Investigation on Information Set Optimization of Neural Network Decoders Suitable for Polar and Convolutional Polar Codes

Yuto Nakamura*, Satoshi Suyama†, Satoshi Nagata†, and Nobuhiko Miki*
*Kagawa University
†NTT DOCOMO, INC.

***Abstract*—One of the possible methods for obtaining reduced latency is the use of neural network decoder (NND) on polar codes. We also apply the convolutional polar codes. We propose an iterative approach to optimize the information set for NND. The proposed approach reduces the block error rate (BLER) performance by exchanging one bit in the information set for one bit in the frozen set. This process is performed until the improvement is saturated. The simulation results show that the BLER performance was enhanced by using the proposed iterative approach for polar codes and convolutional polar codes.**

I. Introduction

Polar codes, introduced by Arıkan in [1], are a new class of channel codes which can achieve the symmetric capacity of memoryless channels asymptotically under successive cancellation decoding (SCD). The polar codes were adopted as the channel codes for control signaling in the 5G mobile communication system. By replacing the block structured polarization step of polar codes by a convoluted structure, the performance at short to moderate block length can be improved, which is called "convolutional polar codes" [2], [3].

Even though the SCD for polar codes and convolutional polar codes can achieve symmetric capacity at a low computational complexity, it is difficult to achieve low latency and high throughput since the decoding relies on the *sequential* algorithm. Research has been conducted to decrease the latency of SCD and successive cancellation list decoding (SCLD) [4], [5].

However, a different strategy is more feasible to reduce the decoding latency. One possible solution is employing the neural network decoder (NND) [6]. In comparison to the SCD and SCLD, the NND finds its estimate by passing each layer only once, which enables low latency implementations, which is called *one-shot* decoding. Although NND is very effective in reducing latency, it suffers from the *curse of dimensionality*, which limits its block length to be short. To overcome this problem, it was proposed that the encoding large graph was partitioning into smaller sub-blocks, which are trained individually. These blocks are then connected via the remaining conventional belief propagation decoding (BPD) [7]. The resulting decoding algorithm is non-iterative and inherently enables a high-level of parallelization.

Such partitioning causes performance degradation. To compensate for the performance degradation, it is effective to construct polar codes and convolutional polar codes (i.e., information set and frozen set) optimized for the NND. However, there exists no explicit construction optimized for other decoding algorithms than SCD. In [8], and [9], the genetic algorithm is used to optimize the information set for SCLD, BPD [10]. In [11], and [12], the information set for BPD is optimized by adding bits to the information set one by one using computer simulation. We proposed the iterative algorithm to optimize the information set for BPD [13]. In [13], one bit in the information set is exchanged for one bit in the frozen set, which decrease the error rate performance. This process is iteratively performed until the performance improvement is saturated.

Although various methods have been proposed, to the best of the author's knowledge, no investigation has been conducted for information set optimization for NND suitable for polar codes. In particular, information set optimization for convolutional polar codes has not been investigated yet. Therefore, in this paper, we optimize the information set in NND suitable for polar codes and convolutional polar codes. Specifically, we optimize the information set using the iterative algorithm proposed in [13]. However, it differs from [13] in that the NND weights depend on the bit-channels in the information set, and thus the NND retraining is applied during the iterative algorithm.

The remainder of this paper is organized in the following manner. We first describe the polar codes and convolutional polar codes briefly in Section II. After describing the NND structure in Section III, Section IV explains the proposed iterative algorithm used in the paper. Section V shows the evaluation results, and finally, conclusions are presented in Section VI.

II. Brief Description of Polar Codes and Convolutional Polar Codes

This section briefly explains the polar codes and conventional polar codes. Both codes generate the $N = 2^n$ polarized synthetic channels by applying n-fold channel transformation. The K most reliable bit-channels are selected to transmit information bits, which is defined as information set, $\mathcal{I}$. The remaining $(N-K)$ bit-channels are set to known bits such as 0, which is defined as frozen set, $\mathcal{F} = \bar{\mathcal{I}}$. Here, $|\mathcal{I}| = K$ and

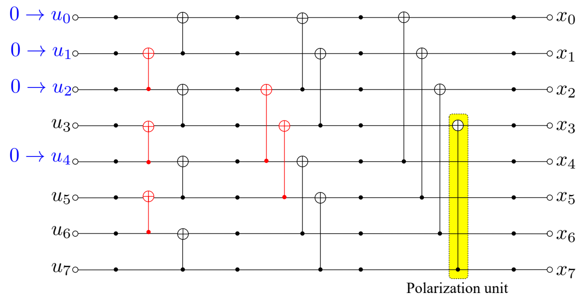


Fig. 1: Polar code and convolutional polar code ($K = 4, N = 8, \mathcal{I} = \{3, 5, 6, 7\}, \mathcal{F} = \{0, 1, 2, 4\}$).

