Generalized SCAN Bit-Flipping Decoding Algorithm for Polar Code

Lou Chen[1], and Guo Rui[1*]
[1] School of Communication Engineering, Hangzhou Dianzi University
Hangzhou, 310018, China
[email:guorui@hdu.edu.cn]
*Corresponding Author: Guo Rui

Received January 14, 2023; revised March 6, 2023; accepted March 26, 2023; published April 30, 2023

Abstract

In this paper, based on the soft cancellation (SCAN) bit-flipping (SCAN-BF) algorithm, a generalized SCAN bit-flipping (GSCAN-BF-Ω) decoding algorithm is carried out, where Ω represents the number of bits flipped or corrected at the same time. GSCAN-BF-Ω algorithm corrects the prior information of the code bits and flips the prior information of the unreliable information bits simultaneously to improve the block error rate (BLER) performance. Then, a joint threshold scheme for the GSCAN-BF-2 decoding algorithm is proposed to reduce the average decoding complexity by considering both the bit channel quality and the reliability of the coded bits. Simulation results show that the GSCAN-BF-Ω decoding algorithm reduces the average decoding latency while getting performance gains compared to the common multiple SCAN bit-flipping decoding algorithm. And the GSCAN-BF-2 decoding algorithm with the joint threshold reduces the average decoding latency further by approximately 50% with only a slight performance loss compared to the GSCAN-BF-2 decoding algorithm.

Keywords: Polar codes, Error correction, Iterative decoding, Bit-flipping decoding, Multiple bit-flipping.

1. Introduction

Polar code is a class of linear block codes proposed by Arikan in [1], which depends on channel polarization. In theory, it has been proved that polar codes can reach the Shannon limit in binary memoryless channels when the code length tends to be infinite. Successive cancellation (SC) decoding algorithm is proposed along with the polar codes. However, error propagation will be caused by the successive decoding characteristic of SC algorithm and thus affect decoding performance. To obtain better decoding performance, the successive cancellation list (SCL) decoding algorithm was proposed in [2]. The proposed cyclic redundancy check (CRC)-aided SCL (CA-SCL) decoding algorithm in [3] achieves the most excellent decoding performance so far.

SCAN decoding algorithm has been attracting increasing attention because of the low decoding latency [4]. SCAN decoder belongs to soft output decoder like Belief Propagation (BP) decoder [5] and propagates information by updating factor graphs. Because of the schedule of SC decoding algorithm introduced in the information propagation of the factor graph, SCAN decoding algorithm has a much faster convergence speed than BP decoding algorithm and can reach the same performance as BP algorithm. Based on factor graphs with different permutations, a soft cancellation list (SCANL) decoding algorithm was proposed in [6] to provide a list of possible code words and select the most likely one as the output. [7] and [8] combine the idea of fast decoding to improve the speed of decoding. Based on the fact that parity-check constraints can enhance the reliability of other information bits over the iterations, a parity-check soft-cancellation (PC-SCAN) algorithm was proposed in [9]. To simplify the traversal of the decoding tree, an improved simplified SCAN (ISSCAN) decoding based on frozen bit check and two new special nodes was proposed in [10]. In [11], the idea of flipping was introduced into SCAN decoding for the first time to improve the error correction performance. The SCAN-BF decoding algorithm was proposed in [12] to locate errors using the segmentation submatrix, which can achieve a relatively significant performance gain for short to medium code lengths.

Although SCAN-BF decoder can get a significant performance gain for short to medium code lengths, there is still a significant performance gap compared to the SCL-like decoders. In this paper, we propose the GSCAN-BF-Ω decoding algorithm to enhance the BLER performance further. The proposed algorithm corrects the prior knowledge of the code bits and flips the prior information of the information bits at the same time to improve decoding performance when the maximum number of iterations is reached or the CRC detection is not satisfied. And for the GSCAN-BF-2 decoding algorithm, a joint threshold scheme to reduce the decoding complexity and latency is proposed.

