An improved algorithm of generating shortening patterns for polar codes

Original Scientific Paper

Issame El Kaime
Ibn Tofail University
National School of Applied Sciences, Advanced Systems Engineering Laboratory
Issame.elkaime@uit.ac.ma
Kenitra, Morocco

Reda Benkhouya
Ibn Tofail University
Faculty of Sciences, MISC Laboratory, Ibn Tofail University, Kenitra, Morocco
reda.benkhouya@gmail.com
Kenitra, Morocco

Abdessalam Ait Madi
Ibn Tofail University
National School of Applied Sciences, Advanced Systems Engineering Laboratory
abdessalam.aitmadi@uit.ac.ma
Kenitra, Morocco

Soufiane Hajbi
Ibn Tofail University
Faculty of Sciences, LCS Laboratory, Ibn Tofail University, Kenitra, Morocco
abdessalam.aitmadi@uit.ac.ma
Kenitra, Morocco

Abstract *– The rate matching in polar codes becomes a solution when non-conventional codewords of length N≠2n are required. Shortening is employed to design arbitrary rate codes from a mother code with a given rate. Based on the conventional shortening scheme, length of constructed polar codes is limited. In this paper, we demonstrate the presence of favorable and unfavorable shortening patterns. The structure of polar codes is leveraged to eliminate unfavorable shortening patterns, thereby reducing the search space. We generate an auxiliary matrix through likelihood and subsequently select the shortening bits from the matrix. Unlike different existing methods that offer only a single shortening pattern, our algorithm generates multiple favorable shortening patterns, encompassing all possible favorable configurations. This algorithm has a reduced complexity and suboptimal performance, effectively identifying shortening patterns and sets of frozen symbols for any polar code. Simulation results underscore that the shortened polar codes exhibit performance closely aligned with the mother codes. Our algorithm addresses this security concern by making it more difficult for an attacker to obtain the information set and frozen symbols of a polar code. This is done by generating multiple shortening patterns for any polar code.*

Keywords: *Polar codes, Bit error rate, Rate matching, Shortening*

1. INTRODUCTION

Polar codes are linear block correcting codes that have gained significant attention in recent years, thanks to their capacity-achieving properties and low decoding complexity. Polar codes were first proposed by Arikan in 2009 [1] and have since been extensively studied and used in various communication systems. Polar codes were subsequently refined by Arikan to incorporate systematic polar coding [2]. This evolution retains the inherent low-complexity nature of polar codes while additionally conferring notable advantages in terms of bit error rate performance. Polar codes can achieve the theoretical limit of the channel capacity. This is in contrast to other error-correcting codes, such as turbo codes [3] and low-density parity-check codes [4] which can only achieve close to the capacity limit. However, researchers have explored concatenating polar codes with convolutional codes to leverage the performance advantages of both error codes [5]. When building a polar code, the standard approach is to use a 2x2 polarization kernel matrix with a length as a power of two. Nevertheless, in practical applications, the code length could either be a power of two or differ from it. To tackle this challenge, rate-matching techniques such as puncturing, shortening, and repetition have been utilized to modify the encoder output. These techniques provide the flexibility to adjust the code length according to the specific requirements of the application. The study of rate-matching techniques for polar codes has been a widely researched topic in the literature. Some of the most notable approaches include puncturing and shortening techniques.

A study in [6] delves into the fundamentals of puncturing and shortening within polar codes, utilizing binary domination as the core investigative basis. In [7], a novel class of symmetric puncturing patterns was introduced, accompanied by an efficient technique for generating these patterns. More recently, research by Hong et al. in [8] demonstrated the feasibility of achieving channel capacity with punctured polar codes.

Furthermore, in [9], a circular-buffer rate matching (CB-RM) approach based on a two-stage polarization by El Khamy et al. was presented. Wanqi et al. introduced a concise algorithm for selecting shortening patterns (SP). This approach involves creating an auxiliary matrix through bit-reversal permutation, followed by the selection of shortening bits from this matrix. This technique streamlines the process of generating shortened polar codes with ease [10]. a novel encoding hardware architecture for the general construction of polar codes in [11], which supports puncturing and shortening modes. In [12], a proposed algorithm called information set approximation puncturing (ISAP) introduces guard bits before known information bits. The algorithm then calculates the required puncturing pattern based on the actual transmission block length. Tianwei Han and alt. introduced an enhanced puncturing scheme for polar codes, where the initial log-likelihood rate (LLR) of the punctured bits is set to infinity to improve decoding performance [13].

Several other shortening algorithms have been proposed in recent years. One particular algorithm that exemplifies this is the mapping shortening (MS) algorithm, which takes into account the effect that the shortening has on the capacity of split channels [14]. The MS algorithm has been shown to outperform existing shortening algorithms in terms of channel capacity and BER and FER performance, especially under high code rates [2].