$|\mathcal{F}| = N - K$. To obtain the reliabilities of the bit-channels[1], we employ the Bhattacharyya bound [14]. In this method, the Bhattacharyya parameters, $Z(W_N^{(i)})$, is used to obtain the error rate of the i-th bit-channel, $W_N^{(i)}$. The upper bound on the Bhattacharyya parameters of the bit-channels evolve as simply as

$$\{z, z\} \to \{2z - z^2, z^2\}. \tag{1}$$

In the evolution, z is initialized as $z = \exp(-\gamma_{\rm d})$. Here, $\gamma_{\rm d}$ is the design-SNR.

Figure 1 shows the encoding structure of polar codes and convolutional polar codes when $K = 4, N = 8$, and $\mathcal{I} = \{3, 5, 6, 7\}$. In the figure, only the "polarization units" shown in black is used for Polar codes. In contrast, the convolutional polar codes improve the polarization speed by adding "polarization units" (shown in red) in addition to the black ones. In both codes, the information bits including frozen bits are denoted as $\boldsymbol{u} = (u_0, u_1, \ldots, u_{N-1})$. In the example, $\boldsymbol{u} = (0, 0, 0, u_3, 0, u_5, u_6, u_7)$. The encoded sequence is defined as $\boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1})$. We assume binary phase shift keying (BPSK) modulation and an additive white Gaussian noise (AWGN) channel. The received signal is defined as $\boldsymbol{y} = (y_0, y_1, \ldots, y_{N-1})$.

III. NND Structure

This section briefly describes the NND structure used in the paper, which follows [6]. In the case of convolutional polar codes, it follows [15], which is an extension of NND in [6]. The decoder can be regarded as the K-dimensional binary classification problem with the input of the received signal, $\boldsymbol{y}$.

Figure 2 schematically illustrates the example of NND structure. In this example, the code length is set to $N = 2^3 = 8$. The information set is defined as $\mathcal{I} = \{3, 5, 6, 7\}$, and number of information bits is set to $K = |\mathcal{I}| = 4$. The NND consists of input layer, L-intermediate layers, and output layer. The numbers of nodes in input and output layer are set to N, and K, respectively. The rectified linear unit (ReLU) function was used as the activation function for each intermediate layer

[1]More precisely, it was used to obtain the initial information set for the iterative algorithm described in Section IV.

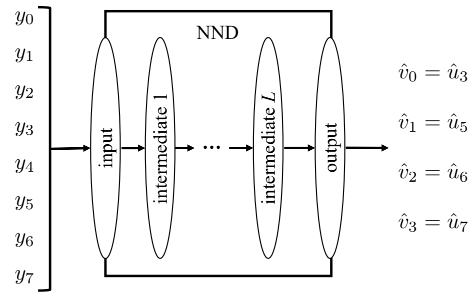


Fig. 2: NND structure ($\mathcal{I} = \{3, 5, 6, 7\}, K = |\mathcal{I}| = 4, N = 8$).

and the sigmoid function for the output layer, as shown in the following equation:

$$g_{\rm relu}(x) = \max\{0, x\}, \quad g_{\rm sig}(x) = \frac{1}{1 + e^{-x}}. \tag{2}$$

We employ batch normalization [16] only for convolutional polar codes based on the results in [15]. It applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.

