# On the Decoding of Polar Codes on Permuted Factor Graphs

Nghia Doan*, Seyyed Ali Hashemi*, Marco Mondelli†, Warren J. Gross*

*Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada
†Department of Electrical Engineering, Stanford University, California, USA
Email: nghia.doan@mail.mcgill.ca, seyyed.hashemi@mail.mcgill.ca, mondelli@stanford.edu, warren.gross@mcgill.ca

*Abstract*—Polar codes are a channel coding scheme for the next generation of wireless communications standard (5G). The belief propagation (BP) decoder allows for parallel decoding of polar codes, making it suitable for high throughput applications. However, the error-correction performance of polar codes under BP decoding is far from the requirements of 5G. It has been shown that the error-correction performance of BP can be improved if the decoding is performed on multiple permuted factor graphs of polar codes. However, a different BP decoding scheduling is required for each factor graph permutation which results in the design of a different decoder for each permutation. Moreover, the selection of the different factor graph permutations is at random, which prevents the decoder to achieve a desirable error-correction performance with a small number of permutations. In this paper, we first show that the permutations on the factor graph can be mapped into suitable permutations on the codeword positions. As a result, we can make use of a single decoder for all the permutations. In addition, we introduce a method to construct a set of predetermined permutations which can provide the correct codeword if the decoding fails on the original permutation. We show that for the 5G polar code of length $1024$, the error-correction performance of the proposed decoder is more than $0.25$ dB better than that of the BP decoder with the same number of random permutations at the frame error rate of $10^{-4}$.

*Index Terms*—polar codes, belief propagation decoding, permuted factor graph.

## I. INTRODUCTION

Polar codes [1] are the first class of error-correcting codes that was proved to achieve channel capacity with efficient encoding and decoding algorithms. As such, they have been adopted for the enhanced mobile broadband (eMBB) control channel of the fifth generation (5G) wireless communications standard. Successive cancellation (SC) and belief propagation (BP) decoding algorithms are two methods introduced in [1] to decode polar codes. Although SC decoding can provide a low-complexity solution, its serial nature prevents the decoding to reach high throughput. Furthermore, the polarization phenomenon of polar codes under SC decoding requires large block lengths. Thus, for short to moderate code lengths, the error-correction performance of SC decoding is insufficient for many practical applications. To improve the performance of SC decoding, SC list (SCL) decoding was proposed in [2]. SCL decoding maintains a list of most likely codewords by running a list of coupled SC decoders in parallel. An enormous improvement in the error probability of SCL decoder is obtained if a cyclic redundancy check (CRC) is used to select

the correct codeword [2]. However, SCL decoding still suffers from the serial nature and low throughput [3], which scales with list size. Several attempts have been made to reduce the computational complexity and increase the throughput of polar code SCL decoders [4]–[6].

Unlike SC and SCL decoding, the message passing process of BP decoding can be executed in parallel, which allows this decoder to obtain high throughput. However, BP decoding cannot achieve low error probability with a limited number of iterations. One of the methods used to improve the error-correction performance of BP decoding for polar codes is to utilize the redundant factor graph representations [7], [8]. It was shown in [7], [8] that for a polar code of length $N$, there are $(\log_2 N)!$ redundant representations which can be constructed by different permutations of the layers in the factor graph of polar codes. [7], [8] also suggested to use only the $\log_2 N$ cyclic permutations in the factor graph layers, in order to limit the large number of factor graph permutations. In [9], it was shown that if the permutations in the factor graph of polar codes are selected randomly and a CRC is concatenated to the polar code, the error probability of decoding a polar code on the permuted factor graph outperforms the performance of a non CRC-aided SCL decoder. However, the number of randomly selected permutations and therefore the number of parallel BP decoders required to achieve a reasonable error probability in [9] is too high. Moreover, the BP decoder scheduling has to be changed for each permutation of the factor graph layers. Therefore, a different decoder has to be designed for each permutation in practical applications.

In this paper, we first show that the permutations on the polar code factor graph can be mapped into permutations on the codeword positions, therefore allowing the use of the same decoder structure for all the permutations. Second, based on the observation that the decoding on the original permutation only fails with a small error probability, we propose a novel empirical approach to construct a set of good permutations. Our experimental results show that for the polar code selected for the eMBB control channel of 5G, with length $1024$, rate $0.5$, and by using a 24-bit CRC, the error-correction performance is more than $0.25$ dB better than that of the BP decoder when the same number of random permutations are employed, at the frame error rate (FER) of $10^{-4}$.