The distinction between the existing method and our proposed approach lies in their capacity to generate multiple SP. While the current method is limited to producing a single pattern. This versatility empowers our algorithm to adapt more flexibly to diverse communication scenarios, amplifying the potential for tailored and optimal performance. Furthermore, the proposed algorithm demonstrates a higher level of generality by encompassing and enhancing the capabilities of existing methods, such as Bit-Reversal Shortening (BRS) and Wang-Liu Shortening (WLS). Notably, our approach generates the same patterns that these methods produce. This integration of previously established methods not only showcases the adaptability of our algorithm but also consolidates various techniques under a unified framework. Beyond performance, our proposed method also introduces a crucial dimension of security enhancement. By dynamically altering the position of frozen bits within the generated patterns, our algorithm strengthens the security of the transmission. In scenarios where an eavesdropper intercepts the signal and is aware of the code's length, they are thwarted from accurately identifying the positions of both the information bits and the frozen bits. This fortification adds an extra layer of protection against potential attacks, safeguarding the integrity and confidentiality of transmitted data.

This paper is organized as follows. Section II provides a review of the background of polar codes and emphasizes the significance of SP in polar code design. Section III describes the details of the proposed algorithm and its functionality. Section IV then discusses the simulation results, along with a comparison to conventional shortening algorithms. Finally, section V presents the conclusions.

2. BACKGROUND

Polar codes are a class of error-correcting codes that have gained popularity in recent years due to their superior performance compared to other codes. They were introduced by Arıkan in 2009 [1], and are now integrated into multiple communication standards, including 5th generation wireless systems (5G) [15], The exceptional error correction performance exhibited by Polar codes renders them well-suited for upholding data integrity within satellite communication links [16]. Polar codes have been subject to investigation for the purpose of error correction within storage systems, notably in flash memory, where errors may arise due to physical limitations. This exploration is motivated by the capacity-achieving attributes inherent to Polar codes, which render them an appealing choice for mitigating errors within storage devices [17]. Furthermore, within the field of quantum communication, Polar codes have garnered attention for their potential applications in error correction and the transmission of information through quantum channels [18]. Additionally, the applicability of Polar codes has extended to wireline communication systems, including optical fiber networks, as they offer a mechanism to alleviate transmission errors and enhance the overall performance of such systems [19].

A polar code, in the context of information theory, is a linear block error-correcting code. Its construction involves iteratively concatenating a short kernel code, which transforms the physical channel into outer virtual channels. As the number of iterations increases, the virtual channels tend to polarize, exhibiting either high or low reliability [1]. Consequently, the data bits are allocated to the most reliable channels, leveraging the polarization effect for effective error correction. In a parallel spirit, the integration of polar coding has been harnessed across diverse applications, exemplified by the expansion of Golay codes through the strategic utilization of polar code techniques [20].

2.1. RATE MATCHING IN POLAR CODES

Rate matching is a crucial process in polar codes that ensures compatibility between the source data rate and the channel transmission rate. It involves adapting the number of bits from the coded output to fit the available channel resources. Rate-matching techniques in polar codes are designed to optimize transmission efficiency and reliability. Polar code implementations encompass diverse rate-matching methods, such as puncturing, shortening, and repetition [21].

i. Puncturing

Puncturing is a process in polar codes where S bits are removed from the original codeword with length N, resulting in a shorter codeword of length Ns. These punctured bits are not transmitted, and, as a result, the decoder on the receiving end lacks any stochastic information about them. Hence, their initial Log-Likelihood Ratios (LLR) is typically set to zero.

The impact of puncturing on channel polarization is particularly significant compared to other rate-matching schemes. When certain bits are punctured, it leads to noticeable degradation of symmetric capacities in certain split channels, reducing them to zero [6].

In the next part, we discuss another technique called "shortening", which is a form of rate matching that optimizes the length of polar codes to match the available resources.

ii. Shortening

The novel puncturing scheme described in [13] serves as the second-rate matching technique, which forms the foundation of the shortening process. When a coded bit is shortened, its corresponding input bit channel is transferred to the set of frozen bits, while another bit is moved to the set of information bits. Specifically, the selection of shortened bits (at the output of the polar encoder kernel) is designed such that when a 0 is inputted to the corresponding input bit channels, the shortened bits also become 0.

If the ratio of K to E exceeds 7/16, execute shortening by eliminating bits from the trailing end of the sub-block [22]. This guarantees that the decoder does not lose any information due to the shortened bit, as the LLR for the shortened bit is set to infinity, indicating that it represents a 0. In a bit-reversal shortening scheme, the coded bits are shortened starting from the end in a reverse bit order.

It is important to carefully select the SP to achieve the desired balance between code length reduction and maintained error correction performance. By identifying and exploiting good SP, the overall efficiency of the polar codes can be optimized while still ensuring reliable communication. Good SP ensure that the shortened codes maintain a high level of error correction capability, effectively preserving the reliability of the original codes. On the other hand, "bad" SP refer to those configurations in which the performance of the shortened polar codes is significantly lower compared to the original (or "mother") codes.

In the upcoming section, we delve into another technique known as "repetition," which serves as a form of rate matching.

iii. Repetition

If the available resource G is slightly larger than the length $N=2^{\lceil log_2 G\rceil-1}$ of polar codes N, it would be beneficial to use a polar code of length N along with repetition. This approach helps avoid performance degradation caused by excessive puncturing compared to selecting a polar code of length $2^{\lceil log_2 G\rceil}$. Additionally, it reduces the complexity of encoding and decoding by approximately half [6].