The K outputs of NND, $x_k^{\rm out}$, correspond to the probabilities of K information bits. Therefore, following equations are used to obtain the decoded bits of the k-th information bits, $\hat{v}_k$.

$$\hat{v}_k = \begin{cases} 1 & (x_k^{\rm out} \geq 0.5) \\ 0 & (\text{otherwise}) \end{cases}. \tag{3}$$

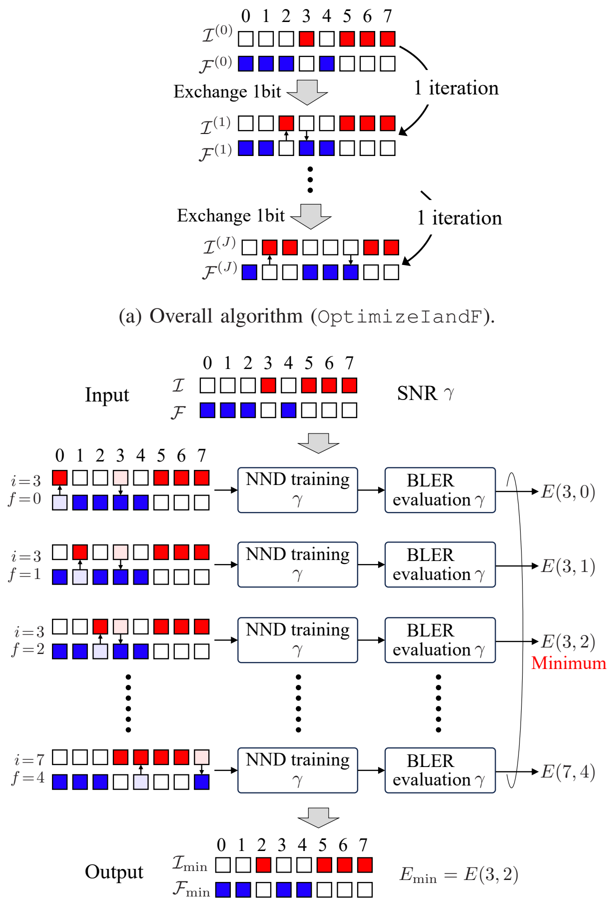
The code structure is learned from the noisy received signals in advance. It is important to note that the weights of the NND varie depending on the bit-channels in the information set. In other words, when using iterative optimization, which is explained in the next section, it is necessary to learn again every time the information set is changed.

IV. Iterative Algorithm

This section explains the proposed iterative algorithm. Figure 3(a) shows the proposed iterative algorithm. In the algorithm, we define $\mathcal{I}^{(j)}$ and $\mathcal{F}^{(j)}$ as the information set, and frozen set after the j-th iteration (let $\mathcal{I}^{(0)}$ be the initial information set before iteration). Let $P_{\rm NND}\,(\gamma, \mathcal{I}, \mathcal{F})$ be defined as the block error rate (BLER) employing NND when the information (frozen) set is $\mathcal{I}$ ($\mathcal{F}$) and the SNR is γ.

According to these definitions, we present the proposed algorithm as `OptimizeIandF`$(\mathcal{I}^{(0)}, \mathcal{F}^{(0)}, \gamma)$ in Algorithm 1. In this algorithm, by inputting the information set, frozen set and SNR, the information set and frozen set are updated to the information set and frozen set with the lowest BLER when exchanging one bit of the information set and frozen set. If BLER, $E^{(j)}$ using the updated information set and the frozen set are the same or greater than the values before the update, $E^{(j-1)}$, the update is stopped. Then, the information set and the frozen set before updating, $\mathcal{I}^{(j-1)}, \mathcal{F}^{(j-1)}$, are output.

The exchange of a single bit between the information set and the frozen set is performed by `UpdateIandF`$(\mathcal{I}, \mathcal{F}, \gamma)$ in Algorithm 2, which is schematically illustrated in Fig. 3(b).



(a) Overall algorithm (`OptimizeIandF`).

(b) Algorithm with one iteration (`UpdateIandF`) .

Fig. 3: Proposed iterative algorithm.

The bit-channel i in the information set $\mathcal{I}$ selected by the for loop in line 2 and the bit-channel f in the frozen set $\mathcal{F}$ selected by the for loop in line 3 are replaced in lines 4 and 5 to obtain a updated information set $\mathcal{I}'$ and frozen set $\mathcal{F}'$. In line 6, the NND is trained assuming information set of $\mathcal{I}'$, since the weights of the NND varie depending on the bit-channels in the information set. In line 7, the BLER employing the NND when the information (frozen) set is $\mathcal{I}'$ $(\mathcal{F}')$ is obtained. After obtaining the BLER using NND, it is compared with the minimum BLER so far, $E_{\min}$. If the obtained BLER is smaller than the minimum value, this information set, the frozen set, is saved as $(\mathcal{I}_{\min}, \mathcal{F}_{\min})$, and the minimum BLER, $E_{\min}$, is updated. This process is performed for all combination of i and f in one updating process. Finally, $(\mathcal{I}_{\min}, \mathcal{F}_{\min})$, and $E_{\min}$ are returned to Algorithm 1.