The rest of the paper is organized as follows. Polar codes and SCAN-BF decoding algorithm are described in Section II. Section III introduces the proposed GSCAN-BF-Ω decoding algorithm and the joint threshold scheme. Simulation results and complexity analysis are presented in Section IV. Section V concludes this paper.

2. Preliminaries

2.1 Foundation of Polar Code

A polar code with information bit size K and length $N = 2^n$ can be denoted as $P(N, K)$. The encoding processing of polar codes depends on the generation matrix, which can be represented as $G_N = F^{\otimes n}$ where $F^{\otimes n}$ is the n-th Kronecker power for $F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$.The source bits vector is noted as u_1^N , and the encoded codeword can be given by $x_1^N = u_1^N \cdot G_N$.

2.2 SCAN Decoding Algorithm

The SCAN decoding algorithm achieves iterative decoding by propagating $L_{i,j}$ messages to left and $R_{i,j}$ messages to right on factor graph [4], where i represents the stage of decoding processing, j represents the row index of decoding processing. And SCAN decoding algorithm follows the same initialization and message propagation rules as the BP decoding algorithm.

The information L is initialized to the received log-likelihood ratios ($LLRs$), and the information R is initialized to $+\infty$ or 0 according to the fact that the bit is frozen bit or information bit. However, SCAN algorithm introduces the schedule of SC decoding algorithm, therefore SCAN algorithm converges much faster than BP algorithm.

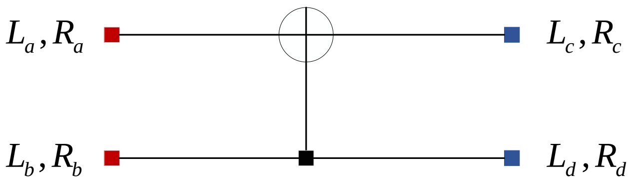


Fig. 1. A processing element of the SCAN decoding algorithm

The basic processing element of SCAN decoding algorithm is shown in **Fig. 1**. And it can represent the basic polar kernel [6]. The L and R information are transferred on the factor graph according to **Fig. 1** and (1).

$$\begin{cases} L_a = f(L_c, L_d + R_b) \\ L_b = L_d + f(L_c, R_a) \\ R_c = f(R_a, L_d + R_b) \\ R_d = R_b + f(L_c, R_a) \end{cases} \tag{1}$$

Where $f(a,b) = \min(|a|,|b|)sign(a)sign(b)$; $\log((1+e^{a+b})/(e^a + e^b))$. An iteration of SCAN decoder is completed when $R_{n+1,j}$ has been updated. And the message propagation is not repeated when maximum iteration is met.

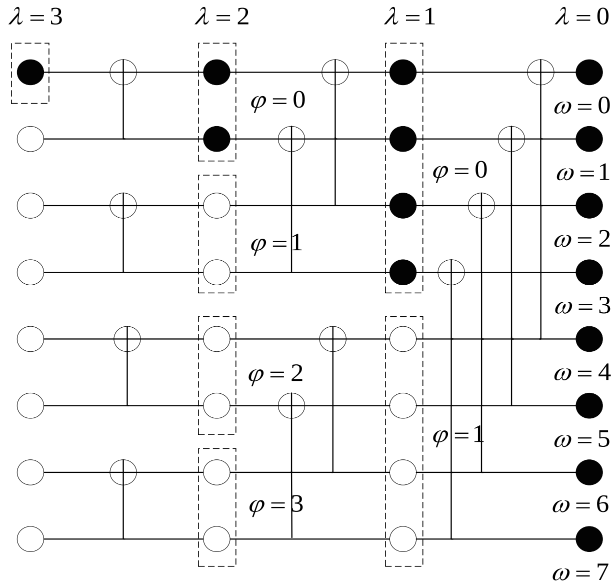


Fig. 2. Factor graph of SCAN decoding algorithm for $N = 8$

Fig. 2 shows the factor graph for $N = 8$ polar codes of SCAN decoding algorithm, where ω is the index of node, φ is the index of the block and λ is the index of column. Each column contains 2^{λ} blocks, and each block contains $2^{n-\lambda}$ nodes, so we can pinpoint each node on the factor graph with $(\lambda, \varphi, \omega)$.

