

SCL Decoding of Non-Binary Linear Block Codes

Jingyu Lin, Li Chen[✉], *Senior Member, IEEE*, and Xiaoqian Ye

Abstract—Non-binary linear block codes (NB-LBCs) are an important class of error-correcting codes that are especially competent in correcting burst errors. They have broad applications in modern communications and storage systems. However, efficient soft-decision decoding of these codes remains to be further developed. This letter proposes successive cancellation list (SCL) decoding for NB-LBCs that are defined over a finite field of characteristic two, i.e., \mathbb{F}_{2^r} , where r is the extension degree. By establishing a one-to- r mapping between the binary composition of each non-binary codeword and r binary polar codewords, SCL decoding of the r polar codes can be performed with a complexity that is sub-quadratic in the codeword length. A simplified path sorting is further proposed to facilitate the decoding. Simulation results on short-length extended Reed-Solomon (eRS) and non-binary extended BCH (NB-eBCH) codes show that SCL decoding can outperform their state-of-the-art soft-decision decoding with fewer finite field arithmetic operations. For length-16 eRS codes, their maximum-likelihood (ML) decoding performances can be approached with a moderate list size.

Index Terms—Non-binary linear block codes, successive cancellation list decoding, soft-decision decoding.

I. INTRODUCTION

NON-BINARY linear block codes (NB-LBCs) are an important class of error-correcting codes with wide applications in modern communications and storage systems. They are especially competent in correcting burst errors. The celebrated NB-LBCs include Reed-Solomon (RS) codes [1], algebraic-geometry (AG) codes [2], and non-binary BCH (NB-BCH) codes [3], [4]. Their algebraic decoding can be categorized into the syndrome-based approach and the curve-fitting-based approach. The former can efficiently correct errors up to half the code's minimum Hamming distance, leading to the Berlekamp-Massey (BM) algorithm [5], [6] being widely adopted in RS coded systems. The latter is also known as the Guruswami-Sudan (GS) algorithm [7]. It improves the error-correction capability beyond this bound while retaining a polynomial-time decoding complexity.

Further performance improvement can be achieved by utilizing soft information observed from the channel. The algebraic soft-decision decoding (ASD), also known as the Kötter-Vardy (KV) decoding [8], enhances the decoding of RS codes by converting the reliability information into the interpolation

multiplicities. For RS codes of length N , KV decoding with a maximum output list size L exhibits a complexity of $\mathcal{O}(N^2L^5)$ [9]. Another soft-decision decoding approach is the Chase decoding [10]. It constructs 2^η test-vectors by flipping the η least reliable symbols. With each test-vector decoded by the BM algorithm, the Chase-BM decoding exhibits a complexity of $\mathcal{O}(2^\eta N^2)$. Hence, soft-decision decoding of NB-LBCs are generally more complex. Efficient soft-decision decoding approach for NB-LBCs remains to be further developed.

In [11], it was revealed that any linear block code can be interpreted as a polar code with dynamic frozen symbols. Subsequently, successive cancellation (SC) decoding of RS codes was proposed in [12], where RS codes are transformed into non-binary polar codes. Sequential decoding and permutation-based decoding were applied in [12], [13], and [14] to further improve the decoding performance. Recent research proposed a general transformation from binary linear block codes (B-LBCs) to polar codes with dynamic frozen symbols [15]. It established a one-to-one mapping between B-LBC codewords and polar codewords, in which a permutation matrix is required to adjust the information set of the polar code. Consequently, SC and SC list (SCL) [16], [17] decoding, and their fast decoding version [18], [19], [20], can be applied to decode B-LBCs.

In this letter, SCL decoding is proposed for NB-LBCs that are defined over a finite field of characteristic two, i.e., \mathbb{F}_{2^r} , where r is the extension degree. By establishing a one-to- r mapping between the binary composition of each non-binary codeword and r binary polar codewords, SCL decoding of the r polar codes can be performed with a complexity of $\mathcal{O}(rLN\log_2 N)$, where L is the SCL decoding list size. A simplified path sorting is further proposed to facilitate the decoding. Simulation results on short-length extended RS (eRS) codes and non-binary extended BCH (NB-eBCH) codes show that SCL decoding can outperform their state-of-the-art soft-decision decoding with fewer finite field arithmetic operations. Moreover, for length-16 eRS codes, their maximum-likelihood (ML) decoding performances can be approached with a moderate list size.