V. Numerical Evaluation

This section shows the evaluation results. In the evaluation, we perform three evaluations. The detailed simulation parameters are listed in Table I.

Before presenting the numerical evaluation results, we briefly discuss the decoding latency. The SCD requires $\lceil \log_2 N \rceil$ steps to decode one information bit. In addition, the

Algorithm 1 `OptimizeIandF`$(\mathcal{I}^{(0)}, \mathcal{F}^{(0)}, \gamma)$

$E^{(0)} = \infty, j \leftarrow 0$
repeat
  $j \leftarrow j + 1$
  $(\mathcal{I}^{(j)}, \mathcal{F}^{(j)}, E^{(j)}) =$
    `UpdateIandF`$(\mathcal{I}^{(j-1)}, \mathcal{F}^{(j-1)}, \gamma)$
until $E^{(j)} \geq E^{(j-1)}$
return $\mathcal{I}^{(j-1)}, \mathcal{F}^{(j-1)}$

Algorithm 2 `UpdateIandF`$(\mathcal{I}, \mathcal{F}, \gamma)$

1: $E_{\min} \leftarrow \infty$
2: **for each** $i \in \mathcal{I}$ **do**
3:   **for each** $f \in \mathcal{F}$ **do**
4:     $\mathcal{I}' \leftarrow \mathcal{I} \backslash \{i\}, \mathcal{F}' \leftarrow \mathcal{F} \cap \{i\}$
5:     $\mathcal{F}' \leftarrow \mathcal{F} \backslash \{f\}, \mathcal{I}' \leftarrow \mathcal{I} \cap \{f\}$
6:     `TrainNND` $(\gamma, \mathcal{I}', \mathcal{F}')$
7:     $E(i,f) = P_{\text{NND}}(\gamma, \mathcal{I}', \mathcal{F}')$
8:     **if** $E(i,f) < E_{\min}$ **then**
9:       $(\mathcal{I}_{\min}, \mathcal{F}_{\min}) = (\mathcal{I}', \mathcal{F}'), E_{\min} = E(i,f)$
10:     **end if**
11:   **end for each**
12: **end for each**
13: **return** $(\mathcal{I}_{\min}, \mathcal{F}_{\min}, E_{\min})$

K information bits need to be decoded successively. Therefore, the SCD requires $(\lceil \log_2 N \rceil K)$ steps in total. Meanwhile, the NND can decode all information bits at the same time, then requires $(L+2)$ steps corresponding to the total number of layers (input, intermediate, and output layers). Table II shows the decoding latency for both decoding algorithms with the parameters used in this paper. As shown in the table, the NND can effectively reduce the decoding latency compared to the SCD.

We then present the numerical evaluation results of the parameter optimization of NND, followed by the evaluation results of the proposed iterative algorithm.

A. Optimization of NND

As explained in Section II, we generated the initial set for iterative algorithm from the Bhattacharyya bound. We set the design-SNR γ_{d} of 1 dB, and obtained $\mathcal{I}^{(0)} = \{7, 9, 10, 11, 12, 13, 14, 15\}$ for polar codes and convolutional polar codes when $N = 16$, and $\mathcal{I}^{(0)} = \{11, 13, 14, 15, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31\}$ for polar code when $N = 32$.

We first optimize the number of epoch, M_{ep}, and the number of nodes in the intermediate layers, N_{int} employing $\mathcal{I}^{(0)}$. Figure 4(a) and 4(b) show the bit error rate (BER) performance as a function of number of epoch, M_{ep} for polar codes and convolutional polar codes when $K = 8, N = 16$. The number of intermediate nodes N_{int} is parametrized. In this evaluation, the number of intermediate nodes N_{int} was used as a parameter. As shown in both figures, all cases converge to the same BER performance when M_{ep} is larger

TABLE I: Simulation Parameters.