During transmission, $G-N$ bits in the codeword can be generated by repeating certain bits in the original message X. Before decoding at the receiver, the LLRs of the repeated bits are combined with those of the corresponding original bits. This LLR combining technique enhances the effective bitwise channels associated with the repeated bits. Choosing a repetition pattern is a crucial factor that influences the decoding performance [6]. The upcoming section delves into a comprehensive discussion of the proposed method for generating favorable SP for polar codes. The details of this approach are thoroughly explained.

3. THE PROPOSED ALGORITHM

In this section, the primary objective is to demonstrate the existence of both favourable and unfavourable SP. Subsequently, we examine the common characteristics shared among all good SP. We then formulate the characteristics of good SP in mathematical terms. Finally, we present the algorithm for generating SP. The algorithm outlines the step-by-step process through which these patterns are generated.

3.1. GOOD AND BAD SHORTENING PATTERNS

Through our analysis, we have identified the presence of two distinct types of SP. The first type demonstrates performance closely resembling that of the original mother code, suggesting that the shortened polar codes retain similar levels of error correction capabilities. This suggests that these patterns effectively preserve the reliability and performance of the original code. However, the second type of shortening pattern demonstrates a significant deviation from the mother code's performance. The decoding performance of the shortened polar codes associated with these patterns is noticeably poorer compared to the original code. As a result, these patterns are considered less favourable in terms of error correction capabilities.

The clear distinction between these two types highlights the existence of good and bad SP. It emphasizes the need to identify and use SP that closely align with the desired performance requirements and error correction objectives. By opting for SP that closely resemble the mother code's performance, we can effectively maintain the reliability and efficiency of the polar codes. After retrieving good and bad SP, we proceed next to the analysis of the common characteristics among all the good SP.

3.2. THE COMMON CHARACTERISTICS AMONG ALL GOOD SHORTENING PATTERNS

The objective of this part is to identify and analyze the shared characteristics among all good SP. Subsequently, we aim to develop a mathematical model capable of generating good SP systematically. By examining multiple instances of good SP, we can observe recurring features that contribute to their effectiveness. Through a comprehensive analysis, our aim is to identify the shared characteristics exhibited by these patterns. Once these common characteristics are identified, our next goal is to construct a mathematical model that can reliably generate these patterns.

The process of shortening in polar coding involves assigning values, which are known to the decoder in advance, to bits that are not transmitted. As a result, the shortened bits possess bit-channels with likelihoods of ∞. After extensive analysis, we have made a notable observation during the decoding stage. Specifically, we have identified a distinctive characteristic in the matrix of likelihood ratios. It has come to our attention that all good SP exhibit the same row value in the position where the shortening occurs. This finding suggests a strong correlation between the row value and the effectiveness of SP. Regardless of the specific shortening position, we consistently observe this shared characteristic among the identified good patterns. By recognizing this common characteristic, we can further optimize the design and implementation of SP within the decoding process. Understanding the relationship between the row value and the effectiveness of SP opens up possibilities for more efficient and reliable decoding techniques, ultimately enhancing the overall performance of the system. After identifying the shared characteristics of good SP, the following paragraph focuses on modeling this concept with the aim of automatically generating good SP.

3.3. FORMULATING THE MATHEMATICAL CHARACTERISTICS OF GOOD SHORTENING PATTERNS

Based on the observations made in the previous section, we have identified a common characteristic among all good SP. To formalize this idea mathematically, we propose a modeling approach that extends the concept of decoding but with a flexible operation. In the initialization stage of the likelihood matrix, we set the values of elements in the first column to be arbitrary, except for the position corresponding to the shortened bit, where the value is set to infinity. Subsequently, the operation for computing the remaining elements follows a similar logic to SC as described in Algorithm 2. The objective is to compute all the elements of the matrix.

During the computation stage, if at least one element in the shortening row is different from infinity, it indicates that the proposed pattern is not good. On the other hand, if all the bits in the shortening row are infinity, it signifies that the proposed shortening pattern is good. By leaning on this mathematical model, we can evaluate the quality of different SP based on the values in the likelihood matrix.

3.4. PROPOSED METHOD STEPS

Our algorithm aims to find suitable SP for polar codes with a code length of N (where N is power of two) and a specified number of shortened bits S. The first step of the algorithm involves generating all possible SP, denoted as Pi. Each pattern Pi is a vector of length S, where the elements are natural numbers lower than N-1, and N-1 itself must be an element of the Pi.

After generating the SP, the algorithm proceeds by selecting the first pattern, denoted as Pi, to examine. Then, the algorithm initializes the likelihood matrix, M, which has a size of $N*(n+1)$, where $n=log_2(N)$.

