As can be observed,  $\mathbf{G}_{\mathcal{A},p}$  in (9) is naturally a stair matrix because  $\mathbf{G}_{\mathcal{A},p}$  can be obtained by deleting the rows of  $\mathbf{G}_p$ , which is an invertible lower triangular matrix. In addition,  $\mathbf{G} = \mathbf{G}_{\text{CRC}} \mathbf{G}_{\mathcal{A},p}$  in (10) is also a stair matrix. Further, we prove that such a case is not a coincidence.

**Lemma 1.** If the nonsystematic CRC generator matrix  $\mathbf{G}_{\text{CRC}} = [\mathbf{U}_{K \times r} | \mathbf{D}_{K \times K}]$  is used,  $\mathbf{G} = \mathbf{G}_{\text{CRC}} \mathbf{G}_{\mathcal{A},p}$  is a stair matrix.

*Proof.* Partition  $\mathbf{G}_{\text{CRC}}$  into two sub-matrices  $\mathbf{G}_{\text{CRC}} = [\mathbf{U}_{K \times r} | \mathbf{D}_{K \times K}]$ , where  $\mathbf{D}_{K \times K}$  is a lower triangular matrix.  $\mathbf{D}_{K \times K}$  is invertible, i.e., all the elements in the main diagonal of  $\mathbf{D}_{K \times K}$  are one. Partition  $\mathbf{G}_{\mathcal{A},p}$  into two sub-matrices as follows:

$$\mathbf{G}_{\mathcal{A},p} = \begin{bmatrix} \mathbf{G}_{1,r \times N} \\ \mathbf{G}_{2,K \times N} \end{bmatrix}. \quad (12)$$

Further, we obtain the following:

$$\mathbf{G} = \mathbf{G}_{\text{CRC}} \mathbf{G}_{\mathcal{A},p} = [\mathbf{U}_{K \times r} | \mathbf{D}_{K \times K}] \begin{bmatrix} \mathbf{G}_{1,r \times N} \\ \mathbf{G}_{2,K \times N} \end{bmatrix} = \mathbf{U} \mathbf{G}_1 + \mathbf{D} \mathbf{G}_2. \quad (13)$$

Prior to analyzing the structure of  $\mathbf{U} \mathbf{G}_1$  and  $\mathbf{D} \mathbf{G}_2$ , we define the following helper function  $\gamma$ .  $\gamma(i)$ ,  $1 \leq i \leq K+r$ , denotes the column number of the last nonzero element in the  $i$ -th row of  $\mathbf{G}_{\mathcal{A},p}$ . For example, in (9),  $(\gamma(1), \dots, \gamma(6)) = (2, 3, 4, 6, 7, 8)$ . First, we analyze the structure of  $\mathbf{U} \mathbf{G}_1$ .

Recall that  $\mathbf{G}_{\mathcal{A},p}$  is a stair matrix and that the last nonzero element in the first row of  $\mathbf{G}_{2,K \times N}$  is in the  $\gamma(r+1)$ -th column, which indicates that the columns from  $\gamma(r+1)$  to  $N$  are all zero column vectors of length  $r$  in  $\mathbf{G}_{1,r \times N}$ . Therefore,  $\mathbf{G}_{1,r \times N}$  can be partitioned as follows:

$$\mathbf{G}_{1,r \times N} = [\tilde{\mathbf{G}}_{1,r \times (\gamma(r+1)-1)} | \mathbf{O}_{r \times (N-\gamma(r+1)+1)}]. \quad (14)$$

Further,  $UG_1$  takes the following form:

$$\begin{aligned} UG_1 &= U[\tilde{G}_{1,r \times (\gamma(r+1)-1)} | \mathbf{O}_{r \times (N-\gamma(r+1)+1)}] \\ &= [U_{K \times r} \tilde{G}_{1,r \times (\gamma(r+1)-1)} | U_{K \times r} \mathbf{O}_{r \times (N-\gamma(r+1)+1)}] \\ &= [U_{K \times r} \tilde{G}_{1,r \times (\gamma(r+1)-1)} | \mathbf{O}_{K \times (N-\gamma(r+1)+1)}]. \end{aligned} \quad (15)$$

Next, we analyze the structure of  $DG_2$ . According to Definition 1,  $G_{2,K \times N}$  is a stair matrix because  $G_{2,K \times N}$  can be obtained by extracting the last  $K$  rows from  $G_{A,p}$  ( $G_{A,p}$  is a stair matrix as defined in Definition 1). The following equation always holds ( $\tilde{1}$  indicates that  $\tilde{1}$  is in the  $\gamma(r+1)$ -th column):

$$\begin{aligned} DG_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ * & 1 & \ddots & 0 \\ * & \ddots & 1 & 0 \\ * & \cdots & * & 1 \end{bmatrix}_{K \times K} \begin{bmatrix} * \cdots * & \tilde{1} & 0 & 0 & 0 & 0 & 0 \\ * \cdots * & * & * & * & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & * & * & * & * & 1 & 0 \\ * \cdots * & * & * & * & * & * & * & * & 1 \end{bmatrix}_{K \times N} \stackrel{(a)}{=} \begin{bmatrix} * \cdots * & \tilde{1} & 0 & 0 & 0 & 0 & 0 \\ * \cdots * & * & * & * & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & * & * & * & * & * & 1 & 0 \\ * \cdots * & * & * & * & * & * & * & * & 1 \end{bmatrix}_{K \times N} \\ &= \tilde{G}_2 \stackrel{(b)}{=} [\tilde{G}_{2,\text{left}} | \tilde{G}_{2,\text{right}}]. \end{aligned} \quad (16)$$