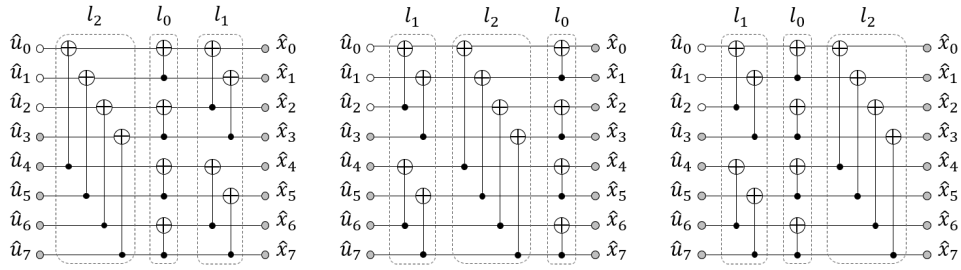The remainder of this paper is organized as follows. Section II briefly introduces a background on polar codes and

BP decoding. Section III and Section IV present the proposed decoder and the experimental results, respectively. Finally, concluding remarks are provided in Section V.

## II. PRELIMINARIES

### A. Polar Codes

A polar code $\mathcal{P}(N, K)$ of length $N$ with $K$ information bits is constructed by applying a linear transformation to the message word $\boldsymbol{u} = \{u_0, u_1, \ldots, u_{N-1}\}$ as $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}^{\otimes n}$ where $\boldsymbol{x} = \{x_0, x_1, \ldots, x_{N-1}\}$ is the codeword, $\boldsymbol{G}^{\otimes n}$ is the $n$-th Kronecker power of the polarizing matrix $\boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, and $n = \log_2 N$. The vector $\boldsymbol{u}$ contains a set $\mathcal{A}$ of $K$ information bits and a set $\mathcal{F}$ of $N - K$ frozen bits. The positions and the value of the frozen bits are known to the encoder and the decoder. The codeword $\boldsymbol{x}$ is then modulated and sent through the channel. In this paper, binary phase-shift keying (BPSK) modulation and additive white Gaussian noise (AWGN) channel model are considered, therefore the soft vector of the transmitted codeword received by the decoder is

$$\boldsymbol{y} = (\boldsymbol{1} - 2\boldsymbol{x}) + \boldsymbol{z}, \qquad (1)$$

where $\boldsymbol{1}$ is an all-one vector with size $N$, and $\boldsymbol{z} \in \mathbb{R}^N$ is the AWGN noise vector with variance $\sigma^2$ and zero mean. In the log-likelihood ratio (LLR) domain, the LLR vector of the transmitted codeword is

$$\mathrm{LLR}(\boldsymbol{x}) = \ln \frac{\Pr(\boldsymbol{x} = 0|\boldsymbol{y})}{\Pr(\boldsymbol{x} = 1|\boldsymbol{y})} = \frac{2\boldsymbol{y}}{\sigma^2}. \qquad (2)$$

### B. Belief Propagation Decoding

BP decoding is an iterative message passing algorithm applied on the factor graph representation of a code. The received channel LLR values, $\mathrm{LLR}(\boldsymbol{x})$, are iteratively propagated through the graph until either the LLR values converge or the maximum number of iterations is reached. The decoder then makes a hard decision based on the resulting LLR values to maximize the *a posteriori* probability $\Pr(\boldsymbol{x}|\boldsymbol{y})$.

Fig. 1a illustrates BP decoding on a factor graph representation of $\mathcal{P}(8, 5)$. The messages are propagated through the processing elements (PEs) [10] located in each layer. An update iteration starts with a right-to-left message pass that propagates the LLR values from the channel (rightmost) layer, to the information bit (leftmost) layer, and ends with the left-to-right message pass which occurs in the reverse order. Fig. 1b shows a PE with its corresponding messages, where $R_{i,l}$ denotes a left-to-right message, and $L_{i,l}$ denotes a right-to-left message of the $i$-th bit index at layer $l$. The update rule [10] of the messages in each PE is given as

$$\begin{aligned} L_{i,l} &= f(L_{i,l+1}, R_{i+2^l,l} + L_{i+2^l,l+1}), \\ L_{i+2^l,l} &= f(L_{i,l+1}, R_{i,l}) + L_{i+2^l,l+1}, \\ R_{i,l+1} &= f(R_{i,l}, L_{i+2^l,l+1} + R_{i+2^l,l}), \\ R_{i+2^l,l+1} &= f(R_{i,l}, L_{i,l+1}) + R_{i+2^l,l}, \end{aligned} \qquad (3)$$

where $f(x, y) = 2\operatorname{arctanh}(\tanh \frac{x}{2} \tanh \frac{y}{2})$ for any $x, y \in \mathbb{R}$.