Notation: Let \mathbb{F}_2 denote the binary field and \mathbb{F}_{2^r} subsequently denote its extension field of extension degree r . Further let $p(X)$ and α denote the primitive polynomial and the primitive element of \mathbb{F}_{2^r} , respectively. Given an element $\sigma \in \mathbb{F}_{2^r}$, it can be represented by $\sum_{j=0}^{r-1} \sigma_j \alpha^j$, where $\sigma_j \in \mathbb{F}_2$. Vector $(\sigma_0, \sigma_1, \dots, \sigma_{r-1})$ is the binary composition of σ . For convenience, we also use $\sigma[j]$ to denote σ_j . Given an integer set $\mathcal{A} \subset \{0, 1, \dots, N-1\}$, its cardinality and complement are denoted by $|\mathcal{A}|$ and \mathcal{A}^c , respectively.

II. PRELIMINARIES

A. Polar Codes

Let us consider a polar code of length $N = 2^n$ and dimension K . With kernel matrix $\mathbf{F} = ((1, 0), (1, 1))^T$, its generator

Received 5 January 2026; revised 29 January 2026; accepted 16 February 2026. Date of publication 23 February 2026; date of current version 4 March 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62471503 and in part by the Natural Science Foundation of Guangdong Province under Grant 2024A1515010213. The associate editor coordinating the review of this letter and approving it for publication was P. Giard. (*Corresponding author: Li Chen.*)

Jingyu Lin and Li Chen are with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China, and also with Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China (e-mail: linjy228@mail2.sysu.edu.cn; chenli55@mail.sysu.edu.cn).

Xiaoqian Ye is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: yexq26@mail2.sysu.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2026.3667113

1558-2558 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on May 26, 2026 at 09:40:04 UTC from IEEE Xplore. Restrictions apply.

matrix is $\mathbf{G}_p = \mathbf{F}^{\otimes n}$, where $\otimes n$ denotes the n -fold Kronecker product [21]. With an input vector $\mathbf{u} = (u_0, u_1, \dots, u_{N-1}) \in \mathbb{F}_2^N$, which is constituted by K information symbols and $N - K$ frozen symbols, its codeword \mathbf{c} is generated by

$$\mathbf{c} = \mathbf{u}\mathbf{G}_p. \quad (1)$$

The indices of information and frozen symbols constitute the information set \mathcal{A} and the frozen set \mathcal{A}^c , respectively. Let $\mathbf{u}^{\mathcal{A}} = (u_i | i \in \mathcal{A})$ denote the message. Codeword \mathbf{c} can be alternatively represented by

$$\mathbf{c} = \mathbf{u}^{\mathcal{A}}\mathbf{M}\mathbf{G}_p, \quad (2)$$

where $\mathbf{M} \in \mathbb{F}_2^{K \times N}$ is a pre-transformed matrix of reduced row echelon form. Note that indices of the pivot columns in \mathbf{M} form \mathcal{A} . The frozen symbol with index i is a linear combination of information symbols with indices smaller than i , i.e.,

$$u_i = \sum_{t=0}^{\tau_i} u_t^{\mathcal{A}} \cdot \mathbf{M}_{t,i}, \quad i \in \mathcal{A}^c, \quad (3)$$

where $\tau_i = |\mathcal{A} \cap \{0, \dots, i-1\}|$. It is referred to as a dynamic frozen symbol [11]. During SC and SCL decoding, all $u_t^{\mathcal{A}}$ with $t \leq \tau_i$ are estimated prior to u_i , which allows u_i to be determined accordingly.

B. Transformation From B-LBCs to Polar Codes

For a B-LBC \mathcal{C}_B of length $N = 2^n$ and dimension K , there exists a polar code with dynamic frozen symbols such that the one-to-one mapping between codewords of \mathcal{C}_B and the polar codewords can be established [15]. Let $\mathbf{G}_B \in \mathbb{F}_2^{K \times N}$ denote a generator matrix of \mathcal{C}_B and $\mathbf{m} \in \mathbb{F}_2^K$ denote a K -dimensional message. Given a permutation matrix $\mathbf{P} \in \mathbb{F}_2^{N \times N}$, codebook \mathcal{C}_B can be defined as

$$\begin{aligned} \mathcal{C}_B &\triangleq \{\mathbf{c} = \mathbf{m}\mathbf{G}_B \mid \forall \mathbf{m} \in \mathbb{F}_2^K\} \\ &= \{\mathbf{c} = \mathbf{u}\mathbf{G}_p\mathbf{P} \mid \mathbf{u} = \mathbf{m}\mathbf{G}_B\mathbf{P}^{-1}\mathbf{G}_p^{-1}, \forall \mathbf{m} \in \mathbb{F}_2^K\}. \end{aligned} \quad (4)$$