Thus,  $G_2$  and  $\tilde{G}_2$  have identical stair structures. Equation (a) in (16) can be easily checked by calculating all the elements in  $\tilde{G}_2$  sequentially. The matrix partition (b) produces  $\tilde{G}_{2,\text{left}}$  and  $\tilde{G}_{2,\text{right}}$ .  $\tilde{G}_{2,\text{left}}$  is a  $K \times (\gamma(r+1)-1)$  matrix, and  $\tilde{G}_{2,\text{right}}$  is a  $K \times (N-\gamma(r+1)+1)$  matrix. With this partition,  $\tilde{G}_{2,\text{right}}$  is a stair matrix with  $K$  stairs. Further, we can obtain the following:

$$\begin{aligned} G &= G_{\text{CRC}} G_{A,p} = [U_{K \times r} | D_{K \times K}] \begin{bmatrix} G_{1,r \times N} \\ G_{2,K \times N} \end{bmatrix} = UG_1 + DG_2 \\ &= [U_{K \times r} \tilde{G}_{1,r \times (\gamma(r+1)-1)} | \mathbf{O}_{K \times (N-\gamma(r+1)+1)}] + [\tilde{G}_{2,\text{left}} | \tilde{G}_{2,\text{right}}] \\ &= [U_{K \times r} \tilde{G}_{1,r \times (\gamma(r+1)-1)} + \tilde{G}_{2,\text{left}} | \tilde{G}_{2,\text{right}}], \end{aligned} \quad (17)$$

where  $\tilde{G}_{2,\text{right}}$  is a stair matrix with  $K$  stairs; thus,  $G$  is also a stair matrix with  $K$  stairs because  $\tilde{G}_{2,\text{right}}$  is placed at the right partition of  $G$ . Further,  $G$  can be used in the proposed sphere decoder.

### 3.3 CRC-polar sphere decoder

In this subsection, the decoding process of the proposed sphere decoder is described. We begin by defining an index matrix  $P$  to characterize the stair structure of the  $K \times N$  matrix  $G$ .

**Definition 2.** A  $K \times 2$  matrix  $P$  that describes the stair structure of  $G$  is referred to as an index matrix if the elements in  $P$  are obtained through  $\rho(i)$  as follows:

$$P_{i,1} = \rho(i-1) + 1, \quad P_{i,2} = \rho(i), \quad 1 \leq i \leq K, \quad (18)$$

where the value of  $\rho(i)$ ,  $1 \leq i \leq K$  is the column number of the last nonzero element in the  $i$ -th row of  $G$  ( $\rho(0) = 0$ ). For example,  $P$  corresponding to  $G$  in (10) can be given as follows:

$$P = \begin{bmatrix} 1 & 6 \\ 7 & 7 \\ 8 & 8 \end{bmatrix}. \quad (19)$$

A simple example of the proposed sphere decoder with  $P$  is given in the following. This example can be easily extended to a general form, which will be discussed later.

**Example 2.** An example of the proposed sphere decoder for CRC-polar codes is shown as follows with data length  $K = 3$ , CRC length  $r = 3$ , and polar code length  $N = 8$ .

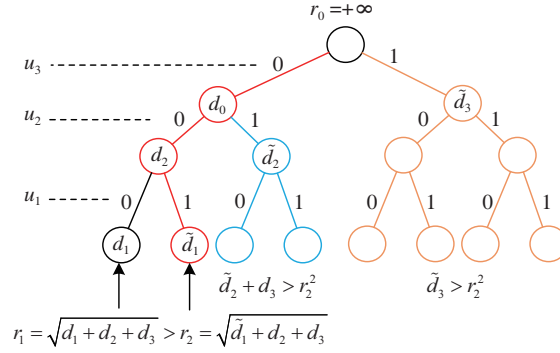


Figure 1 (Color online) Sphere decoding process.

The corresponding generator matrices of CRC ( $\mathbf{G}_{\text{CRC}}$ ), polar code ( $\mathbf{G}_{A,p}$ ), and CRC-polar concatenated code ( $\mathbf{G} = \mathbf{G}_{\text{CRC}}\mathbf{G}_{A,p}$ ) are already presented in (6), (9), and (10), respectively. Here, we denote the signal received from the AWGN channel as  $\mathbf{y}_1^8 = (y_1, \dots, y_8)$ . For simplicity, we express the received signal as  $\tilde{\mathbf{y}}_1^8 = (\tilde{y}_1, \dots, \tilde{y}_8) = (\frac{1-y_1}{2}, \dots, \frac{1-y_8}{2})$ .