Fig. 1: (a) BP decoding on the factor graph of $\mathcal{P}(8, 5)$ with $\{u_0, u_1, u_2\} \in \mathcal{F}$, (b) a processing element (PE).

Initially, $L_{i,n-1}$ ($0 \leq i < N$) are set to the received channel LLR values, $R_{i,0}$ ($i \in \mathcal{F}$) are set to $+\infty$, and all the other left-to-right and right-to-left messages of the PEs are set to 0. The BP decoding performs a predetermined $I_{\max}$ update iterations where the messages are propagated through all PEs in accordance with (3). The decoder then makes a hard decision on the LLR values of the $i$-th bit at the information bit layer to obtain the estimated message word as

$$\hat{u}_i = \begin{cases} 0, & \text{if } R_{i,0} + L_{i,0} \geq 0, \\ 1, & \text{otherwise.} \end{cases} \qquad (4)$$

In order to reduce the decoding latency, $G$-matrix-based [11] or CRC-based [12] early termination conditions are checked after each update iteration. If the early termination condition is satisfied, the decoding is terminated with the assumption that the LLR values have converged to indicate the correct codeword.

### C. BP Decoding on Permuted Factor Graphs

Factor graph permutations are a way to provide multiple representations of a single code. It was observed in [7], [8] that there exists $n!$ different ways to represent a polar code by permuting the layers in its factor graph. Fig. 2 illustrates such permutations for $\mathcal{P}(8, 5)$, where 3 out of $3! = 6$ permutations are shown. Note that the two leftmost factor graphs in Fig. 2 are formed by applying cyclic shifts to the original factor graph depicted in Fig. 1a. In [9], parallel BP decoders are applied on a set of randomly selected factor graphs of a polar code concatenated with a CRC. Although this decoding scheme

Fig. 2: Permuted factor graph representations for $\mathcal{P}(8,5)$.



Original codeword positions
Permuted graph layers

$$\pi(0) = 000_2 = 0$$
$$\pi(1) = 010_2 = 2$$
$$\pi(2) = 100_2 = 4$$
$$\pi(3) = 110_2 = 6$$
$$\pi(4) = 001_2 = 1$$
$$\pi(5) = 011_2 = 3$$
$$\pi(6) = 101_2 = 5$$
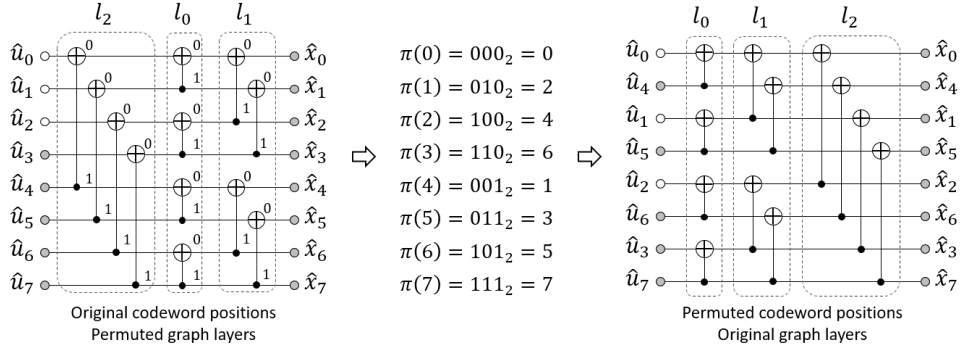$$\pi(7) = 111_2 = 7$$

Permuted codeword positions
Original graph layers

Fig. 3: The proposed mapping from factor graph permutation to bit-index permutation for $\mathcal{P}(8,5)$.

shows improvement in error probability when compared to a non CRC-aided SCL decoder, permuting layers results in different BP scheduling which consequently requires the design of a different BP decoder for each permutation. Moreover, a large number of random permutations are required to achieve a reasonable error-correction performance, which makes this decoding scheme too complex for practical applications.

## III. IMPROVED BP DECODING ON PERMUTED FACTOR GRAPHS

In this section, we first show that there is a one-to-one mapping between the permutations on the factor graph layers and the permutations on the codeword positions. We then propose a method to select the factor graph permutations that improves the error-correction performance of polar codes when decoding is performed on the permuted factor graphs.