2.3 SCAN Bit-Flipping Decoding Algorithm

The bit-flipping scheme is an effective solution to improve the BLER performance of the SCAN decoding algorithm [11]. SCAN-BF decoder can get a large performance gain by flipping the prior information of unreliable information bits from flipping set (FS) when the SCAN decoder has failed.

The FS is composed of unreliable information bits and the flipping strategy selects bits from FS for the flipping operation. Initially, the construction of FS was carried out in such a way that the information bit with the lowest LLR value was selected. And the flipping operation is as follows:

$$R_{1,j} \begin{cases} +\infty, & \mathrm{j} \in FS \; and \; \mathrm{u}_j = 1 \\ -\infty, & \mathrm{j} \in FS \; and \; \mathrm{u}_j = 0 \end{cases} \tag{2}$$

The details of SCAN-BF decoding algorithm are given in Algorithm 1.

Algorithm 1: The SCAN Bit-Flipping Decoding Algorithm

Input: y_1^N **,** $FS = \{j_1, j_2, \ldots, j_\tau\}$

Output: $\hat{u}_1^N$

1: Initialize the $\boldsymbol{L}$, $\boldsymbol{R}$
2: $\boldsymbol{L}' \leftarrow \boldsymbol{L}$, $\boldsymbol{R}' \leftarrow \boldsymbol{R}$
3: **for** $k = 1:\tau$ **do**
4: $\boldsymbol{L} \leftarrow \boldsymbol{L}'$, $\boldsymbol{R} \leftarrow \boldsymbol{R}'$
5: $R_{1,j_k} \leftarrow (1 - u_{j_k}) \times \infty$
6: $\hat{u}_1^N \leftarrow$ conventional SCAN decoding
7: **if** CRC($\hat{u}_1^N$)=true **then**
8: **return** $\hat{u}_1^N$
9: **endif**
10: **endfor**
11: **return** $\hat{u}_1^N$

When the original SCAN decoder fails to produce a result that passes the CRC detection, the bit-flipping strategy is started. According to Algorithm 1, The information L and the information R are reinitialized, then the SCAN-BF decoder sequentially flips the prior information of the information bits within the FS and executes the normal SCAN decoder until it produces a result that passes CRC detection. Where u_{j_k} represents the decoding result of the original SCAN decoder at position j_k. And the concept of flipping can be reflected in the fifth line of Algorithm 1 and in (2) while the concept of flipping means flipping the bit in the other direction [13].

3. The GSCAN-BF Decoding Algorithm

In this section, based on the SCAN-BF decoding algorithm, a generalized SCAN-BF-Ω decoding algorithm is carried out. By correcting the prior knowledge of code bits and flipping the information bits simultaneously, the GSCAN-BF-Ω algorithm further improves the BLER performance. Then, to reduce the additional computational complexity of the exhaustive attempts, a joint threshold scheme is proposed for GSCAN-BF-2 decoding algorithm.

3.1 GSCAN-BF Decoding Algorithm

The existing bit-flipping decoding algorithms for SCAN only take into account the prior information of the information bits, and do not consider the prior information of the code bits. However, exhaustive simulations have shown that correcting the prior information of the code bits can significantly improve the BLER performance, as compared to flipping the prior information of the information bits.

Therefore, we extend SCAN-BF decoding algorithm to a generalized bit-flipping version which called GSCAN-BF-Ω decoding algorithm. Different from the existing bit-flipping algorithms [14], GSCAN-BF-Ω decoding algorithm corrects the prior information of the code bits and flips the prior information of the unreliable information bits simultaneously. It is important to note that we distinguish between flipping and correcting operations. The flipping operation has already been described in the previous section and the correcting operation assign $+\infty$ or $-\infty$ to $L_{n+1,j}$ to perform additional decoding attempts.