By performing Gaussian elimination (GE) on $\mathbf{G}_B\mathbf{P}^{-1}\mathbf{G}_p^{-1}$, the pre-transformed matrix $\mathbf{M} = \mathbf{E}\mathbf{G}_B\mathbf{P}^{-1}\mathbf{G}_p^{-1}$ can be obtained, where $\mathbf{E} \in \mathbb{F}_2^{K \times K}$ is a row elimination matrix. Hence,

$$\mathcal{C}_B \triangleq \{\mathbf{c} = \mathbf{u}^{\mathcal{A}}\mathbf{M}\mathbf{G}_p\mathbf{P} \mid \forall \mathbf{u}^{\mathcal{A}} \in \mathbb{F}_2^K\}, \quad (5)$$

where $\mathbf{u}^{\mathcal{A}} = \mathbf{m}\mathbf{E}^{-1}$. Therefore, each codeword of \mathcal{C}_B can be mapped to a permuted polar codeword with dynamic frozen symbols. \mathcal{C}_B can be decoded by estimating the polar codewords through SC or SCL decoding.

III. DECOMPOSITION OF NB-LBCS

To facilitate SCL decoding of NB-LBCs, a mapping between NB-LBCs and binary polar codes is required. It is established by the following Theorem 1.

Theorem 1: Let \mathcal{C}_{NB} denote an $(N = 2^n, K)$ NB-LBC defined over \mathbb{F}_{2^r} . The binary composition of any codeword $\mathbf{c} \in \mathcal{C}_{NB}$ can be represented as the concatenation of r permuted binary polar codewords.

Proof: Let $\mathbf{G}_{NB} \in \mathbb{F}_{2^r}^{K \times N}$ denote a generator matrix of \mathcal{C}_{NB} . With a message $\mathbf{m} \in \mathbb{F}_{2^r}^K$, its codeword \mathbf{c} is generated

by $\mathbf{c} = \mathbf{m}\mathbf{G}_{NB}$. Given a permutation matrix \mathbf{P} , \mathbf{c} can also be represented by

$$\mathbf{c} = \mathbf{u}\mathbf{G}_p\mathbf{P}, \quad (6)$$

where $\mathbf{u} = \mathbf{m}\mathbf{G}_{NB}\mathbf{P}^{-1}\mathbf{G}_p^{-1}$. Let

$$\begin{aligned} \mathbf{u}^B &= (u_{0,0}, \dots, u_{0,r-1}, u_{1,0}, \dots, u_{1,r-1}, \\ &\quad \dots, u_{N-1,0}, \dots, u_{N-1,r-1}) \end{aligned} \quad (7)$$

denote the binary composition of \mathbf{u} . Furthermore, let

$$\begin{aligned} \mathbf{c}^B &= (c_{0,0}, \dots, c_{0,r-1}, c_{1,0}, \dots, c_{1,r-1}, \\ &\quad \dots, c_{N-1,0}, \dots, c_{N-1,r-1}) \end{aligned} \quad (8)$$

denote the binary composition of \mathbf{c} . Since \mathbf{G}_p and \mathbf{P} are binary matrices, the multiplication between \mathbf{u} and $\mathbf{G}_p\mathbf{P}$ involves only \mathbb{F}_{2^r} additions, which can be decomposed into \mathbb{F}_2 additions. For $j = 0, 1, \dots, r-1$, let

$$\mathbf{u}_j^B = (u_{0,j}, u_{1,j}, \dots, u_{N-1,j}), \quad (9)$$

$$\mathbf{c}_j^B = (c_{0,j}, c_{1,j}, \dots, c_{N-1,j}). \quad (10)$$

Based on (6), one can obtain

$$\mathbf{c}_j^B = \mathbf{u}_j^B\mathbf{G}_p\mathbf{P}. \quad (11)$$

By performing GE on $\mathbf{G}_{NB}\mathbf{P}^{-1}\mathbf{G}_p^{-1}$, the non-binary pre-transformed matrix $\mathbf{T} = \mathbf{R}\mathbf{G}_{NB}\mathbf{P}^{-1}\mathbf{G}_p^{-1}$ can be obtained, where $\mathbf{R} \in \mathbb{F}_{2^r}^{K \times K}$ is a row elimination matrix. Consequently,

$$\mathbf{u} = \mathbf{u}^{\mathcal{A}}\mathbf{T}, \quad (12)$$

where $\mathbf{u}^{\mathcal{A}} = \mathbf{m}\mathbf{R}^{-1}$. Hence, $\mathbf{c}_0^B, \mathbf{c}_1^B, \dots, \mathbf{c}_{r-1}^B$ share an identical information set, i.e., indices of the pivot columns in \mathbf{T} . The frozen symbols are determined as in (3) but through \mathbb{F}_{2^r} additions and multiplications. Therefore, \mathbf{c}^B is the concatenation of r permuted binary polar codewords. \square

Therefore, \mathcal{C}_{NB} can be decoded by estimating the r binary polar codewords through SC or SCL decoding and then reconstructing the non-binary codeword. Obtaining the pre-transformed matrix \mathbf{T} requires a complexity of $\mathcal{O}(KN \log_2 N + NK^2)$. Since this can be performed offline, it does not affect the overall decoding complexity.