### A. From Permutations on Factor Graph Layers to Permutations on Codeword Positions

We denote by $\{l_{n-1}, \ldots, l_0\}$ the layers of the original factor graph of polar codes (the one represented in the right part of Fig. 3) and by $\{b_{n-1}^{(i)}, \ldots, b_0^{(i)}\}$ the binary expansion of the integer $i$.

**Theorem 1.** *Let $\pi : \{0, \ldots, n-1\} \rightarrow \{0, \ldots, n-1\}$ be a permutation. Then, the synthetic channel associated to the position with binary expansion $\{b_{n-1}^{(i)}, \ldots, b_0^{(i)}\}$ on the factor graph with layers $\{l_{n-1}, \ldots, l_0\}$ is the same as the synthetic channel associated to the position with binary expansion $\{b_{\pi(n-1)}^{(i)}, \ldots, b_{\pi(0)}^{(i)}\}$ on the factor graph with layers $\{l_{\pi(n-1)}, \ldots, l_{\pi(0)}\}$.*

*Proof.* Consider the original factor graph of polar codes. Then, the synthetic channel associated to the position with binary expansion $\{b_{n-1}^{(i)}, \ldots, b_0^{(i)}\}$ is given by

$$(((W^{(b_{n-1}^{(i)})})^{(b_{n-2}^{(i)})})^{\cdots})^{(b_0^{(i)})},$$

where $W$ is the transmission channel and the transformations $W \rightarrow W^{(0)}$ and $W \rightarrow W^{(1)}$ are the "minus" and "plus" polar transforms formally defined in (2.3) and (2.4) of [7]. Note that the $i$-th component of the message word $u_i$ is connected to the $i$-th component of the codeword $x_i$ by associating a "XOR" to a 0 and a "dot" to a 1 in the binary expansion of the integer $i$. Note also that, by permuting the layers of the factor graph, we simply permute the order of those "XOR"s and "dot"s operations. Hence, the effect of applying a permutation $\pi$ to the binary expansion of $i$ is the same as the effect of applying the same permutation on the layers of the factor graph. $\square$

Fig. 3 shows an example of the proposed mapping applied to a permuted factor graph of $\mathcal{P}(8,5)$. It can be seen that, by using the permuted bit indices, the structure of the factor graph is unchanged. Therefore, the original decoder can be used to perform decoding on all the required permutations. In addition, the proposed mapping allows to use other decoding algorithms, such as SC and SCL, on the permuted factor graphs without changing the decoder structure. This is particularly useful for hardware implementation.

### B. Selection of Good Permutations

First, we observe that BP decoding only fails to decode with a small probability on the original factor graph of polar

**Algorithm 1:** Forming the permutation set

**Input** : $n, k$
**Output** : $\mathbb{P}$
// Fixed layers
1 $p_{fix} \leftarrow \{l_0, l_1, \ldots, l_{n-k-1}\}$
// Permuted layers
2 $p_{per} \leftarrow \{l_{n-k}, l_{n-k+1}, \ldots, l_{n-1}\}$
/* Recursively form the permutation set */
3 $\mathbb{P}_{per} \leftarrow \text{FormPermutation}(p_{per})$
/* Concatenate $p_{fix}$ to each element of $\mathbb{P}_{per}$ */
4 $\mathbb{P} \leftarrow \emptyset$
5 **for** $j \leftarrow 0$ **to** $|\mathbb{P}_{per}| - 1$ **do**
6 $\quad \mathbb{P} \leftarrow \mathbb{P} \cup [p_{fix}, \mathbb{P}_{per}[j]]$
7 **return** $\mathbb{P}$

---

**Algorithm 2:** $\text{FormPermutation}(p_{per})$

**Input** : $p_{per}$ /* Factor graph layer order */
**Output:** $\mathbb{P}_{per}$ /* Set of all permutations */
1 $\mathbb{P}_{per} \leftarrow \emptyset$
2 **if** $length(p_{per}) = 2$ **then**
3 $\quad \mathbb{P}_{per}[0] \leftarrow [p_{per}[0], p_{per}[1]]$
4 $\quad \mathbb{P}_{per}[1] \leftarrow [p_{per}[1], p_{per}[0]]$
5 **else**
6 $\quad$ **for** $i \leftarrow 0$ **to** $length(p_{per}) - 1$ **do**
7 $\quad\quad p'_{per} \leftarrow p_{per}$
8 $\quad\quad l_i \leftarrow p'_{per}[i]$
9 $\quad\quad p'_{per} \leftarrow p'_{per} \setminus l_i$
10 $\quad\quad \mathbb{P}'_{per} \leftarrow \text{FormPermutation}(p'_{per})$
11 $\quad\quad$ **for** $j \leftarrow 0$ **to** $|\mathbb{P}'_{per}| - 1$ **do**
12 $\quad\quad\quad \mathbb{P}_{per} \leftarrow \mathbb{P}_{per} \cup [l_i, \mathbb{P}'_{per}[j]]$