In the first column of matrix M, arbitrary numbers are assigned to the elements, except for the elements $m_{(i1)}$ ($i \in Pi$) corresponding to the shortened bits. For those elements, their values are set to infinity, as shown in equation (1).

$$m_{(i1)} = \{\infty \; if \; i \in \; P_i \quad arbitry \;\; if \; i \notin \;\; P_i \tag{1}$$

The first value of matrix M is given by equation (2), where "nan" represents the placeholder for unspecified values, and "inf" represents infinity:

$$\begin{pmatrix} m_{01} & nan & nan & nan \\ inf & nan & nan & nan \\ m_{03} & nan & nan & nan \\ m_{04} & nan & nan & nan \\ inf & nan & nan & nan \\ m_{06} & nan & nan & nan \\ m_{07} & nan & nan & nan \\ inf & nan & nan & nan \end{pmatrix} \tag{2}$$

We start calculating the elements for the matrix M using Algorithm 3 until all members have been assigned a value or until one member in the shortening row reaches a non-infinite value. The shortened row is a specific row in the matrix M. This row refers to the positions of the shortening bits; it is visually emphasized with a red highlight in (2). The bad rate matching patterns refer to those patterns in which at least one element of the shortened row in the corresponding matrix M is different from infinity, as illustrated in (3) where $m_{43} \neq inf$.

$$\begin{pmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ inf & inf & inf & inf \\ m_{20} & m_{21} & m_{22} & inf \\ m_{30} & m_{31} & m_{32} & inf \\ inf & inf & inf & \boxed{m_{43}} \\ m_{50} & m_{51} & m_{52} & m_{53} \\ m_{60} & m_{61} & m_{62} & m_{63} \\ inf & inf & inf & inf \end{pmatrix} \tag{3}$$

The effective SP are characterized by having all elements of the shortening row in the corresponding matrix equal to infinity, as showed in (4).

$$\begin{pmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ inf & inf & inf & inf \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \\ inf & inf & inf & inf \\ m_{50} & m_{51} & m_{52} & m_{53} \\ m_{60} & m_{61} & m_{62} & m_{63} \\ inf & inf & inf & inf \end{pmatrix} \quad (4)$$

For a detailed understanding of the approach, the proposed Algotihm 3 gives a step-by-step process to identify favorable SP. For simplicity reasons, the main Algorithm 3 uses Algorithm 2 which in turn calls the Algorithm 1. Note that Algorithm 1 determines the position of the first "1" in the binary representation of the input number, which is required by Algorithm 2 to compute elements of the LLR matrix *M*.

Algorithm 1: Get active LLR

```
Input = (i,n)
Mask=2n
C=1
for k =1 to n
        If Mask ==0 and i ==0
                C=C+1
                mask >>= 1
        Else
                Break
Return the min between C and n
```

Algorithm 2: Perform LLR update

```
Input :N,M
M=Matrix of LLR
N=the mothercode block length
Output:
        Shortening patterns state
n= int(log2(N))
a= n - activeLLR(l, n)
for s = a to n
        bl = 2^(s+1)
        bs =the remainder of the euclidean division
        between bl and 2
        for j from 1 to n step=bs

            if (j % bl) < bs
                top= M[j, s]
                btm=M[j+bs, s]
                if M[j, s]== inf
                    M[j, s + 1] = M[j + bs, s]
                    return "bad"
                else
                    M[j, s + 1] = inf
            else
                btm = M[j, s]
                top = M[j - bs, s]
                bit = B[j - bs, s + 1]
                if M[j, s]== inf
                    if M[j - bs, s]!=inf
                        M[j, s + 1] = inf-M[j - bs, s]
                        if M[j, s + 1]!= inf:
                            return "bad"
                    else
                        M[j, s + 1]=M[j, s]
                        if M[j, s + 1]!= inf:
                            return "bad"
                else
                    M[j, s + 1] = M[j, s]
```

Algorithm 3: Identify favourable shortening pattern

```
Input: N, S
    N=the mothercode block length
    S= the number of shortened bits
n= int(log2(N))
P= Combination of (S-1) elements from 0 to (N-2)
Append N-1 to each P_j in P
For P_j in P
    M={mij,N*(n+1)}
    m(i,1)=inf if i ∈ P_j
    m(i,1)=arbitray number if i ∉ P_j
    For i=1 to N
        For l in Bit-reversal (i,n)
            if update_llrs(l,n,M)=="bad":
                Return bad
        return Good
```

For making the proposed algorithm easy to follow, the sequencing of steps is illustrated visually in Fig. 1.

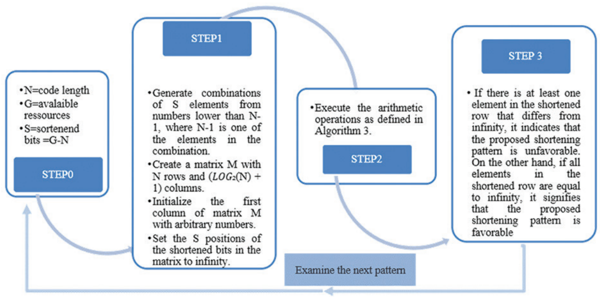

Fig 1. The simplified process of the proposed procedure

3.5. COMPARAISON OF LDPC AND POLAR CODES

LDPC codes outperformed polar codes for longer messages, while polar codes outperformed LDPC codes for shorter messages. This is because LDPC codes are better at correcting burst errors, which are common in long messages, while polar codes are better at correcting random errors, which are more common in short messages [23]. In 5G and beyond wireless communication systems, data channels typically carry long messages, such as streaming video and file transfers. Control channels, on the other hand, typically carry short messages, such as signaling and configuration information. Therefore, LDPC codes are a better choice for data channels, while polar codes are a better choice for control channels [23]. Gaussian noise dominates the BER error at lower SNR values because it is a more fundamental source of noise in electronic circuits. Phase estimation error dominates the BER error at higher SNR values as it is caused by imperfections in the oscillators used to generate and synchronize the carrier signals.