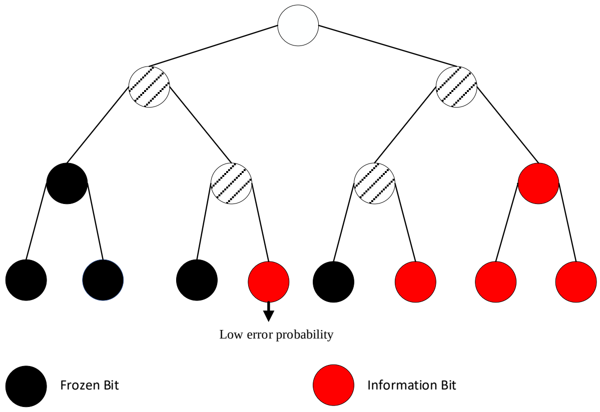


Fig. 3. A full code tree of polar code for N=8

There are two critical sets in GSCAN-BF-Ω algorithm, one is correction set (CS) and the other is FS. The selection of critical sets is not nested like higher-order algorithms. This means that a completely different approach can be taken to the selection of critical sets which ensures differences between CS and FS. And the performance of critical set is affected by the method used to construct the set.

Regarding the method used to construct the FS of unreliable information bits, the method used for SCAN-BF is adopted. The nodes can be classified as rate-1 and rate-0 nodes according to the decoding binary tree. A rate-1 node with the length of 2^s contains 2^s information bis, and a rate-0 node with the length of 2^s contains 2^s frozen bis. If the rate-1 node is immediately followed by a rate-0 node, this rate-1 node has a low error probability for the last bit. At the same time, a long rate-1 node can be splitted into several rate-1 nodes with the length 4. For these rate-1 nodes, the last information bit can be considered with low error probability, and all the other information bits are considered to be of high error probability. And a full code tree can be shown as **Fig. 3**.

We sort the bits with a high error probability in ascending order according to LLR, and select the smallest τ information bits to construct $FS = \{j_1, j_2, ..., j_\tau\}$.

CS represents the correction set, and bits with low prior knowledge need to be corrected in GSCAN-BF-Ω decoding process. In this paper, CS is constructed by selecting, in the most intuitive way, the index subscript with the smallest LLR value. We sort the received LLR in ascending order and select the subscripts of the least reliable ω values to construct $CS = \{j_1, j_2, ..., j_\omega\}$. And the sorting result of the received LLR can be represented as :

$$|L_{n+1,j_1}| \le |L_{n+1,j_2}| \le \dots \le |L_{n+1,j_N}|$$

The details of GSCAN-BF-Ω decoding algorithm are given in Algorithm 2.

Algorithm 2: The Proposed GSCAN-BF-Ω decoding algorithm

Input: y_1^N **,** $CS=\{j_1, j_2, \ldots, j_\omega\}$, $FS=\{j_1, j_2, \ldots, j_\tau\}$

Output: $\hat{u}_1^N$

Initilalize the $\boldsymbol{L}, \boldsymbol{R}$
$\hat{u}_1^N \leftarrow$ conventional SCAN decoding
while CRC($\hat{u}_1^N$)!=true
  $\{j_1, j_2, \ldots, j_{\Omega_1}\} \leftarrow$ Get Ω_1 elements in an exhaustive way from FS
  $\{j_1, j_2, \ldots, j_{\Omega_2}\} \leftarrow$ Get Ω_2 elements in an exhaustive way from CS
  $\boldsymbol{L}' \leftarrow \boldsymbol{L}, \boldsymbol{R}' \leftarrow \boldsymbol{R}$
  for $c = 0 : 2^{\Omega_1} - 1$
    for $\mathrm{b} = 0 : 2^{\Omega_2} - 1$
      $d_1^{\Omega_1} \leftarrow dec2bin(c, \Omega_1)$
      $a_1^{\Omega_2} \leftarrow dec2bin(b, \Omega_2)$
      for $k = 1 : \Omega_1$ **do**
        $R_{1, j_k} \leftarrow (1 - 2d_k) \times \infty$
      end for
      for $m = 1 : \Omega_2$ **do**
        $L_{n+1, j_m} \leftarrow (1 - 2a_m) \times \infty$
      end for
      $\hat{u}_1^N \leftarrow$ conventional SCAN decoding
      if CRC($\hat{u}_1^N$)=true **then**
        return $\hat{u}_1^N$
      end if
    end for
  end for $// \Omega_1 + \Omega_2 = \Omega$
end while
return $\hat{u}_1^N$