Theoretically, the SC decoding error probability of binary polar codes is upper bounded by $P_e^{\text{UB}} = \sum_{i \in \mathcal{A}} P_e(\mathcal{W}_i)$, where \mathcal{W}_i denotes the i -th polarized subchannel of a length- N polar code and $P_e(\mathcal{W}_i)$ denotes its error probability [21]. This implies that the SC decoding performance of \mathcal{C}_{NB} is determined by the information set \mathcal{A} , which is further determined by the permutation matrix \mathbf{P} . Hence, a permutation matrix that minimizes P_e^{UB} should be chosen to optimize the SC decoding performance. According to [12] and [15], there exists an effective permutation for eRS codes, which is defined as

$$\mathbf{P}_{a,b} = \begin{cases} 1, & \text{if } a = \sum_{j=0}^{m-1} (\alpha^b[j] \cdot 2^j), 0 \leq b \leq N-2 \\ & \text{or } a = 0, b = N-1; \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

where $\mathbf{P}_{a,b}$ is the row- a column- b entry of \mathbf{P} . It results in the first few subchannels being frozen, which helps reduce

P_e^{UB} since their error probabilities are relatively high. E.g., for code rate between 0.25 and 0.5, \mathcal{W}_0 , \mathcal{W}_1 and \mathcal{W}_2 are frozen regardless of N , implying that the first information symbol is transferred over \mathcal{W}_3 . Due to channel polarization [21], $P_e(\mathcal{W}_3)$ increases with N , resulting in a corresponding increase in P_e^{UB} . Hence, when \mathbf{P} of (13) is applied, SC decoding performance of eRS codes degrades as N increases. In this letter, this \mathbf{P} is applied in SCL decoding of short-length eRS and NB-eBCH codes, which are extended to length 2^n by padding a parity symbol as $c_{2^n-1} = \sum_{i=0}^{2^n-2} c_i$. Its optimality remains to be proven. For other NB-LBCs, heuristic search approaches can be applied to obtain effective permutations.

IV. SCL DECODING OF NB-LBCS

A. SC and SCL Decoding

Assume that codeword $\mathbf{c} = (c_0, c_1, \dots, c_{N-1}) \in \mathbb{F}_{2^r}^N$ of \mathcal{C}_{NB} is transmitted over a memoryless channel and $\mathbf{y} = (y_0, y_1, \dots, y_{N-1}) \in \mathbb{R}^N$ is the received vector. Let $\mathbf{c}' = \mathbf{c}\mathbf{P}^{-1}$ and $\mathbf{y}' = \mathbf{y}\mathbf{P}^{-1}$. Let

$$\mathcal{L} = (\mathcal{L}_{0,0}, \dots, \mathcal{L}_{0,r-1}, \mathcal{L}_{1,0}, \dots, \mathcal{L}_{1,r-1}, \dots, \mathcal{L}_{N-1,0}, \dots, \mathcal{L}_{N-1,r-1}) \quad (14)$$

denote the log-likelihood ratio (LLR) vector with entries defined as

$$\mathcal{L}_{i,j} = \ln \frac{P(y'_i | c'_{i,j} = 0)}{P(y'_i | c'_{i,j} = 1)}, \quad (15)$$

where $i = 0, 1, \dots, N-1$ and $j = 0, 1, \dots, r-1$. These LLRs are partitioned into r groups, each of which is the input LLR vector of an SC decoder. In particular, the input LLR vector of the j -th SC decoder is

$$\mathcal{L}_j^{(n)} = (\mathcal{L}_{j,0}^{(n)}, \mathcal{L}_{j,1}^{(n)}, \dots, \mathcal{L}_{j,N-1}^{(n)}), \quad (16)$$

where $\mathcal{L}_{j,i}^{(n)} = \mathcal{L}_{i,j}$. For $0 \leq s \leq n-1$, the stage- s LLRs are computed by [22]

$$\begin{aligned} \mathcal{L}_{j,i}^{(s)} &= f(\mathcal{L}_{j,i}^{(s+1)}, \mathcal{L}_{j,i+2^s}^{(s+1)}), \\ \mathcal{L}_{j,i+2^s}^{(s)} &= (-1)^{\hat{u}_{j,i}^{(s)}} \mathcal{L}_{j,i}^{(s+1)} + \mathcal{L}_{j,i+2^s}^{(s+1)}, \end{aligned} \quad (17)$$

where $f(\mathcal{X}, \mathcal{Y}) \triangleq \ln \frac{e^{\mathcal{X}} e^{\mathcal{Y}} + 1}{e^{\mathcal{X}} + e^{\mathcal{Y}}}$ and $\mathcal{X}, \mathcal{Y} \in \mathbb{R}$. When reaching stage-0, hard decisions are made based on the LLRs, i.e.,

$$\eta_{j,i} = \begin{cases} 0, & \text{if } \mathcal{L}_{j,i}^{(0)} \geq 0; \\ 1, & \text{otherwise.} \end{cases} \quad (18)$$