Parameter	Values		
	Polar codes	Polar codes	Convolutional polar codes
Number of information bits, K	8	16	8
Code length, N	16	32	16
Coding rate, R	$1/2$	$1/2$	$1/2$
Number of nodes (input layer)	$N = 16$	$N = 32$	$N = 16$
Number of intermediate layer,L	3	4	4
Number of nodes (intermediate layers)	(N_{int})–$(N_{\text{int}}/2)$–$(N_{\text{int}}/4)$	(N_{int})–$(N_{\text{int}}/2)$–$(N_{\text{int}}/4)$ –$(N_{\text{int}}/8)$	(N_{int})–$(N_{\text{int}}/2)$–$(N_{\text{int}}/4)$ –$(N_{\text{int}}/8)$
Number of nodes (out layer)	$K = 8$	$K = 16$	$K = 8$
Loss function	Mean squared error	Mean squared error	Mean squared error
Batch size	$2^K = 256$	$2^K = 65536$	$(2^K)/4 = 64$
Batch normalization	Not applied	Not applied	Applied
Optimizer	Adam	Adam	Adam

TABLE II: Decoding Latency

(K, N, L)	SCD $(\lceil \log_2 N \rceil K)$	NND $(L+2)$
$(8, 16, 3)$	32	5
$(16, 32, 4)$	80	6

than 2^{18}. Therefore, we used $N_{\text{int}} = 128$ (256) for polar codes (convolutional polar codes), $M_{\text{ep}} = 2^{18}$ in the subsequent evaluation.

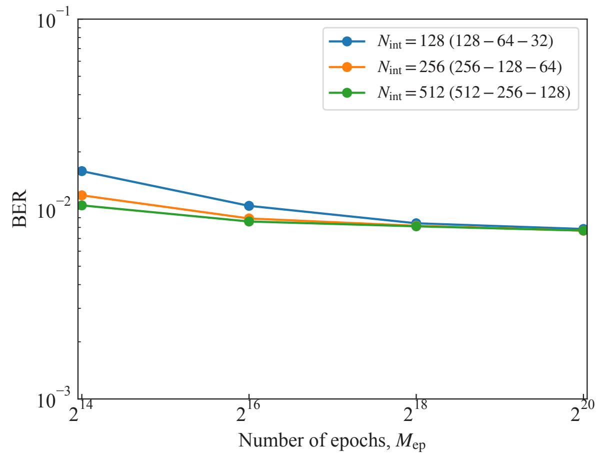
B. Optimization of Information and Frozen Set

We then show the effect of the iterative algorithm on the error rate.

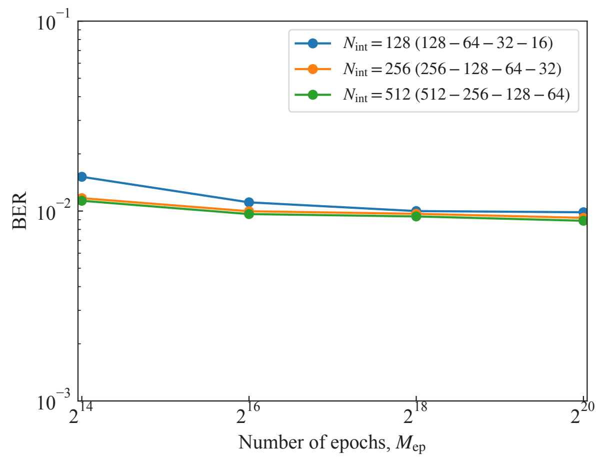
1) Polar code: Figure 5 shows the BLER performance for polar codes when $K = 8$, and $N = 16$. In the figure, the x-axis indicates the bit-channel index i to be removed from the information set, with multiple BLERs shown at the same bit-channel index, i. This is because there are multiple bit-channel indexes f to be added to the information set. The x values for which no error rate is shown are those defined as frozen set. In addition, the dashed line indicates the BLER when $\mathcal{I}^{(0)}$ is used for information set. As shown in Fig. 5(a), the BLER when $i = 10$ and $f = 5$ shows the minimum BLER performance. Furthermore, it shows better performance than that using $\mathcal{I}^{(0)}$. Therefore, the information set is updated to $\mathcal{I}^{(1)} = \{5, 7, 9, 11, 12, 13, 14, 15\}$. In the second iteration in Fig. 5(b), the BLER becomes worse than that using $\mathcal{I}^{(1)}$ (dashed line). Therefore, the iterative algorithm stops, and $\mathcal{I}^{(1)}$ is the optimized information set.