The bit-flipping choice of the GSCAN-BF-Ω decoding algorithm is flexible. It means that when $\Omega > 2$, correction and bit-lipping operations can be flexibly assigned. For example, when $\Omega = 3$, we can choose two-bit flipping operation and one-bit correcting operation or two-bit correcting operation and one-bit flipping operation. However, considering the superiority of correcting the prior information of code bits, we tend to prefer to perform correcting operation.

And it is because of the flexibility and the dramatic increase in complexity that we only discuss the GSCAN-BF-2 decoding algorithm with $\Omega_1 = 1$, $\Omega_2 = 1$ and GSCAN-BF-1 decoding algorithm with $\Omega_2 = 1$ in the following. Note that when $\Omega_1 = 1$ the GSCAN-BF-1 decoding algorithm can be seen as SCAN-BF algorithm but with exhaustive enumeration.

And different from the SCAN-BF algorithm, GSCAN-BF-1 algorithm focuses on the priori knowledge of code bits.

3.2 Joint Threshold Scheme for GSCAN-BF-2 Decoding Algorithm

Although the GSCAN-BF-2 decoding algorithm exhibits good decoding performance, it is the norm that the correct decoding result cannot be obtained when traversing the combinations between critical sets. This leads to additional decoding attempts, which in turn increases the complexity of decoding. Taking into account the high additional complexity of the exhaustive bilateral attempts of the GSCAN-BF-2 decoding algorithm, we propose a joint threshold scheme to reduce decoding attempts. This stems from the fact that if the reliability of a bit channel is high enough, we can assume that the estimate of that bit channel is also reliable enough. However, it is not wise to directly remove bits from the critical set where the reliability of the bit channel is high enough without taking into account the effect of the absolute LLR value [13].

Therefore, this scheme first proposes a metric $M(m,k)$ for the GSCAN-BF-2 decoding algorithm that takes into account both the bit channel reliability and the code bit reliability. Combining these two factors, the metric can more accurately identify the reliability of the current combination. And this scheme reduces the average decoding complexity by eliminating the combinations of critical sets that have high reliability.

Inspired by [15], the reliability of the m-th bit in CS can be expressed as $L_{n+1,j_m}+R_{n+1,j_m}$, and the channel reliability of the k-th bit in FS can be expressed as $ind(j_k)/|FS|$,where ind is the reliability sequence of bit channels. So we can obtain the following equation for the combined metric between critical sets where $LLR_{\max}$ is the maximum value of the accepted LLR and can be seen as a penalty for bit channel reliability. And $M(m,k)$ can be expressed as:

$$M(m,k)=(|L_{n+1,j_m}+R_{n+1,j_m}|)+\ln(\frac{ind(j_k)}{|FS|}-LLR_{\max}) \tag{3}$$

Then, based on the metric, we can obtain the joint threshold γ. And γ can be seen as the average reliability of CS and FS. In the same way we can derive the equation for the joint threshold as follow:

$$\gamma=\ln(\frac{\sum_{k=1}^{|FS|}ind(k)}{|FS|}-LLR_{\max})+(\frac{\sum_{m=1}^{|CS|}(|L_{n+1,j_m}+R_{n+1,j_m}|)}{|CS|}) \tag{4}$$

The reliability of such critical sets combinations is determined by calculating the values of $M(m,k)$ and γ. And scheme based on the joint threshold can be briefly described as follows: if $M(m,k)>\gamma$, then this combination can be considered reliable enough and the decoding attempt based on this combination can be left unexecuted. And if $M(m,k)\le\gamma$, the GSCAN-BF-2 decoder performs as usual. In this way we can reduce the complexity of decoding without losing performance as much as possible.