According to the stair structure of  $\mathbf{G}$  in (10) and with the help of  $\mathbf{P}$ , the segmented distance metrics  $d_1$ ,  $d_2$ , and  $d_3$  are calculated as follows ( $\sum_{i=1}^3 d_i$  equals the square of Euclidean norm in (3)):

$$d_3 = \sum_{j=P_{3,1}}^{P_{3,2}} \left( \tilde{y}_j - \sum_{l=3}^3 u_l G_{l,j} \right)^2 = (\tilde{y}_8 - u_3)^2, \quad (20)$$

$$d_2 = \sum_{j=P_{2,1}}^{P_{2,2}} \left( \tilde{y}_j - \sum_{l=2}^3 u_l G_{l,j} \right)^2 = (\tilde{y}_7 - u_2)^2, \quad (21)$$

$$d_1 = \sum_{j=P_{1,1}}^{P_{1,2}} \left( \tilde{y}_j - \sum_{l=1}^3 u_l G_{l,j} \right)^2 = (\tilde{y}_6 - u_1 \oplus u_2 \oplus u_3)^2 + (\tilde{y}_5 - u_1)^2 + (\tilde{y}_4 - u_1)^2 + (\tilde{y}_3 - u_1 \oplus u_3)^2 \\ + (\tilde{y}_2 - u_1 \oplus u_2)^2 + (\tilde{y}_1 - u_1 \oplus u_2 \oplus u_3)^2, \quad (22)$$

where  $u_l$  denotes the  $l$ -th information bit and  $G_{l,j}$  denotes the element of  $\mathbf{G}$  in the  $l$ -th row and  $j$ -th column.

It can be observed that  $u_3$  appears in  $d_3$  for the first time.  $u_2$  and  $u_1$  appear for the first time in  $d_2$  and  $d_1$ , respectively. This implies that the sphere decoder can be implemented via a binary tree search, as shown in Figure 1. Therefore, we can obtain generator matrix  $\mathbf{G}$  with a stair structure.

As depicted in Figure 1, the initial decoding radius  $r_0$  is  $+\infty$ . Here,  $(u_3, u_2, u_1) = (0, 0, 0)$  is initially obtained as the currently optimal estimate, and  $r_1 = \sqrt{d_1 + d_2 + d_3}$  is used to update the current sphere radius. Further,  $u_1$  is flipped to 1, and  $(u_3, u_2, u_1) = (0, 0, 1)$  is obtained.  $r_2 = \sqrt{\tilde{d}_1 + d_2 + d_3}$  is then compared to  $r_1$ , where  $\tilde{d}_1$  denotes the recalculation of (22) after  $u_1$  is flipped. The result is  $r_1 > r_2$ ; thus,  $r_2$  is selected as the current radius, and  $(u_3, u_2, u_1) = (0, 0, 1)$  is saved as the currently optimal estimate (corresponding to the red line in Figure 1). Further,  $u_2$  is flipped to 1; therefore,  $(u_3, u_2) = (0, 1)$  is obtained. Subsequently, the decoder finds that  $\sqrt{\tilde{d}_2 + d_3} > r_2$ , where  $\tilde{d}_2$  denotes the recalculation of (21) after  $u_2$  is flipped. Therefore, the branches with blue lines are pruned because they cannot be the ML estimates (the distance metric increases monotonically). Further,  $u_3$  is flipped to 1, and  $\tilde{d}_3$  is calculated. After comparison,  $\sqrt{\tilde{d}_3} > r_2$ ; thus, the branches with yellow lines are pruned. Finally, the currently optimal estimate  $(u_3, u_2, u_1) = (0, 0, 1)$  is selected as the decoding output.

The aforementioned process can be extended to a general form, i.e., in a general case, the binary tree in Figure 1 grows taller. Under the  $(N, K)$  CRC-polar codes, the height of the full binary tree is  $K$ . The distance metric at the  $i$ -th level (counting from the bottom) is obtained through the  $i$ -th row of the

index matrix  $\mathbf{P}$  and the corresponding columns in  $\mathbf{G}$ . A general sphere decoder for CRC-polar codes is described in Algorithms 1–3.

Algorithm 1 is the main function of the proposed decoding algorithm. In Algorithm 2, we employ a recursive process to describe the sphere decoding process, which is more flexible than pseudocodes [10]. Algorithm 3 is a sub-function that obeys the sphere decoding principle proposed in [20], i.e., the bit value (0 or 1) of  $u_i$  that results in a small distance metric  $d_i$  is first selected as an estimate of  $u_i$ .

In the remainder of this section, we discuss about the proposed decoder. Note that the proposed decoder performs joint decoding of CRC-polar concatenated codes and immediately estimates the data bits  $\mathbf{u}_1^K$  with optimal ML performance. The error correction capabilities of the polar and CRC codes are combined while using  $\mathbf{G} = \mathbf{G}_{\text{CRC}}\mathbf{G}_{\mathcal{A},p}$  to calculate the Euclidean distance between the estimated codewords and the received signal. We avoid the conventional scheme in which the polar codes are decoded first; further, CRC detection is performed; thus, the error correction capabilities of the polar and CRC codes are not reflected individually.