Then the binary input symbols are estimated by

$$\hat{u}_{j,i}^{(0)} = \begin{cases} \eta_{j,i}, & \text{if } i \in \mathcal{A}; \\ \left(\sum_{t=0}^{\tau_i} \hat{u}_t^A \cdot \mathbf{T}_{t,i} \right) [j], & \text{if } i \in \mathcal{A}^c, \end{cases} \quad (19)$$

where \hat{u}_t^A is the estimation of the t -th non-binary information symbol. The estimations of the non-binary input symbols is determined by

$$\hat{u}_i = \sum_{j=0}^{r-1} \hat{u}_{j,i}^{(0)} \alpha^j, \quad (20)$$

where α again is the primitive element of \mathbb{F}_{2^r} . For $1 \leq s \leq n$, the stage- s binary estimations are computed by

$$\begin{aligned} \hat{u}_{j,i}^{(s)} &= \hat{u}_{j,i}^{(s-1)} + \hat{u}_{j,i+2^{s-1}}^{(s-1)}, \\ \hat{u}_{j,i+2^{s-1}}^{(s)} &= \hat{u}_{j,i+2^{s-1}}^{(s-1)}. \end{aligned} \quad (21)$$

Therefore, the SC decoding of \mathcal{C}_{NB} is performed by running r SC decoders and estimating $\hat{\mathbf{u}} = (\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-1})$ in a symbol-by-symbol manner.

SCL decoding of \mathcal{C}_{NB} evolves from the above SC decoding by considering all possible values for information symbols, i.e.,

$$\hat{u}_i = \begin{cases} \forall \sigma \in \mathbb{F}_{2^r}, & \text{if } i \in \mathcal{A}; \\ \sum_{t=0}^{\tau_i} \hat{u}_t^A \cdot \mathbf{T}_{t,i}, & \text{if } i \in \mathcal{A}^c. \end{cases} \quad (22)$$

For each information symbol, a decoding path is split into 2^r paths. In order to curb this exponentially increasing complexity, only the L most likely paths are kept by the decoder. For this, a path metric should be defined to measure the likelihood of the paths. Assume that there are L surviving paths after estimating \hat{u}_{i-1} . Their path metrics are defined as [22]

$$\Phi_{i-1}(l) = \sum_{h=0}^{i-1} \sum_{j: \hat{u}_{j,h}^{(0)}(l) \neq \eta_{j,h}(l)} |\mathcal{L}_{j,h}^{(0)}(l)|, \quad (23)$$

where $0 \leq l \leq L-1$. Paths with smaller metrics are more likely to be correct. If $i \in \mathcal{A}$, the l -th decoding path is split into 2^r decoding paths with metrics

$$\Phi_i(l, \sigma) = \Phi_{i-1}(l) + \sum_{j: \sigma_j \neq \eta_{j,i}(l)} |\mathcal{L}_{j,i}^{(0)}(l)|, \quad \forall \sigma \in \mathbb{F}_{2^r}. \quad (24)$$

Overall, $2^r L$ decoding paths are generated. A $(2^r L)$ -to- L path pruning is performed to select the L paths with the smallest metrics. When $i = N-1$, the decoding path with the smallest metric will be selected as the decoding result.

Algorithm 1 summarizes the above SCL decoding process. Complexity of the LLR computation and path sorting are $\mathcal{O}(rLN \log_2 N)$ and $\mathcal{O}(K2^r L \log_2(2^r L))$, respectively. Assume that operations of (17) are completed in one clock cycle and the r SC decoders operate in parallel, the number of required clock cycles of the SCL decoding is $2N-2+KT_{\text{sort}}$, where T_{sort} denotes the number of required clock cycles for path sorting. It should be noted that the actual runtime is implementation-dependent, while the above analysis focuses on the dominant operations.