13 **return** $\mathbb{P}_{per}$

---

**Algorithm 3:** BP decoding on predetermined bit-index permutations

**Input** : $\boldsymbol{y}, \mathbb{P}_b$
**Output:** $\hat{\boldsymbol{u}}$
1 $isTerminate \leftarrow False$
2 **for** *all permutations in* $\mathbb{P}_b$ **do**
/* Permute $\boldsymbol{y}$ using Theorem 1 */
3 $\quad$ **for** $i \leftarrow 0$ **to** $N - 1$ **do**
4 $\quad\quad \boldsymbol{y_\pi}[\pi(i)] \leftarrow \boldsymbol{y}[i]$
/* Apply BP decoding on $\boldsymbol{y_\pi}$ */
5 $\quad [\hat{\boldsymbol{u}}_\pi, isTerminate] \leftarrow \text{BPDecoding}(\boldsymbol{y_\pi})$
/* Obtain $\hat{u}$ if early termination */
6 $\quad$ **if** $(isTerminate = True)$ **then**
7 $\quad\quad$ **for** $i \leftarrow 0$ **to** $N - 1$ **do**
8 $\quad\quad\quad \hat{\boldsymbol{u}}[i] \leftarrow \hat{\boldsymbol{u}}_\pi[\pi(i)]$
9 $\quad\quad$ **return** $\hat{\boldsymbol{u}}$

/* If no early termination */
10 **if** $(isTerminate = False)$ **then**
11 $\quad \hat{\boldsymbol{u}} \leftarrow \text{BPDecoding}(\boldsymbol{y})$
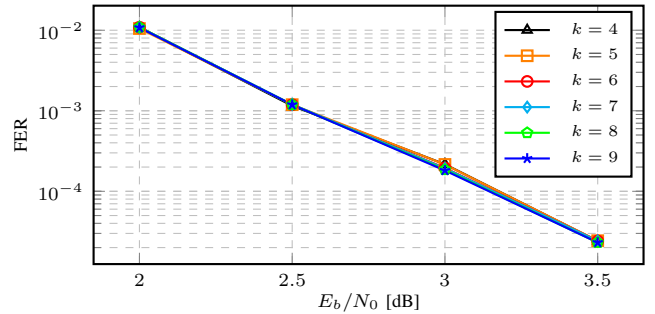12 $\quad$ **return** $\hat{\boldsymbol{u}}$



Fig. 4: FER performance of BP decoding on 16 best permuted factor graphs of $\mathcal{P}(1024, 512)$ with a 24-bit CRC for various sizes of $\mathbb{P}$, where $|\mathbb{P}| = k!$.

codes. Therefore, we keep the original factor graph in the set of good permutations and create a set of permutations that provides the correct codeword when the decoder fails on the original factor graph. To this end, we first provide an approach to construct the permutation set $\mathbb{P}$ of a polar code with $n$ factor graph layers which represents a search space for the good factor graph permutations. This approach is summarized in Algorithm 1, which uses the recursive function $\text{FormPermutation}$ described in Algorithm 2. The main idea is to recursively create $\mathbb{P}$ by permuting only some of the layers on the right-most side of the graph (see Fig. 1a), in order to limit the size of the search space. In other words, we fix the layers $\{l_0, \ldots, l_{n-k-1}\}$, with $0 \leq k < n$, and we consider permutations only of the remaining layers $\{l_{n-k}, \ldots, l_{n-1}\}$. Note that the size of the search space $\mathbb{P}$ is $|\mathbb{P}| = k!$.

At this point, we evaluate numerically the probability of correct decoding for each element in $\mathbb{P}$, when the decoding fails on the original factor graph of polar codes. Then, we select the $M$ elements in $\mathbb{P}$ with the highest probability

of successful decoding and form the subset $\mathbb{P}_b$ of $\mathbb{P}$. Note that $|\mathbb{P}_b| = M$. At the decoder, we consider only the $M$ permutations in $\mathbb{P}_b$. This is described in Algorithm 3. Note that Theorem 1 is used to perform decoding on the permuted bit indices instead of the permuted factor graph layers. In addition, all the BP decoders run a maximum of $I_{\max}$ iterations and an early stopping criteria is used to terminate the BP decoding process if one of the BP decoders satisfies the termination condition. If none of the BP decoders satisfy the early stopping criteria or the maximum number of iterations is reached, the message word given by the original permutation is selected as the final decoding result.