The CRC-polar code has better BLER than a single polar code because outer CRC improves the distance spectrum of polar codes, e.g., increased minimum distance [21]. If we completely explore such improvement afforded by the outer CRC, i.e., by achieving the ML decoding performance of concatenated codes, the error correction ability of CRC is reflected automatically. The proposed sphere decoder achieves the ML performance of the CRC-polar concatenated codes; therefore, we conclude that the error correction abilities of CRC and polar codes are completely utilized.

---

**Algorithm 1** Proposed sphere decoder, main function
 

---

**Input:**  $g(x), \mathbf{G}_{\mathcal{A},p}, \tilde{\mathbf{y}}_1^N$ .  
**Output:**  $\hat{\mathbf{u}}_{1,\text{output}}^K$ .  
 1: Calculate  $\mathbf{G}_{\text{CRC}}$  through  $g(x)$ ;  
 2:  $\mathbf{G} = \mathbf{G}_{\text{CRC}}\mathbf{G}_{\mathcal{A},p}$ ;  
 3: Obtain  $\mathbf{P}$  in Definition 2;  
 4:  $\hat{\mathbf{u}}_1^K \leftarrow$  null array; //Temporary bit estimate  
 5:  $\mathbf{d}_1^K \leftarrow$  null array; //Distance metric  
 6:  $\bar{\mathbf{d}}_1^K \leftarrow$  null array; //Auxiliary distance metric to avoid redundant calculations  
 7:  $r \leftarrow +\infty$ ; //Initial radius  
 8:  $\hat{\mathbf{u}}_{1,\text{output}}^K \leftarrow \text{SphereDecoder}(K, \mathbf{P}, \mathbf{G}, \tilde{\mathbf{y}}_1^N, \hat{\mathbf{u}}_1^K, \mathbf{d}_1^K, \bar{\mathbf{d}}_1^K, r)$ .

---



---

**Algorithm 2** SphereDecoder
 

---

**Input:**  $k, \mathbf{P}, \mathbf{G}, \tilde{\mathbf{y}}_1^N, \hat{\mathbf{u}}_1^K, \mathbf{d}_1^K, \bar{\mathbf{d}}_1^K, r$ .  
**Output:**  $\hat{\mathbf{u}}_{1,\text{current}}^K$  optimal.  
 1: **for**  $a \leftarrow 1 : 2$  **do**  
 2:   **if**  $a = 1$  **then**  
 3:      $[\hat{u}_k, d_k, \bar{d}_k] \leftarrow \text{SelectFirstBit}(k, \mathbf{P}, \mathbf{G}, \tilde{\mathbf{y}}_1^N, \hat{\mathbf{u}}_1^K)$ ;  
 4:   **else**  
 5:      $\hat{u}_k \leftarrow \hat{u}_k \oplus 1, d_k \leftarrow \bar{d}_k$ ;  
 6:   **end if**  
 7:   **if**  $\sum_{i=k}^K d_i > r^2$  **then**  
 8:     **continue** //Pruning operation  
 9:   **else**  
 10:     **if**  $k = 1$  **then**  
 11:        $r^2 \leftarrow \sum_{i=1}^K d_i$ ; //Update radius  
 12:        $\hat{\mathbf{u}}_{1,\text{current}}^K \leftarrow \hat{\mathbf{u}}_1^K$ ; //Update bit estimate  
 13:     **else**  
 14:        $\hat{\mathbf{u}}_{1,\text{current}}^K \leftarrow \text{SphereDecoder}(k-1, \mathbf{P}, \mathbf{G}, \tilde{\mathbf{y}}_1^N, \hat{\mathbf{u}}_1^K, \mathbf{d}_1^K, \bar{\mathbf{d}}_1^K, r)$ .  
 15:     **end if**  
 16:   **end if**  
 17: **end for**

---

**Algorithm 3** SelectFirstBit**Input:**  $k, P, G, \tilde{\mathbf{y}}_1^N, \hat{\mathbf{u}}_1^K$ .**Output:**  $\hat{u}_k, d_k, \bar{d}_k$ .

---

```

1:  $\hat{u}_k \leftarrow 0$ ;
2:  $d_{\text{tmp1}} \leftarrow \sum_{i=P_{k,1}}^{P_{k,2}} (\tilde{y}_i - \sum_{j=k}^K \hat{u}_j G_{j,i})^2$ ;
3:  $\hat{u}_k \leftarrow 1$ ;
4:  $d_{\text{tmp2}} \leftarrow \sum_{i=P_{k,1}}^{P_{k,2}} (\tilde{y}_i - \sum_{j=k}^K \hat{u}_j G_{j,i})^2$ ;
5: if  $d_{\text{tmp1}} > d_{\text{tmp2}}$  then
6:    $\hat{u}_k \leftarrow 1$ ;
7:    $d_k \leftarrow d_{\text{tmp2}}$ ;
8:    $\bar{d}_k \leftarrow d_{\text{tmp1}}$ ;
9: else
10:   $\hat{u}_k \leftarrow 0$ ;
11:   $d_k \leftarrow d_{\text{tmp1}}$ ;
12:   $\bar{d}_k \leftarrow d_{\text{tmp2}}$ ;
13: end if

```

---

## 4 Initial radius selection

Proper selection of the initial radius is important for reducing the complexity of sphere decoding [22, 23]. In Algorithm 1, the initial radius is simply set to  $+\infty$ , which may increase unnecessary binary tree searches. In this section, a new initial radius selection algorithm is proposed. First, we introduce various notations for the existing method. Further, we discuss the proposed initial radius selection algorithm.