B. Simplified Path Sorting

In the above SCL decoding, redundant comparisons exist in the full path sorting at each information symbol. E.g., let us consider $\beta, \gamma \in \mathbb{F}_{2^r}$ and their binary compositions only differ at the j' -th component, i.e., for $j = 0, 1, \dots, j'-1, j'+1, \dots, r-1$, $\beta_j = \gamma_j$. If $\beta_{j'} = \eta_{j',i}(l)$ and $\gamma_{j'} \neq \eta_{j',i}(l)$, then based on (24), one can immediately obtain $\Phi_i(l, \beta) \leq \Phi_i(l, \gamma)$ without explicit comparison. This observation leads to a simplified path sorting [20]. It eliminates redundant path splits and unnecessary path metric comparisons during the full path sorting process.

Algorithm 1 SCL Decoding of NB-LBCs

```

1 for  $i = 0, 1, \dots, N - 1$  do
2   Compute  $\mathcal{L}_{j,i}^{(0)}(l)$  as in (17);
3   if  $i \in \mathcal{A}$  then
4     Compute  $\Phi_i(l, \sigma)$  as in (24);
5     Select  $L$  paths with the smallest path metrics;
6   else
7     Compute  $\hat{u}_i(l) = \sum_{t=0}^{\tau_i} \hat{u}_t^A(l) \cdot \mathbf{T}_{t,i}$ ;
8     Compute  $\Phi_i(l)$  as in (23);
9   Select the decoding path with the smallest path metric;
10  Reconstruct  $\hat{c}$  as in (6);

```

It starts by sorting the path metrics of the surviving paths in ascending order. Let $\tilde{x}_l = \Phi_{i-1}(p_l)$, where $p_l \in \{0, 1, \dots, L-1\}$ denotes the original path index corresponding to the l -th smallest path metric. The ordered metric vector and its associated index vector are given by

$$\tilde{\mathbf{x}} = (\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{L-1}), \quad (25)$$

$$\mathbf{p} = (p_0, p_1, \dots, p_{L-1}), \quad (26)$$

where $\tilde{x}_0 \leq \tilde{x}_1 \leq \dots \leq \tilde{x}_{L-1}$.

The subsequent sorting procedure then proceeds in r successive steps. At step- j , where $0 \leq j \leq r-1$, let $x_l^+ = \tilde{x}_l + |\mathcal{L}_{j,i}^{(0)}(p_l)|$. This yields the updated metric vector

$$\mathbf{x}^+ = (x_0^+, x_1^+, \dots, x_{L-1}^+). \quad (27)$$

Sorting \mathbf{x}^+ in ascending order yields $\tilde{x}_l^+ = x_{p_l^+}^+$, where $p_l^+ \in \{0, 1, \dots, L-1\}$ denotes the original index of the l -th smallest value in \mathbf{x}^+ . The ordered version of \mathbf{x}^+ and its associated index vector are thus

$$\tilde{\mathbf{x}}^+ = (\tilde{x}_0^+, \tilde{x}_1^+, \dots, \tilde{x}_{L-1}^+), \quad (28)$$

$$\mathbf{p}^+ = (p_0^+, p_1^+, \dots, p_{L-1}^+), \quad (29)$$

where $\tilde{x}_0^+ \leq \tilde{x}_1^+ \leq \dots \leq \tilde{x}_{L-1}^+$. Since $\tilde{x}_l^+ = x_{p_l^+}^+$ and $x_{p_l^+}^+ = \tilde{x}_{p_l^+} + |\mathcal{L}_{j,i}^{(0)}(p_{p_l^+})|$, the decoding path with metric \tilde{x}_l^+ is split from the $(p_{p_l^+})$ -th original path. By performing a merge sort on $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}^+$, the L smallest values among the two sequences can be selected. The ordered metric vector $\tilde{\mathbf{x}}$ is then updated with these values and their original path indices are recorded in \mathbf{p} . The remaining elements in $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}^+$ are discarded. Since path metrics increase monotonically at each step, any metric that is not among the L smallest values at the current step cannot become one of the L smallest after further increments. As a result, after step- $(r-1)$, the same L paths as those selected by full path sorting are retained.

Algorithm 2 summarizes the above sorting procedure. During it, $(r+1)$ full sorts of length- L sequences and r merge sorts between two ordered length- L sequences are required. The overall sorting complexity is reduced from $\mathcal{O}(2^r L \log_2(2^r L))$ to $\mathcal{O}((r+1)L \log_2 L + rL)$.