4. Simulation Results And Analysis

In this section, we give the simulation results of the proposed algorithm under the condition that the channel is additive white Gaussian noise (AWGN) and the modulation method is binary phase shift keying (BPSK).

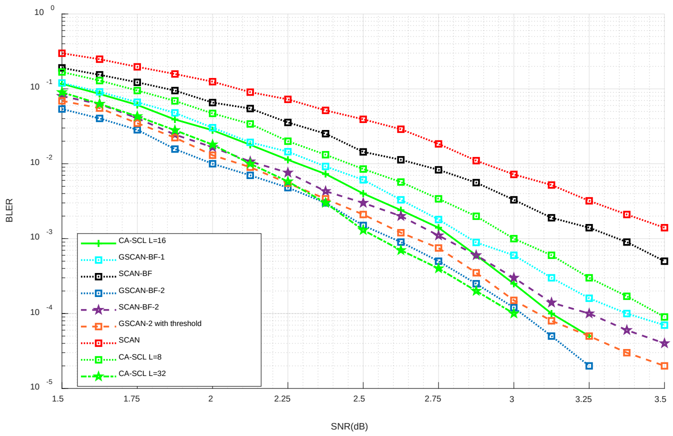
4.1 Analysis of Decoding Performance

Fig. 4 shows the BLER performance comparison of CA-SCL (L=8,16,32), SCAN, SCAN-BF, SCAN-BF-2, GSCAN-BF-1, and GSCAN-BF-2 decoding algorithms. Here, code length N=256 and the CRC length is 16, and all iteration times are set to 4. **Fig. 4.1** and **Fig. 4.2** show the results at different rates respectively. **Fig. 4.1** shows the result for rate R=0.5 and **Fig. 4.2** shows the result for rate R=0.6.

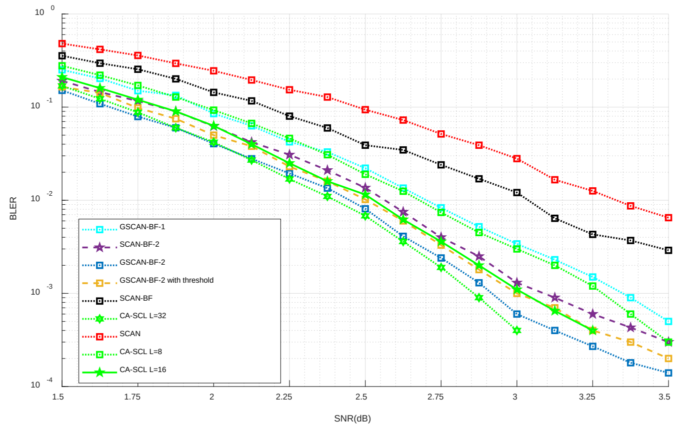
As shown in **Fig. 4.1**, the GSCAN-BF-1 decoding algorithm exhibits performance that exceeds CA-SCL (L=8), at a BLER of $=10^{-3}$, it improves the decoding performance by 0.1dB compared with CA-SCL (L=8) decoding algorithm. And the GSCAN-BF-1 decoding algorithm can reach almost the same BLER performance as CA-SCL (L=16) at low to medium SNR region. The correction set size of GSCAN-BF-1 is set to 20, which matches the flipping set size of the SCAN-BF algorithm for fairness. And the GSCAN-BF-1 decoder obviously has much better BLER performance than the SCAN-BF decoder.