Based on the distance lower bound method proposed in [12], the authors in [15] have introduced a progressive initial radius selection criterion. Upon receiving  $\tilde{\mathbf{y}}_1^N = (\tilde{y}_1, \dots, \tilde{y}_N) = (\frac{1-y_1}{2}, \dots, \frac{1-y_N}{2})$ , the minimum possible radius can be calculated as follows:

$$r_{\min}^2 = \sum_{i=1}^N d_{\min}(i), \quad (23)$$

where  $d_{\min}(i) = \min\{(\tilde{y}_i - 0)^2, (\tilde{y}_i - 1)^2\}$ . In this method, the coded bit  $x_i$  corresponding to  $\tilde{y}_i$  is either 0 or 1, and the bit value resulting in small  $(\tilde{y}_i - x_i)^2$  is selected to estimate the minimum possible distance between  $\tilde{\mathbf{y}}_1^N$  and  $\mathbf{x}_1^N$ . Because  $r_{\min}^2$  may not be achievable, the sphere decoding radius is set as follows [15]:

$$r_w^2 = r_{\min}^2 + w\alpha, \quad (24)$$

where  $w$  denotes the  $w$ -th sphere decoding and the initial radius is  $r_1^2 = r_{\min}^2 + \alpha$ . Eq. (24) implies that the radius will continue to increase until the ML codeword is included in the sphere once  $r_w^2$  is small and causes failure in finding the ML codeword, i.e., no codeword is included in radius  $r_w^2$ . Here, as proposed in [15], the typical  $\alpha$  value is 1, and  $\alpha$  is fixed as constant regardless of the remaining parameters such as SNR. However, the complexity of sphere decoding is dependent on SNR, i.e., high SNR results in low complexity [12]. Therefore, SNR should be considered while designing the initial radius.

The proposed initial radius selection method considers both  $r_{\min}^2$  in (23) and SNR. In the AWGN channel with BPSK, when codeword  $\mathbf{x}_1^N$  is transmitted, we can obtain the received signal  $\mathbf{y}_1^N$  as follows:

$$\mathbf{y}_1^N = (1 - 2\mathbf{x}_1^N) + \mathbf{n}_1^N, \quad (25)$$

where  $\mathbf{n}_1^N = (n_1, \dots, n_N)$  is the  $N$ -dimensional Gaussian additive noise with variance  $\sigma_{\text{AWGN}}^2$ . Here, let  $\tilde{\mathbf{y}}_1^N = (1 - \mathbf{y}_1^N)/2$ . Therefore,  $R_{\text{square}} = \|\tilde{\mathbf{y}}_1^N - \mathbf{x}_1^N\|^2 = \|\mathbf{n}_1^N/2\|^2$  is subject to Chi-square distribution. Because the codeword length  $N$  is even, the cumulative distribution function of  $R_{\text{square}}$  is given as follows:

$$F(R_{\text{square}}) = 1 - e^{-\frac{R_{\text{square}}}{2\sigma^2}} \sum_{k=0}^{N/2-1} \frac{1}{k!} \left( \frac{R_{\text{square}}}{2\sigma^2} \right)^k, \quad R_{\text{square}} > 0, \quad (26)$$

where  $\sigma = 0.5\sigma_{\text{AWGN}}$  because we have transformed  $\mathbf{y}_1^N$  into  $\tilde{\mathbf{y}}_1^N$ . After receiving  $\tilde{\mathbf{y}}_1^N$ , the minimum possible radius  $r_{\min}^2$  is obtained through (23), which implies that  $R_{\text{square}} \geq r_{\min}^2$ . Inspired by the increasing radius

search (IRS) [23], we set the following:

$$\Pr(R_{\text{square}} \leq \beta_0 | R_{\text{square}} \geq r_{\min}^2) = \gamma, \quad (27)$$

where  $\beta_0 \geq r_{\min}^2$ .

Eq. (27) indicates that the probability that the ML codeword is included in the radius value  $\beta_0$  is  $\gamma$  when  $r_{\min}^2$ . Using simple calculations,  $\beta_0$  can be obtained as follows ( $\beta_0$  is the zero point of (28)):

$$F(\beta_0) - \gamma - (1 - \gamma)F(r_{\min}^2) = 0. \quad (28)$$

Further,  $\beta_0$  is selected as the initial decoding radius. When  $\beta_0$  is small and leads to failure in finding the ML codeword, we replace  $r_{\min}^2$  in (28) with  $\beta_0$  to obtain a new  $\beta_1$  for the next decoding turn as follows:

$$F(\beta_1) - \gamma - (1 - \gamma)F(\beta_0) = 0. \quad (29)$$

Generally, once radius  $\beta_k$  fails to find the ML codeword, the next search radius  $\beta_{k+1}$  can be obtained as follows:

$$F(\beta_{k+1}) - \gamma - (1 - \gamma)F(\beta_k) = 0. \quad (30)$$

This process continues until the ML codeword is included in radius  $\beta_k$ . Note that  $\beta_{k+1} > \beta_k$  because  $F(\beta)$  is an increasing function in  $\beta$ .  $\beta_{k+1}$  in (30) can be calculated offline, and  $\gamma$  can be obtained via simulations. We observed that  $\gamma = 0.6$  is suitable for various configurations, and the corresponding simulation results are given in Section 5.