Figure 6 shows the BLER performance for the 1st iteration when $K = 16$, and $N = 32$. In the evaluation, N_{int} is set to 1024. The format of the graph is the same as in Fig. 5. As shown in the figure, the BLER when $i = 24$ and $f = 7$ shows the minimum BLER performance. Furthermore, it shows better performance than that using $\mathcal{I}^{(0)}$. Therefore, the information set is updated to $\mathcal{I}^{(1)} = \{7, 11, 13, 14, 15, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31\}$. We do not see further improvement in the 2nd iteration (performance is not shown in the paper). Therefore, $\mathcal{I}^{(1)}$ is the optimal information set.

Figure 7 shows the BER and BLER performance employing polar codes. As a comparison, we show the performance when using the optimal information set for SCD, optimized using the Bhattacharyya bound, $\mathcal{I}^{(0)}$. As shown in the figure, by applying the proposed algorithm, the BER and BLER is improved especially for large block length $N = 32$.



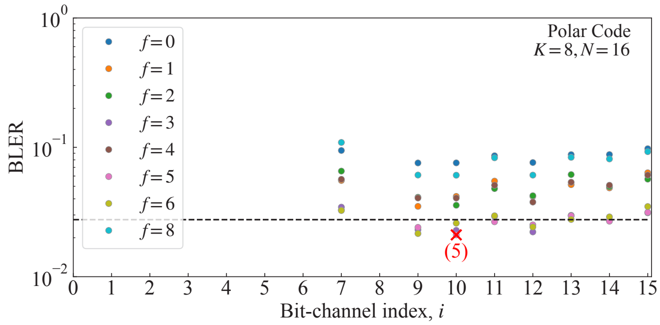
(a) Polar code $(K = 8, N = 16)$.



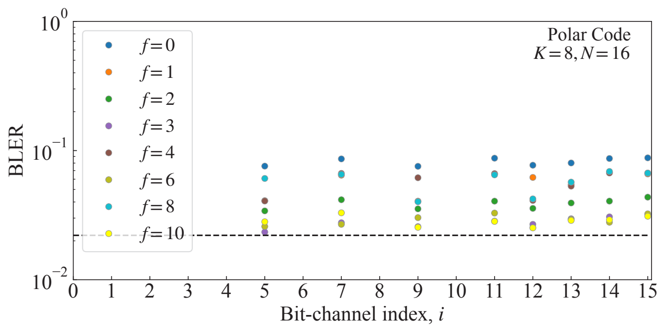
(b) Convolutional polar code $(K = 8, N = 16)$.

Fig. 4: NND optimization.

2) Convolutional polar code: We next evaluate the convolutional polar codes. Figure 8 shows the BLER performance for convolutional polar codes when $K = 8$, and $N = 16$. The format of the graph is the same as in Fig. 5. As shown in Fig. 8(a), and 8(b), the performance is improved in the 1st and 2nd iterations. We do not see further improvement



(a) 1st iteration.



(b) 2nd iteration.

Fig. 5: Performance of proposed iterative algorithm for polar code $(K = 8, N = 16)$.