In addition to the above, here are some other factors to consider when choosing between LDPC and polar codes:

- Complexity: LDPC codes are more complex to design and decode than polar codes.
- Latency: LDPC codes can have higher latency than polar codes.
- Power consumption: LDPC codes can consume more power than polar codes.

Overall, LDPC and polar codes are both promising channel coding techniques for 5G and beyond wireless communication systems [23]. The best choice of code for a particular application will depend on a number of factors, including the required code rate, the channel conditions, the complexity constraints of the system, and the latency and power consumption requirements

3.6. HARDWARE IMPLIMENTATION

The proposed shortening pattern generation algorithm can be implemented in hardware on an FPGA or ASIC. This would allow for high-throughput SP generation, which is important for applications such as real-time polar coding.

One way to implement the proposed algorithm in hardware is to use a parallel processing architecture. This would allow multiple SPs to be generated simultaneously, which would improve the throughput of the algorithm.

Another way to improve the performance of the algorithm in hardware is to use pipelining. This would allow the algorithm to start processing the next input while the previous input is still being processed.

The proposed algorithm is also relatively memory-efficient, which makes it suitable for implementation in hardware. The algorithm only requires a small amount of memory to store the mother code and the shortening patterns that have been generated.

The proposed algorithm can be implemented in hardware using PYNQ. PYNQ is an open-source framework that makes it easy to develop and deploy Python applications on FPGAs.

There are several advantages to implementing the proposed algorithm in hardware using PYNQ:

- Performance: FPGAs can provide significantly higher performance than software implementations of the proposed algorithm.
- Efficiency: FPGAs can be configured to efficiently implement the proposed algorithm, which can lead to reduced power consumption and increased battery life.
- Flexibility: PYNQ allows users to develop and deploy Python applications on FPGAs, which provides a high degree of flexibility and control.

3.7. WIRETAP CHANNEL SECURITY

A wiretap channel is a communication channel where an attacker can eavesdrop on the communication between the sender and the receiver. Wiretap channels are a major security concern, as they can allow attackers to steal sensitive information or to disrupt communication altogether.

Conventional shortening schemes for polar codes suggest only a single shortening pattern, which is known to both the receiver and attackers. The proposed shortening algorithm addresses this security concern by generating multiple favorable shortening patterns for any polar code, encompassing all possible favorable configurations. This makes it more difficult for an attacker to determine the information set and frozen symbols of a polar code, even if they are able to eavesdrop on the communication. if an attacker is able to eavesdrop on the communication, they will only be able to see the shortened polar code. However, they will not be able to determine which shortening pattern was used, as the sender has generated multiple shortening patterns. This makes it much more difficult for the attacker to exploit any weaknesses in the coding scheme. The proposed algorithm can be used to generate polar codes with a high minimum distance, which would make it more difficult for an attacker to decode the code.

4. RESULTS AND DISCUSSION

In this section, we apply the proposed algorithm to generate good SP for different code lengths. We then evaluate the performance of these patterns in terms of BER and FER.

Fig. 2 and Fig. 3 represent respectively the curves of Bit Error Rate (BER) and Frame Error Rate (FER) versus energy per bit to noise power spectral density ratio (Eb/No) for all possible shortening polar codes with a code length of $N_s = 30$ and information length of $K = 25$. The analysis of Fig. 2 and Fig. 3 reveals the existence of two distinct types of SP.

The first type of SP exhibits performance that closely resembles that of the mother code, However, the second type of SP demonstrates a significant deviation from the mother code's performance. The decoding performance of the shortened polar codes associated with the second type of patterns is noticeably worse compared to the original.

It is essential to emphasize that the selected parameters for this example (N = 32, K = 25, Ns = 30) serve to demonstrate the evaluation process. Nevertheless, the analysis can be applied to polar codes of varying lengths

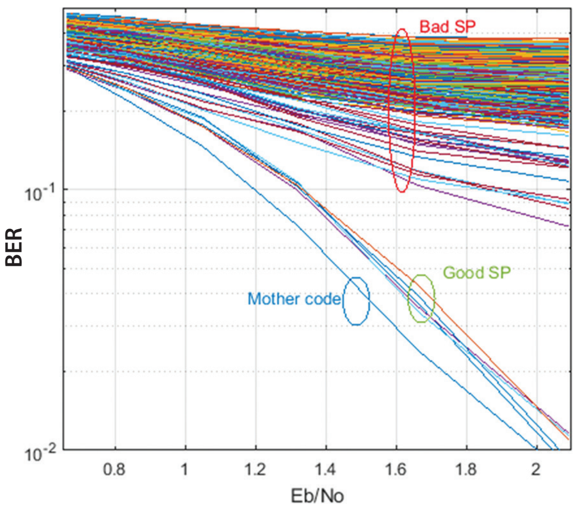

Fig. 2. Comparative Analysis of BER vs. Eb/No for various Rate Matching Patterns where Ns=30, K=25.