## 5 Simulation results

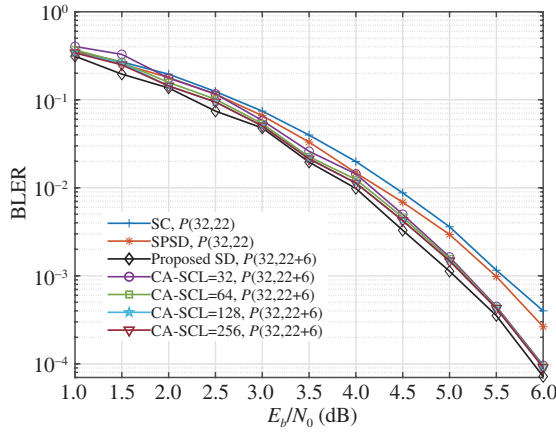
Here, we first demonstrate the BLER performance of the proposed sphere decoder. Further, we discuss the average decoding complexity with the proposed initial radius selection method.

### 5.1 BLER performance

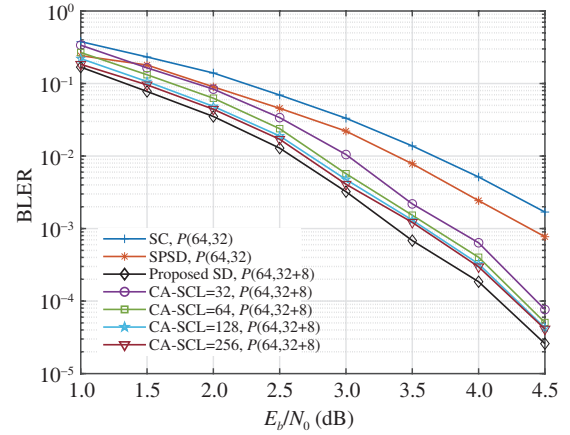
A sphere decoder is suitable for short codes; thus, here, we focus on short polar codes of lengths 32 and 64, as done in [10–12]. Here, the channel is AWGN with BPSK modulation, and  $E_b/N_0$  denotes the SNR in terms of the average energy of information bits. For a single polar code,  $R = K/N$ , where  $K$  is the number of information bits and  $N$  is the codeword length. For CRC-polar concatenated codes with  $K$  information bits and  $r$  CRC bits, the rate remains  $R = K/N$  because  $r$  CRC bits that are determined by  $K$  information bits do not carry information. Here, let  $P(N, K)$  denote the polar codes that carry  $K$  information bits of length  $N$ .  $P(N, K + r)$  represents the CRC-polar concatenated codes with  $K$  information bits,  $r$  CRC bits, and codeword length  $N$ . SPSD denotes a single polar code (SP) that is decoded by SD. Note that all the codes discussed in this section are constructed by Gaussian approximation [16] at  $E_b/N_0 = 6$  dB. For SC-based algorithms, min-sum approximation is employed for the log-likelihood ratio (LLR) update in a check node, and a hardware-friendly LLR-based path metric [24] is used in SCL decoding.

Here,  $P(32, 22)$  and  $P(32, 22 + 6)$  are simulated for polar codes of length 32. The generator polynomial of the 6-bit CRC is  $g(x) = x^6 + x + 1$ . Figure 2 depicts the BLER of  $P(32, 22)$  and  $P(32, 22 + 6)$ . As can be observed, the SC decoder and SPSD demonstrate the worst BLER performance because they do not concatenate with CRC. As the list size  $L$  increases from 32 to 256, the BLER of the CA-SCL decoder remains mostly unchanged, which indicates that  $L = 32$  is sufficient for  $P(32, 22 + 6)$ . The proposed sphere decoder yields the best BLER, achieving an ML decoding performance of  $P(32, 22 + 6)$ .

$P(64, 32)$  and  $P(64, 32 + 8)$  are simulated for polar codes of length 64. The generator polynomial of the 8-bit CRC is  $g(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$ . Figure 3 depicts the BLER of  $P(64, 32)$  and  $P(64, 32 + 8)$ . As shown, the SC decoder and SPSD still provide the worst BLER, and, as  $L$  increases, the BLER of CA-SCL continues to improve up to  $L = 256$ . Even when  $L = 256$  is used for the CA-SCL decoder, the BLER of the proposed sphere decoder still outperforms the CA-SCL decoder due to the ML performance of the proposed method.