V. SIMULATION RESULTS

This section shows our simulation results on SCL decoding of eRS codes and NB-eBCH codes. They are obtained over the

Algorithm 2 Simplified Path Sorting

```

Input:  $\mathcal{L}_{j,i}^{(0)}(l), \Phi_{i-1}(l)$ ;
Output:  $\Phi_i(l)$ ;
1 Sort  $\Phi_{i-1}(0), \dots, \Phi_{i-1}(L-1)$ , yielding  $\tilde{\mathbf{x}}$  and  $\mathbf{p}$ ;
2 for  $j = 0, 1, \dots, r-1$  do
3    $x_l^+ = \tilde{x}_l + |\mathcal{L}_{j,i}^{(0)}(p_l)|, l = 0, 1, \dots, L-1$ ;
4   Sort  $\mathbf{x}^+$ , yielding  $\tilde{\mathbf{x}}^+$  and  $\mathbf{p}^+$ ;
5   Perform merge sort on  $\tilde{\mathbf{x}}$  and  $\tilde{\mathbf{x}}^+$  and keep the  $L$ 
     smallest values in  $\tilde{\mathbf{x}}$ ;
6   Update  $\mathbf{p}$  as the original indices of paths in  $\tilde{\mathbf{x}}$ ;
7    $\{\Phi_i(l) : l = 0, 1, \dots, L-1\} \leftarrow \tilde{\mathbf{x}}$ ;

```

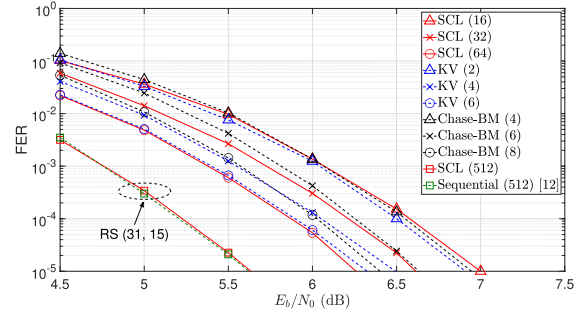


Fig. 1. SCL decoding performance of the (32, 15) eRS code.

TABLE I
DECODING COMPLEXITY OF THE (32, 15) ERS CODE

Scheme	\mathbb{F}_{2^5} oper.	FLOPs
SCL(16)	3.11×10^3	1.41×10^4
SCL(32)	6.08×10^3	3.27×10^4
SCL(64)	1.19×10^4	7.32×10^4
KV(2)	6.37×10^4	6.80×10^4
KV(4)	5.06×10^5	1.13×10^5
KV(6)	2.05×10^6	1.59×10^5
Chase-BM(4)	4.93×10^4	7.99×10^3
Chase-BM(6)	2.00×10^5	8.71×10^3
Chase-BM(8)	8.08×10^5	1.14×10^4

additive white Gaussian noise (AWGN) channel using BPSK modulation, where noise variance is $\frac{N_0}{2}$. The signal-to-noise ratio (SNR) is defined as $\frac{E_b}{N_0}$, where E_b is the transmitted energy per information bit. Note that the Chase-BM decoding that flips the η least reliable positions is denoted as Chase-BM(η). The KV decoding with a maximum output list size of L is denoted as KV(L).

Fig. 1 compares the frame error rate (FER) performance of the (32, 15) eRS code under SCL, KV and Chase-BM decoding. Table I further compares their decoding complexity at the SNR of 6 dB. It can be seen that SCL decoding can outperform KV and Chase-BM decoding with fewer finite field arithmetic operations. E.g., the SCL(64) decoding not only slightly outperforms the KV(6) decoding, but also reduces the number of finite field arithmetic operations by two orders of magnitude. Compared with the Chase-BM(8) decoding, the SCL(64) decoding yields a 0.1 dB performance gain while reducing the number of finite field arithmetic operations by an order of magnitude. Fig. 1 also compares the performance of the SCL(512) decoding and the sequential decoding with a maximum list size of 512 [12] on the (31, 15) RS code, which

TABLE II
FLOPS REQUIRED FOR SCL DECODING OF THE (32, 15) ERS CODE

	Full path sorting	Simplified path sorting
SCL(16)	9.29×10^4	1.41×10^4
SCL(32)	2.05×10^5	3.27×10^4
SCL(64)	6.97×10^5	7.32×10^4

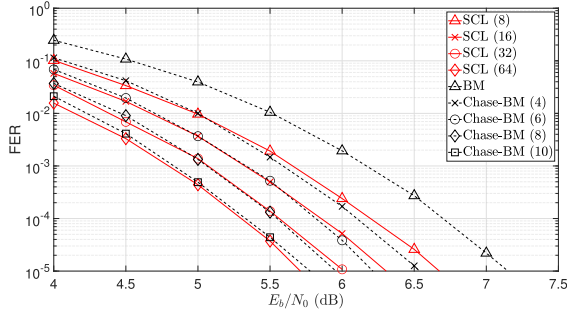


Fig. 2. SCL decoding performance of the (64, 27) NB-eBCH code.