It should be noted that the computational complexity of forming $\mathbb{P}_b$ is directly proportional to the size of $\mathbb{P}$, since all the elements in $\mathbb{P}$ are evaluated numerically. However, this operation is performed only once and is done off-line so there is no complexity overhead in the decoding process.
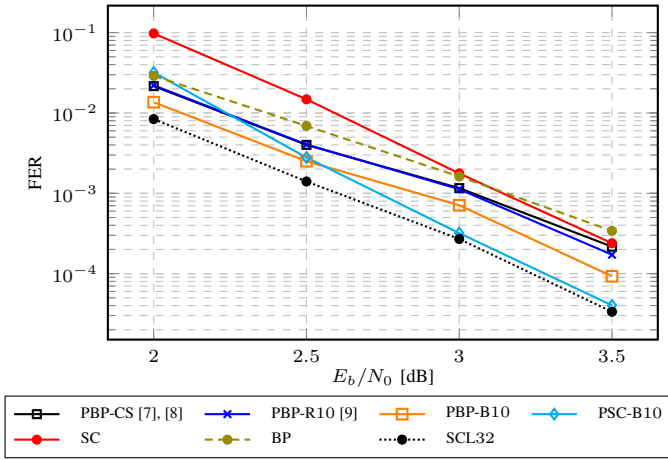
Fig. 5: FER performance of decoding $\mathcal{P}(1024, 512)$ on permuted factor graphs considering $M = 10$ best permutations, in comparison with considering 10 random and cyclic shift permutations, when no CRC is used.

In addition, the error-correction performance of BP decoding on permuted factor graphs selected from $\mathbb{P}_b$ does not change significantly even for small values of $k$. Fig. 4 illustrates the FER curves of applying BP decoding on permuted factor graphs in $\mathbb{P}_b$ to decode $\mathcal{P}(1024, 512)$ with a 24-bit CRC. We set $M = 16$ and we consider different values of $k$ ($k \in \{4, 5, 6, 7, 8, 9\}$) corresponding to different sizes of $\mathbb{P}$ ($|\mathbb{P}| \in \{24, 120, 720, 5040, 40320, 362880\}$). It can be seen that the FER performance of BP decoding on permuted factor graphs for all cases is similar, showing that the good permutations are those obtained by permuting only the layers on the right-most side of the polar codes factor graph for practical values of $k$. Therefore, the size of the search space $\mathbb{P}$ can be significantly reduced, in order to form $\mathbb{P}_b$ with low computational complexity.

## IV. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed technique on the decoding of the polar code $\mathcal{P}(1024, 512)$, which is selected for the 5G eMBB control channel [13], in terms of FER performance and average decoding latency. All the BP decoders considered in this section has the same number of $I_{\max} = 200$ maximum iterations. Note that the BP decoding is early stopped with the $G$-matrix-based termination condition [11] when no CRC is concatenated with the polar code. On the other hand, if a CRC is utilized, the CRC-based termination condition [12] is used for early stopping.

Fig. 5 shows the FER performance of decoding $\mathcal{P}(1024, 512)$ on permuted factor graphs (permuted bit indices) when no CRC is used. Since there are 10 cyclic shift permutations for $\mathcal{P}(1024, 512)$, we used $M = 10$ in order to have a fair comparison with other decoders. In Fig. 5, PBP-CS denotes BP decoding on permuted factor graphs with $n$ cyclic shift permutations [7], [8], PBP-R$M$ denotes BP decoding on permuted factor graphs with $M$ random permutations
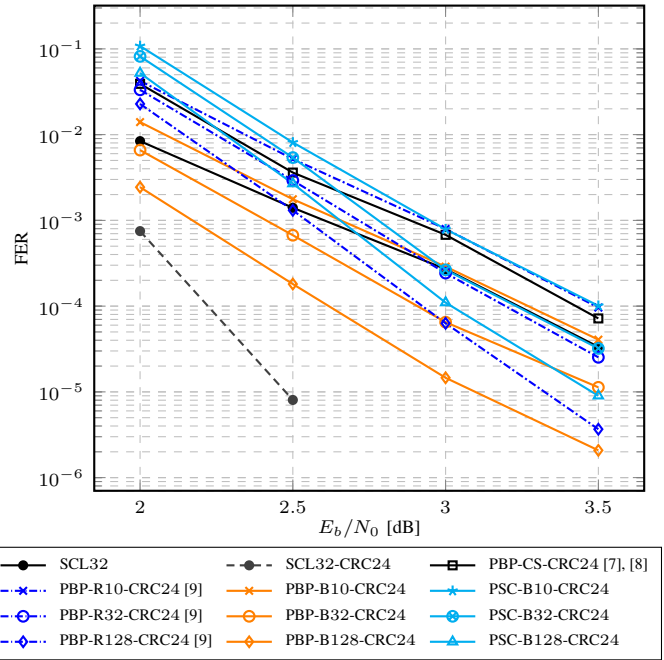