**Figure 2** (Color online) BLER performance of  $P(32, 22)$  and  $P(32, 22 + 6)$ .



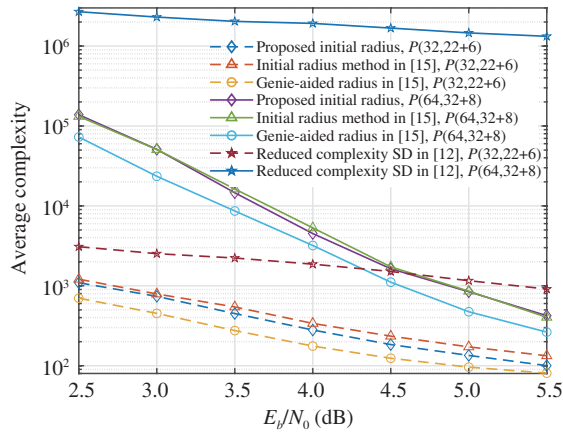
**Figure 3** (Color online) BLER performance of  $P(64, 32)$  and  $P(64, 32 + 8)$ .

## 5.2 Average complexity

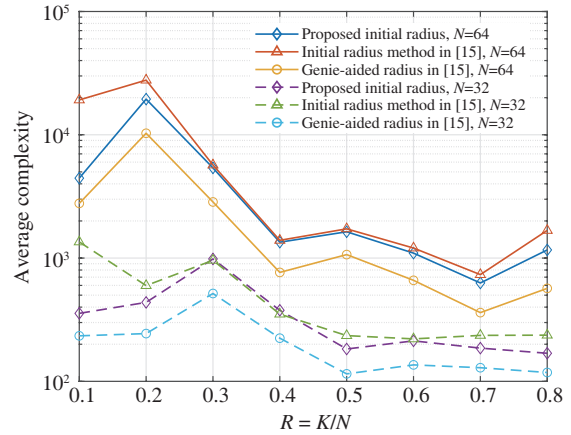
Here, we present the average complexity of the sphere decoder with the proposed initial radius selection algorithm. The computation of one  $(\tilde{y}_j - \sum u_l G_{l,j})^2$  assumes unit complexity because it is the basic computation in Algorithm 3. One such computation is also referred to as a visited node in binary tree search [12, 15]. Note that the fixed lower bound algorithm in [12] is employed in our simulations. In addition, in the proposed radius selection method, the  $\gamma$  parameter is fixed to 0.6 in all the simulations.

We compare the proposed radius selection method to the existing radius method [15], the genie-aided radius  $r_{\text{genie}}^2$  [15], and the fixed lower bound technique [12]. For any radius selection method, the sphere decoder with the genie-aided radius [15] yields the lower bound of the average complexity because it is an ideal radius. The genie-aided radius [15] indicates that the decoder already knows the distance between the ML codeword  $\mathbf{x}_{1,\text{ML}}^N$  and the received signal  $\tilde{\mathbf{y}}_1^N$ , i.e.,  $r_{\text{genie}}^2 = \|\tilde{\mathbf{y}}_1^N - \mathbf{x}_{1,\text{ML}}^N\|^2$ . In simulations, this can be realized by performing sphere decoding twice, i.e., the first decoding is employed to obtain  $r_{\text{genie}}^2 = \|\tilde{\mathbf{y}}_1^N - \mathbf{x}_{1,\text{ML}}^N\|^2$ , and the second decoding uses  $r_{\text{genie}}^2$  as the initial radius. The complexity of the second decoding is considered equal to that of the genie-aided radius. The fixed lower bound technique [12] does not involve a radius selection scheme; thus, the average complexity of this technique serves as a baseline that indicates the degree to which the complexity is reduced using the proposed radius selection scheme. Note that the proposed sphere decoder relies on the same binary search process in [10]. Therefore, the average complexity of the proposed sphere decoder is essentially the same as that of the common sphere decoder.

We denote the average complexity under different  $E_b/N_0$  and different rate  $R = K/N$  in Figures 4 and 5, respectively. Here, the number of iterations in the simulation is  $10^4$ . Under various configurations, the average complexity of the sphere decoder with the proposed initial radius is less than that of a previous method [15] and approaches the complexity of the genie-aided radius. Specifically, as depicted in Figure 4, for  $P(32, 22 + 6)$ , the proposed scheme exhibits advantages over the existing method [15] relative to a higher  $E_b/N_0$  regime ( $E_b/N_0 \geq 3.5$  dB). For  $P(64, 32 + 8)$ , the proposed scheme does not exhibit obvious advantages over the existing method [15]. This phenomenon may result from the fact that  $\alpha = 1$  in [15] is near-optimal for decoding  $P(64, 32 + 8)$ . Note that, compared to [12], the proposed radius selection scheme significantly reduces the average complexities by up to two orders of magnitude, which confirms the effectiveness of the proposed radius selection method. In Figure 5, for both  $P(32, 32R + 6)$  and  $P(64, 64R + 8)$ , the proposed scheme demonstrates advantages over the existing method [15] relative to the low ( $R \leq 0.3$ ) and high ( $R \geq 0.6$ ) rate regions, while the proposed and existing methods demonstrate similar average complexity in the medium rate region.



**Figure 4** (Color online) Average complexity comparison under different  $E_b/N_0$  (proposed radius scheme:  $\gamma = 0.6$ ; previous radius scheme [15]:  $\alpha = 1$ ).



**Figure 5** (Color online) Average complexity comparison under different  $R$  at  $E_b/N_0 = 4.5$  dB (proposed radius scheme:  $\gamma = 0.6$ ; previous radius scheme [15]:  $\alpha = 1$ ).