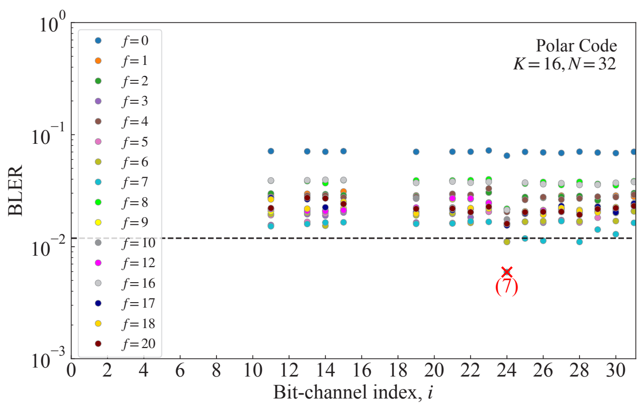
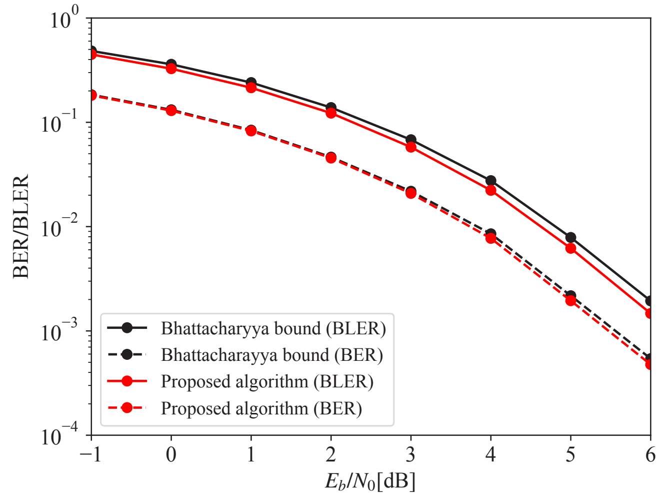


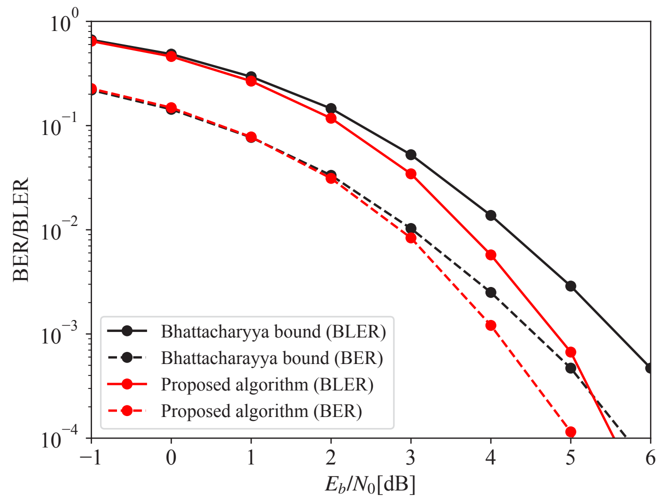
Fig. 6: Performance of proposed iterative algorithm for polar code $(K = 16, N = 32)$ for 1st iteration.

in the 3rd iteration (performance is not shown in the paper). Therefore, $\mathcal{I}^{(2)} = \{3, 8, 9, 10, 11, 13, 14, 15\}$ is the optimal information set. Please note that the optimal information set for convolutional polar code is different from that for polar code.

Figure 9 shows the BER and BLER performance employing convolutional polar codes. Same as Fig. 7, by applying the proposed algorithm, the BER and BLER is improved. Furthermore, the convolutional polar code can slightly improve the performance compared to polar code.



(a) $K = 8, N = 16$.



(b) $K = 16, N = 32$.

Fig. 7: BER and BLER performance for polar code.

VI. Conclusion

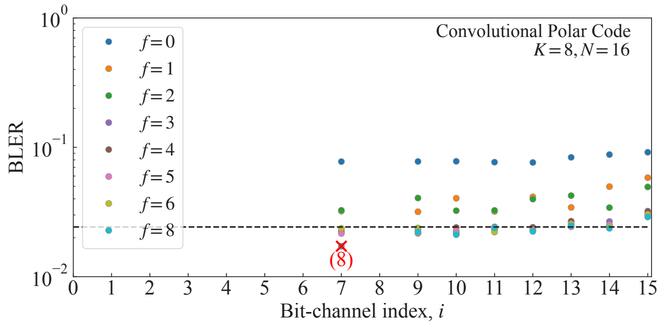
This paper proposed an iterative approach to optimize the information set for NND for polar codes and convolutional polar codes. The proposed approach reduces the BLER performance by exchanging one bit in the information set for one bit in the frozen set. This process is performed until the improvement is saturated. The simulation results showed that the BLER performance was enhanced by using the proposed iterative approach for polar codes and convolutional polar codes.