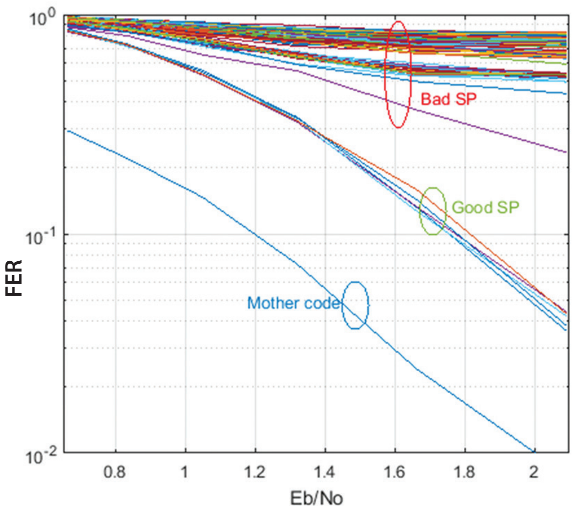

Fig. 3. Comparative Analysis of FER vs. EB/NO for various Rate Matching Patterns where Ns=30, K=25.

To confirm the last proposition, we conducted another test using all SP of polar code with a length of 15 and an information length of 12. From Fig. 4, which represents the Bit Error Rate (BER) versus Eb/No for all possible SP, we observe the existence of two distinct pattern types. The first type of patterns exhibited performances that were close to those of the mother codes. However, the second pattern showed significantly lower performance compared to the mother codes.

Based on the last figures, we have confirmed the existence of good and bad SP within polar codes. These findings underscore the criticality of thoughtful design and careful selection of patterns when utilizing polar codes. Fig. 5 depicts the BER of the proposed SP for a polar code length of 13. In order to provide a more comprehensive evaluation of our algorithm's performance, we also considered bad SP for comparison. The purpose of evaluating these bad patterns was to highlight the difference in performance between them and the proposed patterns.

Therefore, we conducted simulations of the Bit Error Rate (BER) for the proposed SP with a polar code length of 13, as depicted in Fig. 5.

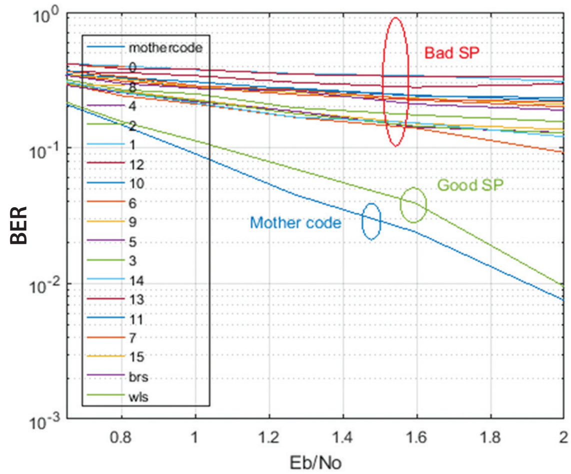

Fig 4. Comparative Analysis of BER vs. Eb/No for various Rate Matching Patterns where Ns=15, K=12.

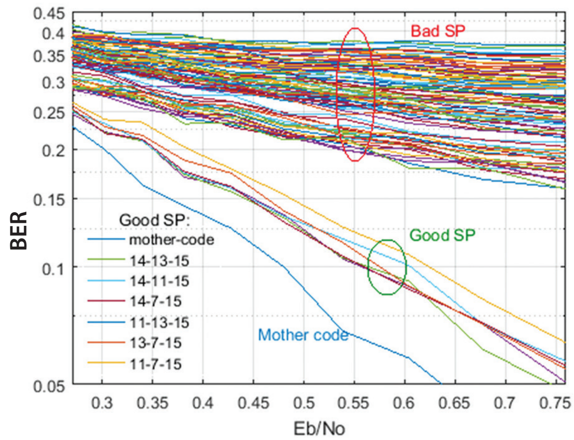

Fig 5. Comparative Analysis of BER vs Eb/No for various Rate Matching Patterns where Ns=13, K=7

The results in Fig. 5 indicate that the patterns produced by the proposed algorithm and mentioned in Table 1 demonstrate favorable BER values across various Eb/No, where Table 1 presents the set of good SP generated by our algorithm for different code lengths. This suggests that the proposed algorithm effectively generates SP that exhibit good BER performance.

Table 1 Good shortening patterns for different code lengths

Polar Code length	Good shortening patterns	BRS	WLS
2046	[1023, 2047], [1535, 2047], [1791, 2047], [1919, 2047], [1983, 2047], [2015, 2047], [2031, 2047], [2039, 2047], [2043, 2047], [2045, 2047], [2046, 2047]	[1023,2047]	[2046,2047]