Fig. 6: FER performance of decoding $\mathcal{P}(1024, 512)$ on permuted factor graphs considering $M \in \{10, 32, 128\}$ best permutations, in comparison with considering the same number of random permutations, when a 24-bit CRC is used.

[9], and PBP-B$M$ denotes the proposed BP decoding on permuted factor graphs with $M$ predetermined permutations in $\mathbb{P}_b$. The results are also compared with SC, BP, and SCL decoding with list size 32 (SCL32). It can be seen that while the FER performance of PBP-CS and PBP-R10 are similar, our proposed decoder results in 0.15 dB improvement in comparison with both PBP-CS and PBP-R10 at FER= $10^{-3}$. However, there is still a 0.25 dB gap between the FER performance of the proposed BP decoding on permuted factor graphs and that of SCL32. Using the result of Theorem 1, we performed SC decoding on permuted factor graphs and it can be seen in Fig. 5 that the FER performance gap between using $M = 10$ best permutations for SC decoding (denoted by PSC-B$M$) and that of SCL32 is less than 0.05 dB in the high $E_b/N_0$ region. It should be noted that in order to select the correct codeword among the $M$ SC decoders, we used the absolute LLR value of the last decoded bit as a reliability measure of the decoding process, which is also used in [14].

Fig. 6 presents the FER performance of decoding $\mathcal{P}(1024, 512)$ when a CRC of length 24 is used to aid the decoding process. We use the CRC which is selected to be used in 5G together with polar codes with the CRC polynomial

$$\begin{aligned} \text{CRC24} = & x^{24} + x^{23} + x^{21} + x^{20} + x^{17} + x^{15} + x^{13} + x^{12} \\ & + x^8 + x^4 + x^2 + x + 1. \end{aligned} \tag{5}$$

The curves in Fig. 6 are obtained by considering $M \in \{10, 32, 128\}$ and the suffix -CRC24 is used to denote that the decoding is performed with the CRC. It can be seen that

when $M = 10$, BP decoding with cyclic shift permutations results in a better error-correction performance than when the permutations are selected randomly. Our proposed method results in an additional $0.2$ dB performance improvement over cyclic shift permutations when $M = 10$, at FER$= 10^{-4}$. When $M = 32$ and $M = 128$, BP decoding with the proposed selection of permutations results in around $0.3$ dB improvement in comparison with BP decoding on randomly selected factor graphs. We have also performed SC decoding on the permuted factor graphs by using CRC and it can be seen in Fig. 6 that when adding a CRC, the FER performance of the BP decoder is superior to that of the SC decoder. This is in contrast to the results in Fig. 5 where no CRC is used.

Although the FER performance of the BP decoder on permuted factor graphs can be improved significantly by using the proposed method to select the best permutations, it is still far from that of SCL32-CRC24 for practical values of $M$ as depicted in Fig. 6. However, the parallel nature of BP decoding makes this decoder interesting for applications with stringent latency requirements. For BP decoding, the average decoding latency in the number of time steps can be given as