Acknowledgment

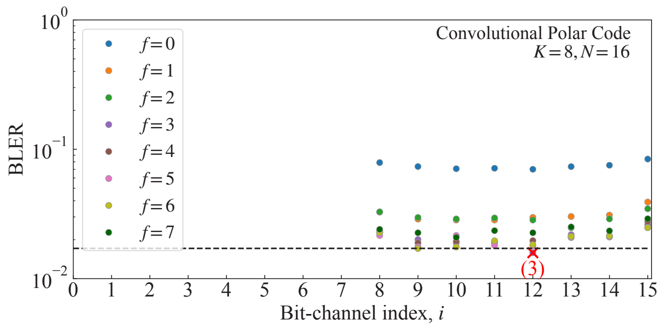
This research and development work was supported by MIC/SCOPE JP225003003.

References

[1] E. Arıkan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.



(a) 1st iteration.



(b) 2nd iteration.

Fig. 8: Performance of proposed iterative algorithm for convolutional polar code $(K = 8, N = 16)$.

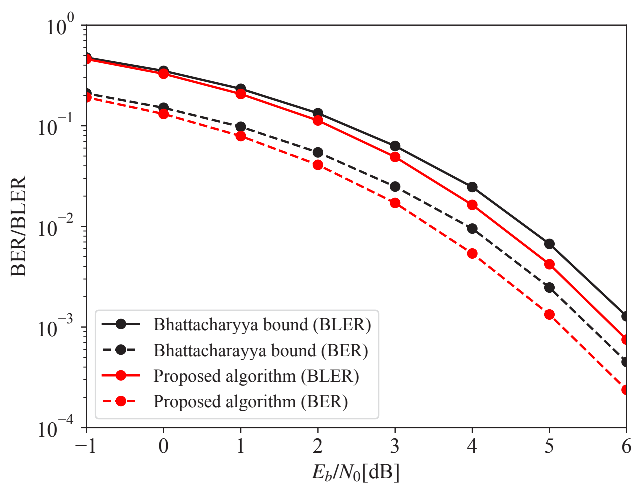


Fig. 9: BER and BLER performance for convolutional polar code.

[2] A. J. Ferris, C. Hirche, and D. Poulin, "Convolutional polar codes," *arXiv:1704.00715*, Apr. 2017.

[3] R. Morozov and P. Trifonov, "Efficient SC decoding of convolutional polar codes," *2018 International Symposium on Information Theory and Its Applications (ISITA)*, Oct. 2018.

[4] C. Zhang, B. Yuan, and K. K. Parhi, "Reduced-latency SC polar decoder architectures," *2012 IEEE International Conference on Communications (ICC)*, Jun. 2012.

[5] C. Xia, J. Chen, Y. Fan, C. ying Tsui, J. Jin, H. Shen, and B. Li, "A high-throughput architecture of list successive cancellation polar codes decoder with large list size," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3859–3874, jul 2018.

[6] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2017.

[7] S. Cammerer, T. Gruber, J. Hoydis, and S. ten Brink, "Scaling deep learning-based decoding of polar codes via partitioning," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[8] A. Elkelesh, M. Ebada, S. Cammerer, and S. t. Brink, "Decoder-tailored polar code design using the genetic algorithm," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4521–4534, 2019.

[9] H. Zhou, W. J. Gross, Z. Zhang, X. You, and C. Zhang, "Low-complexity construction of polar codes based on genetic algorithm," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3175–3179, 2021.

[10] E. Arıkan, "Polar codes : A pipelined implementation," *International Symposium on Broadband Communication 2010*, pp. 11—14, 2010.

[11] J. Liu and J. Sha, "Frozen bits selection for polar codes based on simulation and bp decoding," *IEICE Electronics Express*, vol. advpub, p. 14.20170026, 2017.

[12] S. Sun and Z. Zhang, "Designing practical polar codes using simulation-based bit selection," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 594–603, 2017.

[13] K. Ueda, S. Suyama, S. Nagata, and N. Miki, "Information set optimization for polar codes under bp decoder," in *2023 VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 2023, pp. 1–2.

[14] L. Zhang, Z. Zhang, X. Wang, Q. Yu, and Y. Chen, "On the puncturing patterns for punctured polar codes," in *2014 IEEE International Symposium on Information Theory*, June 2014, pp. 121–125.

[15] R. Maeda, S. Suyama, T. Asai, and N. Miki, "Performance of neural network decoder employing convolutional polar codes," *IEICE Proceeding Series*, vol. 63, pp. J2–2, Dec. 2020.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, 07–09 Jul 2015.