The GSCAN-BF-2 decoding algorithm achieves slightly better BLER performance than CA-SCL(L=32) when SNR is below 2.5 dB, and performs similarly to CA-SCL(L=32) when SNR is higher than 2.5dB. Compared with the GSCAN-BF-1 algorithm, GSCAN-BF-2 decoding algorithm yields a 0.4dB gain at a BLER of $=10^{-2}$. And it is obvious that the GSCAN-BF-2 introduces a 0.25dB BLER performance gain over the SCAN-BF-2 decoding algorithm at a BLER of $=10^{-4}$. As for the proposed GSCAN-BF-2 decoding algorithm with the joint threshold scheme, it exhibits only a slight performance loss when compared to GSCAN-BF-2 decoding algorithm. However, it still shows better BLER performance than SCAN-BF-2 decoding algorithm.

Fig. 4.2 shows the case when R = 0.6. The GSCAN-BF-1 decoding algorithm shows almost the same decoding performance as CA-SCL (L=8). GSCAN-BF-2 decoding algorithm can exhibit BLER performance close to CA-SCL(L=32), and get nearly 0.2dB performance gain at a BLER of $=10^{-3}$ compared to SCAN-BF-2. The GSCAN-BF-2 decoding algorithm with the joint threshold scheme shows the same trend as when R = 0.6. It still shows better BLER performance than SCAN-BF-2.



(1)



(2)

Fig. 4. BLER Comparison of different decoding algorithms for R=0.5 and R=0.6 polar code.

The GSCAN-BF-2 decoding algorithm follows a specific order that flipping the information bits preferentially. In **Fig. 5** we provide the result of exchanging the nesting order of the loop and the effect of different sizes of ω and τ on BLER performance in GSCAN-BF-2 decoding algorithm. The first set in the legend means to be flipped or corrected first.

Fig. 5 shows the fact that the BLER performance of the GSCAN-BF-2 decoding algorithm improves as the ω and τ increases. And the GSCAN-BF-2 decoder with $\omega = 20$ and $\tau = 10$ introduces a 0.25dB performance gain over with $\omega = 10$ and $\tau = 20$ at a BLER of $= 10^{-4}$ which means that the size of ω plays a more critical role in the impact of GSCAN-BF-2 BLER performance. **Fig. 5** also shows that the exchange of the nesting order has almost no effect on the BLER performance of GSCAN-BF-2 decoder.

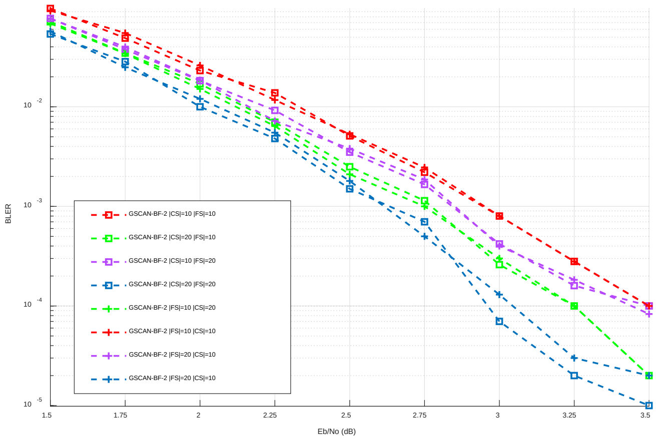


Fig. 5. BLER Comparison of different ω and τ for GSCAN-BF-2 decoding algorithm

4.2 Decoding complexity Analysis

According to [1], the decoding complexity of SC decoding algorithm is $O(N\log_2 N)$, while the decoding complexity of the SCL decoder is depended on the size of L and can be denoted as $O(L \cdot N\log_2 N)$. The decoding complexity of SCAN decoder with single-iteration is consistent with the SC decoder, while the decoding complexity of SCAN decoder with multi-iteration is related to the number of iterations M_{iter}, which can be expressed as $O(M_{iter} \cdot N\log_2 N)$.