$$\mathcal{T}_{\mathrm{BP}} = 2nI_{\mathrm{avg}}, \qquad (6)$$

where $I_{\mathrm{avg}}$ is the average number of iterations required to finish the decoding process. We now evaluate the average decoding latency of BP decoding on permuted factor graphs of $\mathcal{P}(1024, 512)$ as shown in Fig. 7, when the permutations are selected based on the method in this paper, in comparison with cyclic shift [7], [8] and random [9] permutations. Note that CRC24 is used to early terminate the decoding when one of the decoders passes the CRC test. It can be seen that the average decoding latency of the proposed method $\mathcal{T}_{\mathrm{PBP\text{-}B}M\text{-}\mathrm{CRC24}}$, is less than 330 time steps for $2 \leq E_b/N_0 \leq 3.5$, and it is always less than the average decoding latency requirements of [7], [8] ($\mathcal{T}_{\mathrm{PBP\text{-}CS\text{-}CRC24}}$) and [9] ($\mathcal{T}_{\mathrm{PBP\text{-}R}M\text{-}\mathrm{CRC24}}$) when $M \in \{10, 32\}$. This latency saving is more significant in the low $E_b/N_0$ region. At $E_b/N_0 = 3$ dB, the average decoding latency of the proposed decoder with $M = 32$ and CRC24 in accordance with (6) is around 100 time steps. In comparison, the SCL32 decoder in [3] requires 2582 time steps to decode $\mathcal{P}(1024, 512)$ with CRC24, and the Fast-SSCL-SPC decoder in [6] requires 673 time steps to decode the same code.

## V. CONCLUSION

In this paper, we first introduced a mapping to represent a polar code factor graph permutation with an equivalent bit-index permutation. This scheme enables a single decoding scheduling that can be applied on different factor graph permutations, which is particularly interesting for hardware implementations. Second, we provided an empirical method to construct the good permutations of polar codes, in order to use only a small number of predetermined permutations to obtain a reasonable error-correction performance. We demonstrated that the error-correction performance of the proposed method applied to belief propagation (BP) decoding can obtain more than $0.25$ dB improvement at the frame error rate of $10^{-4}$,
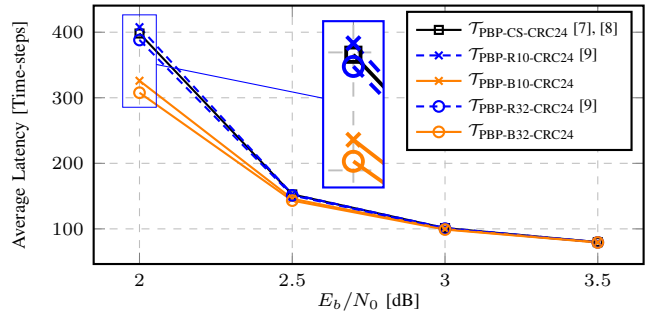


Fig. 7: Comparison between average decoding latency requirements of BP decoding on permuted factor graphs of $\mathcal{P}(1024, 512)$ when the permutations are selected based on the proposed method in this paper and the methods in [7], [8] and [9]. A 24-bit CRC is used for early termination.

in comparison with BP decoding applied to the same number of randomly-selected permuted factor graphs for the 5G polar code of length $1024$ and rate $0.5$, concatenated with a $24$-bit cyclic redundancy check. Significant performance improvements are also observed when the proposed decoder is applied to various polar code configurations with different code lengths, code rates, and CRC lengths.

### 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, July 2009.

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

[3] 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.

[4] S. A. Hashemi, M. Mondelli, S. H. Hassani, R. Urbanke, and W. J. Gross, "Partitioned list decoding of polar codes: Analysis and improvement of finite length performance," in *IEEE Global Commun. Conf. (GLOBECOM)*, Dec 2017, pp. 1–7.

[5] S. A. Hashemi, C. Condo, F. Ercan, and W. J. Gross, "Memory-efficient polar decoders," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 4, pp. 604–615, Dec. 2017.

[6] 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.

[7] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2009.

[8] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *IEEE Int. Symp. on Inf. Theory*, 2009, pp. 1488–1492.

[9] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, "Belief propagation decoding of polar codes on permuted factor graphs," in *IEEE Wireless Commun. and Net. Conf. (WCNC)*, April 2018, pp. 1–6.

[10] E. Arıkan, "Polar codes: A pipelined implementation," in *Proc. 4th Int. Symp. on Broad. Commun.*, 2010, pp. 11–14.

[11] B. Yuan and K. K. Parhi, "Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6496–6506, Dec. 2014.

[12] Y. Ren, C. Zhang, X. Liu, and X. You, "Efficient early termination schemes for belief-propagation decoding of polar codes," in *IEEE 11th Int. Conf. on ASIC (ASICON)*, Nov 2015, pp. 1–4.

[13] 3GPP, "Multiplexing and channel coding (Release 15) 3GPP TS 38.212 v15.0.0." *Valbonne, France*, Dec. 2017. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/38_series/38.212/38212-121.zip

[14] C. Condo, S. A. Hashemi, and W. J. Gross, "Blind detection with polar codes," *IEEE Commun. Letters*, vol. 21, no. 12, pp. 2550–2553, Dec 2017.