TABLE III
DECODING COMPLEXITY OF THE (64, 27) NB-eBCH CODE

Scheme	\mathbb{F}_4 oper.	FLOPs
SCL(16)	8.31×10^3	1.52×10^4
SCL(32)	1.63×10^4	3.28×10^4
SCL(64)	3.23×10^4	7.13×10^4
Chase-BM(6)	4.63×10^5	2.77×10^3
Chase-BM(8)	1.87×10^6	4.04×10^3
Chase-BM(10)	7.57×10^6	9.06×10^3

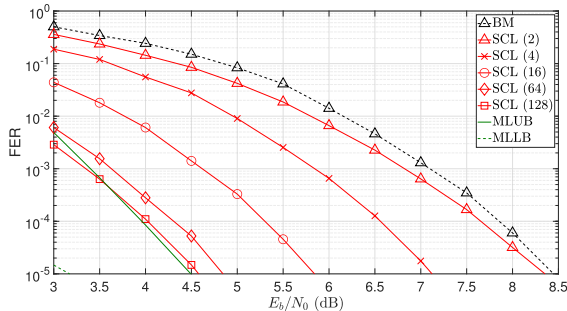


Fig. 3. SCL decoding performance of the (16, 7) eRS code.

can be seen as a punctured (32, 15) eRS code. They achieve a similar performance. Sequential decoding with a maximum list size of L has a worst-case complexity of $\mathcal{O}(L(N \log_2^2 N + N^2))$ [12], which is higher than that of the SCL(L) decoding, i.e., $\mathcal{O}(rLN \log_2 N)$. Table II shows that the proposed simplified path sorting reduces the number of FLOPs by an order of magnitude.

Fig. 2 compares the FER performance of the (64, 27) NB-eBCH code under the SCL, BM and Chase-BM decoding. The NB-eBCH code is defined over \mathbb{F}_4 . Table III further compares their decoding complexity at the SNR of 6 dB. Similarly, SCL decoding reduces the number of finite field arithmetic operations at the cost of an increased number of FLOPs. E.g., while the SCL(64) and Chase-BM(10) decoding achieve similar FER performances, the SCL(64) decoding requires two orders of magnitude fewer finite field arithmetic operations and an order of magnitude more FLOPs.

Finally, Fig. 3 shows the FER performance of SCL decoding of the (16, 7) eRS code. The ML decoding upper and

lower bounds [23], denoted as MLUB and MLLB, are also shown. When the list size $L = 128$, the SCL decoding approaches the MLUB. However, as the code length increases, approaching the MLUB becomes increasingly difficult, since the SC decoding suffers from performance degradation when the permutation matrix of (13) is applied. Therefore, the SCL decoding would require a larger list size to achieve a good performance.

REFERENCES

- I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.
- V. D. Goppa, "Codes associated with divisors," *Problemy Peredachi Informatsii*, vol. 13, no. 1, pp. 33–39, 1977.
- R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, Mar. 1960.
- A. Hocquenghem, "Codes correcteurs d'erreurs. Chiffres (Paris), 2, 147–156," *Math. Rev.*, vol. 22, p. 652, Sep. 1959.
- E. R. Berlekamp, *Algebraic Coding Theory*. New York, NY, USA: McGraw-Hill, 1968.
- J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.
- V. Guruswami and M. Sudan, "Improved decoding of Reed–Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, Sep. 1999.
- R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.
- J. Xing, L. Chen, and M. Bossert, "Progressive algebraic soft-decision decoding of Reed–Solomon codes using module minimization," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7379–7391, Nov. 2019.
- D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.
- P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2013, pp. 1–5.
- P. V. Trifonov, "Successive cancellation decoding of Reed–Solomon codes," *Problems Inf. Transmiss.*, vol. 50, no. 4, pp. 303–312, Oct. 2014.
- V. Miloslavskaya and P. Trifonov, "Sequential decoding of Reed–Solomon codes," in *Proc. Int. Symp. Inf. Theory Appl.*, Oct. 2014, pp. 453–457.
- P. Trifonov, "Successive cancellation permutation decoding of Reed–Solomon codes," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2014, pp. 386–390.
- C.-Y. Lin, Y.-C. Huang, S.-L. Shieh, and P.-N. Chen, "Transformation of binary linear block codes to polar codes with dynamic frozen," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 333–341, Mar. 2020.
- I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.
- A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011.
- H. Sun, E. Viterbo, B. Dai, and R. Liu, "Fast decoding of polar codes for digital broadcasting services in 5G," *IEEE Trans. Broadcast.*, vol. 70, no. 2, pp. 731–738, Jun. 2024.
- S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, Nov. 2017.
- E. Arikan, "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.
- A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.
- M. El-Khamy and R. J. McEliece, "Bounds on the average binary minimum distance and the maximum likelihood performance of Reed–Solomon codes," in *Proc. 42nd Allerton Conf. Commun. Control Comput.*, Sep. 2004, pp. 290–299.