## 6 Conclusion

In this study, a sphere decoder has been proposed to decode the CRC-polar concatenated codes. The existing CA-SCL decoder divides the decoding process into two steps, where the SCL decoder is followed by a CRC check and where CRC is only used for error detection. The proposed sphere decoder finishes decoding in a single step and achieves the ML performance with complete exploration of the structure of the CRC-polar code. In addition, a new initial radius selection method was proposed to reduce the average decoding complexity. The simulation results confirm that the BLER performance of the proposed decoder acts as the lower bound of the CA-SCL decoding performance. In addition, the proposed radius selection method achieves a lower average complexity than that achieved using several existing methods. The sphere decoder for decoding longer CRC-polar codes will be studied in the further work.

**Acknowledgements** This work was partially supported by National Major Project (Grant No. 2017ZX03001002-004), National Natural Science Foundation Project (Grant No. 61521061), National Natural Science Foundation of China (Grant No. 61571123), and 333 Program of Jiangsu (Grant No. BRA2017366).

## References

- 1 Arikan E. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans Inform Theor*, 2009, 55: 3051–3073
- 2 Tal I, Vardy A. List decoding of polar codes. *IEEE Trans Inform Theor*, 2015, 61: 2213–2226
- 3 Niu K, Chen K. CRC-aided decoding of polar codes. *IEEE Commun Lett*, 2012, 16: 1668–1671
- 4 Tahir B, Schwarz S, Rupp M. BER comparison between convolutional, turbo, LDPC, and polar codes. In: *Proceedings of the 24th International Conference on Telecommunications (ICT)*, Limassol, 2017. 1–7
- 5 Cao C, Kuang J, Fei Z, et al. Low complexity list successive cancellation decoding of polar codes. *IET Commun*, 2014, 8: 3145–3149
- 6 Xu Q Y, Pan Z W, Liu N, et al. A complexity-reduced fast successive cancellation list decoder for polar codes. *Sci China Inf Sci*, 2018, 61: 022309
- 7 Ryan W E, Lin S. *Channel Codes Classical And Modern*. New York: Cambridge University Press Ltd., 2009. 110–111
- 8 Kazakov P. Fast calculation of the number of minimum-weight words of CRC codes. *IEEE Trans Inform Theor*, 2001, 47: 1190–1195
- 9 Castagnoli G, Brauer S, Herrmann M. Optimization of cyclic redundancy-check codes with 24 and 32 parity bits. *IEEE Trans Commun*, 1993, 41: 883–892
- 10 Kahraman S, Celebi M E. Code based efficient maximum-likelihood decoding of short polar codes. In: *Proceedings of 2012 IEEE International Symposium on Information Theory (ISIT)*, Cambridge, 2012. 1967–1971
- 11 Niu K, Chen K, Lin J. Low-complexity sphere decoding of polar codes based on optimum path metric. *IEEE Commun Lett*, 2014, 18: 332–335
- 12 Guo J, Fabregas A G. Efficient sphere decoding of polar codes. In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, 2015. 236–240

- 13 Hashemi S A, Condo C, Gross W J. List sphere decoding of polar codes. In: Proceedings of the 49th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, 2015. 1346–1350
- 14 Hashemi S A, Condo C, Gross W J. Matrix reordering for efficient list sphere decoding of polar codes. In: Proceedings of 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, 2016. 1730–1733
- 15 Husmann C, Nikolaou P C, Nikitopoulos K. Reduced latency ML polar decoding via multiple sphere-decoding tree searches. *IEEE Trans Veh Technol*, 2018, 67: 1835–1839
- 16 Trifonov P. Efficient design and decoding of polar codes. *IEEE Trans Commun*, 2012, 60: 3221–3227
- 17 Tal I, Vardy A. How to construct polar codes. *IEEE Trans Inform Theor*, 2013, 59: 6562–6582
- 18 He G, Belfiore J C, Land I, et al. Beta-expansion: a theoretical framework for fast and recursive construction of polar codes. In: Proceedings of IEEE Global Communications Conference, Singapore, 2017. 1–6
- 19 Koopman P, Chakravarty T. Cyclic redundancy code (CRC) polynomial selection for embedded networks. In: Proceedings of International Conference on Dependable Systems and Networks, Florence, 2004. 145–154
- 20 Agrell E, Eriksson T, Vardy A, et al. Closest point search in lattices. *IEEE Trans Inform Theor*, 2002, 48: 2201–2214
- 21 Zhang Q, Liu A, Pan X, et al. CRC code design for list decoding of polar codes. *IEEE Commun Lett*, 2017, 21: 1229–1232
- 22 Hassibi B, Vikalo H. On the sphere-decoding algorithm I. Expected complexity. *IEEE Trans Signal Process*, 2005, 53: 2806–2818
- 23 Zhao W, Giannakis G B. Sphere decoding algorithms with improved radius search. *IEEE Trans Commun*, 2005, 53: 1104–1109
- 24 Balatsoukas-Stimming A, Parizi M B, Burg A. LLR-based successive cancellation list decoding of polar codes. *IEEE Trans Signal Process*, 2015, 63: 5165–5179