1022	[511, 1023], [767, 1023], [895, 1023], [959, 1023], [991, 1023], [1007, 1023], [1015, 1023], [1019, 1023], [1021, 1023], [1022, 1023]	[511, 1023]	[1022,1023]
63	[63]	[63]	[63]
31	[31]	[31]	[31]
30	[15, 31], [23, 31], [27, 31], [29, 31], [30, 31]	[15,31]	[30, 31]
14	[7, 15], [11, 15], [13, 15], [14, 15]	[7,15]	[14,15]
6	[3, 7], [5, 7], [6, 7]	[3, 7]	[6, 7]
5	[3, 5, 7], [3, 6, 7], [5, 6, 7]	[5 3 7]	[5, 6, 7]
7	[7]	[7]	[7]
13	[7, 11, 15], [7, 13, 15], [7, 14, 15], [11, 13, 15], [11, 14, 15], [13, 14, 15]	[11, 7, 15]	[13, 14, 15]
61	[31, 47, 63], [31, 55, 63], [31, 59, 63], [31, 61, 63], [31, 62, 63], [47, 55, 63], [47, 59, 63], [47, 61, 63], [47, 62, 63], [55, 59, 63], [55, 61, 63], [55, 62, 63], [59, 61, 63], [59, 62, 63],[61, 62, 63]	[47, 31, 63]	[61, 62, 63]
29	[15, 23, 31], [15, 27, 31], [15, 29, 31], [15, 30, 31], [23, 27, 31], [23, 29, 31], [23, 30, 31],[27, 29, 31], [27, 30, 31], [29, 30, 31]	[23,15,31]	[29, 30, 31]
15	[15]	[15]	[15]

Results depicted in Fig. 6 present BER for all SP of polar codes with a code length of Ns=30. These results demonstrate that certain patterns exhibit BER values very close to those of the original codes. These specific patterns are also generated by our algorithm and are listed in Table 1.

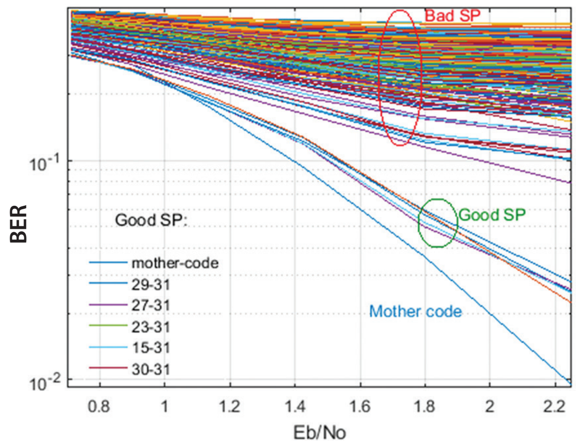

Fig 6. Comparative Analysis of BER vs. Eb/No for various Rate Matching Patterns where Ns=30,K=27

In contrast, the suboptimal patterns, which were not generated by our algorithm, exhibit poor performance. The results presented in all figures demonstrate that the patterns generated by the proposed algorithm present good performance. These findings reinforce the effectiveness of our algorithm in producing high-quality SP that maintain the performance of the mother codes, even when dealing with diverse code lengths.

To provide further evidence of the effectiveness of our algorithm in generating good SP for polar codes, we conducted a comparison with two existing methods: Bit-Reversal Shortening (BRS) and Wang-Liu Shortening (WLS). The Fig. 7 illustrates the comparison of BER of the SP generated by the proposed algorithm and those generated by the two existing methods (BRS and WLS).

The results provide compelling evidence that show that the performance of the proposed patterns is very similar to the patterns generated by the algorithms of WLS and BRS. Based on Fig. 7., it is evident that the patterns proposed by the BRS method are 6-5-7, while the WLS method suggests the pattern 5-3-7.

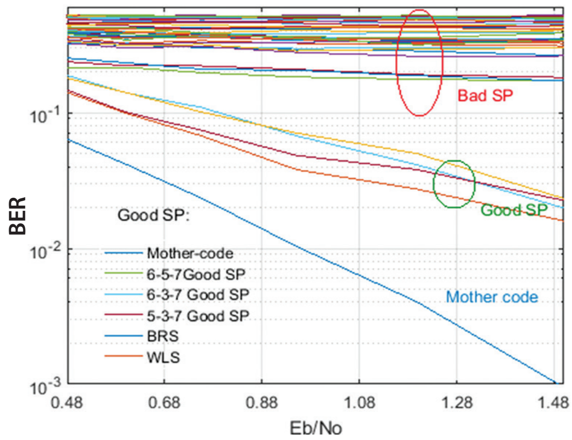

Fig 7. Comparative Analysis of BER vs. Eb/No for various Rate Matching Patterns where Ns=5, K=3.

Remarkably, our algorithm not only reproduces these patterns but also generates various others like 6-3-7.

A comparison between the patterns listed in Table 1 and the SP produced by the BRS and WLS methods for different codes length reinforces our algorithm's capacity. From Table 1, it is evident that the SP provided by both algorithms BRS and WLS are also suggested by the proposed method.The results demonstrate the effectiveness of the proposed algorithm in generating high-quality SP for polar codes.

5. CONCLUSION

The proposed algorithm outperforms existing algorithms in terms of flexibility for practical applications because the ability to generate multiple SPs allows it to select the SP that is best suited for the specific application requirements. This can lead to improved performance over existing algorithms, which typically generate a single SP.

The proposed algorithm outperforms existing algorithms in terms of security due to its the ability to generate multiple SPs and dynamically change the position of frozen bits. This can make it more difficult for eavesdroppers to intercept and decode polar code transmissions. Additionally, the proposed algorithm introduces randomness in pattern selection, which further enhances confidentiality.

Finally, the proposed algorithm outperforms existing algorithms in terms of complexity as it does not require the computation of the reliability of each channel. This reduces the complexity of the algorithm and makes it more suitable for high-throughput applications. The difference in performance between the proposed and existing methods is due to the fact that the proposed algorithm tests all possible shortening patterns, while existing methods only provide one.

However, it is important to note that the proposed algorithm requires further improvement in the future to develop mechanisms for synchronizing the employed patterns between senders and receivers.

6. REFERENCES

[1] E. Arikan, “Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels”, IEEE Transactions on Information Theory, Vol. 55, No. 7, 2009, pp. 3051-3073.

[2] E. Arikan, “Systematic Polar Coding”, IEEE Communications Letters, Vol. 15, No. 8, 2011, pp. 860-862.

[3] D. Divsalar, F. Pollara, “On the Design of Turbo Codes”, Telecommunications and Data Acquisition Progress Report, Vol. 42, 1995.

[4] R. Gallager, “Low-density parity-check codes”, IEEE Transactions on Information Theory, Vol. 8, No. 1, 1962, pp. 21-28.

[5] A. Agarwal, S. N. Mehta, “PC-CC: An advancement in forward error correction using polar and convolutional codes for MIMO-OFDM system”, Journal of King Saud University - Computer and Information Sciences, Vol. 32, No. 8, 2020, pp. 917-927.

[6] M. Jang et al. “Rate matching for polar codes based on binary domination”, arXiv:1901.02287, 2019.

[7] L. Chandesris, V. Savin, D. Declercq, “On Puncturing Strategies for Polar Codes”, arXiv:1701.04746, 2017.

[8] S.-N. Hong, D. Hui, I. Marić, “Capacity-Achieving Rate-Compatible Polar Codes”, IEEE Transactions on Information Theory, Vol. 63, No. 12, 2017, pp. 7620-7632.

[9] M. El-Khamy, H.-P. Lin, J. Lee, I. Kang, “Circular Buffer Rate-Matched Polar Codes”, IEEE Transactions on Communications, Vol. 66, No. 2, 2018, pp. 493-506.

[10] W. Li et al. “A shortening pattern selection method of shortened Polar codes”, Proceedings of the 19th International Conference on Optical Communications and Networks, Qufu, China, 23-27 August 2021, pp. 1-3.

[11] W. Song, Y. Shen, L. Li, K. Niu, C. Zhang, “A General Construction and Encoder Implementation of Polar Codes”, arXiv:1903.02899, 2019.

[12] J. Zhao, W. Zhang, Y. Liu, “A Novel Puncturing Scheme of Low Rate Polar Codes Based on Fixed Information Set”, IEEE Communications Letters, Vol. 25, No. 7, 2021, pp. 2104-2108.

[13] T. Han, M. Fu, M. Zhang, G. Hu, “Puncturing scheme for polar codes based on channel reliability estimation”, Telecommunication Systems, Vol. 82, 2023, pp. 499-508.

[14] H. Zhu, Y. Zhao, “A Mapping Shortening Algorithm for Polar Codes”, IEEE Access, Vol. 7, 2019, pp. 87110-87117.

[15] N. Radha, M. Maheswari, “An empirical analysis of concatenated polar codes for 5G wireless communication”, Telecommunication Systems, 2023, pp. 1-24. (in press)

[16] A. C. Vaz, N. T. Hegde, A. A. Shetty, “Performance analysis of Polar Codes in a Visible Light Communication System”, Journal of Physics: Conference Series, Vol. 2466, 2023, p. 012006.

[17] N. K. Gerrar, S. Zhao, L. Kong, “Error correction in data storage systems using polar codes”, IET Communications, Vol. 15, No. 148, 2021, pp. 1859-186.

[18] X. Chen, W. Qian, K. Tao, Y. Wang, W. Wang, Y. Wang, “Design of Polar Turbo Product Codes for Optical Fiber Communications Based on FPGA”, Proceedings of the Asia Communications and Photonics Conference, Shenzhen, China, 5-8 November 2022, pp. 968-971.

[19] T. Ahmad, F. Khan, “On the Coding Gain of Polar Codes for High Speed Communications over AWGN Channels”, Proceedings of the 17th International Bhurban Conference on Applied Sciences

and Technology, Islamabad, Pakistan, 14-18 January 2020, pp. 690-694.

[20] D. Khebbou, I. Chana, H. Ben-Azza, "Decoding of the extended Golay code by the simplified successive-cancellation list decoder adapted to multi-kernel polar codes", TELKOMNIKA (Telecommunication Computing Electronics and Control), Vol. 21, No. 3, 2023.

[21] M. S. Abed, "Novel Hybrid Model Investing in 5G Network Optimization Under Suzuki Fading Channel", International Journal of Electronics and Telecommunications, Vol. 69, No. 3, 2023.

[22] P. D, A. Puthuraya, G. Srinivasan, H. Kumar, K. N. Reddy, "Polar Coding in 5G systems", International Research Journal on Advanced Science Hub, Vol. 5, No. 5, 2023, pp. 175-180.

[23] S. Höt, "Performance analysis of LDPC and Polar codes for message transmissions over different channel models", 1st Edition, Wiley, 2019.