However, the number of correction times T also needs to be taken into account. Therefore, the decoding complexity of the GSCAN-BF-1 decoding algorithm can be expressed as $O((1+T)\cdot M_{iter} \cdot N\log_2 N)$. It is obvious that the number of correction times is not a constant, so we should take into account the average decoding complexity. Similar to SCAN-BF algorithm [12], the average computational complexity of GSCAN-BF-1 can be expressed as:

$$\overline{f}(n) = O((1+T\cdot P_B(SNR,r))\cdot M_{iter}\cdot N\log_2 N) \tag{5}$$

$P_B(SNR,r)$ can be defined as the BLER of polar codes, and $T\cdot P_B(SNR,r)$ can denote the average correction times of GSCAN-BF-1 decoder. And as SNR continues to rise, the average correction times decreases continuously. When the SNR tends to infinity, the average correction times tends to be 0, the computational complexity of GSCAN-BF-1 decoder converges to $O(M_{iter}\cdot N\log_2 N)$.

As for the decoding complexity of the GSCAN-BF-2 algorithm, it not only needs to consider the complexity of correcting the prior knowledge of code bits, but also needs to consider the additional complexity caused by information bits flipping. Here we denote the correction times of the code bits as T_c. The flipping times of the code bits can be expressed as T_f, then the decoding complexity of GSCAN-BF-2 algorithm can be expressed as $O((1+T_c\cdot T_f)\cdot M_{iter}\cdot N\log_2 N)$, and the average computational complexity of GSCAN-BF-2 algorithm can be expressed as (6).

$$\overline{f}(n) = O((1+T_c \cdot T_f \cdot P_B(SNR, r)) \cdot M_{iter} \cdot N \log_2 N) \quad (6)$$

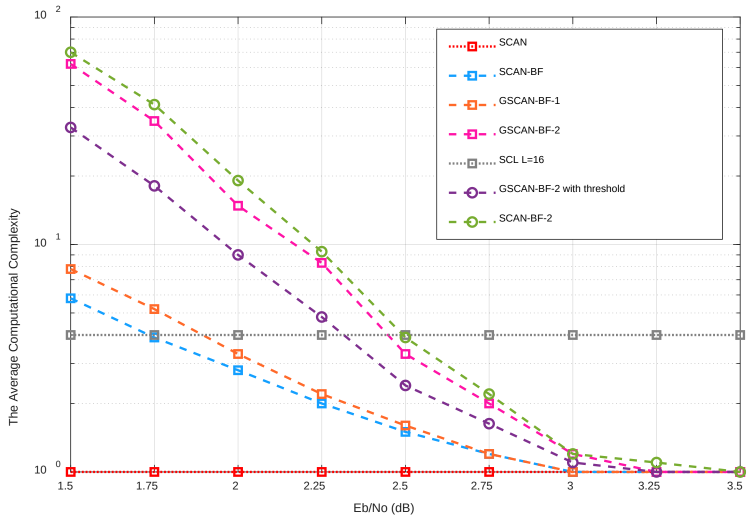


Fig. 6. Average computational complexity comparison of different algorithms.

We can get the average decoding complexity of GSCAN-BF-1 and GSCAN-BF-2 algorithms by simulation. As be seen in **Fig. 6**, SCAN decoding algorithm with M_{iter} =4 is used to be the benchmark for comparison. As shown in **Fig. 6** the average decoding complexity of the GSCAN-BF-1 decoder is slightly higher than SCAN-BF decoder at medium and low SNR areas, and almost the same at high SNR region. It is evident that the computational complexity of SCAN-BF-2 decoding algorithm is the highest. This means that the GSCAN-BF-2 algorithm has better performance than the SCAN-BF-2 algorithm while having lower average complexity. When $SNR < 2.5$, the average complexity of GSCAN-BF-2 is higher than SCL (L=16) algorithm. As SNR increases, the average decoding complexity of GSCAN-BF-2 is gradually decreasing and close to SCAN and GSCAN-BF-1 decoding algorithms. This is predictable, as the channel noise decreases, the GSCAN-BF-2 decoder needs to perform fewer decoding attempts. And the proposed GSCAN-BF-2 decoding algorithm based on the joint threshold scheme can reduce decoding complexity by almost 50% compared to the GSCAN-BF-2 decoding algorithm.

5. Conclusion

In this paper, to improve the BLER performance, a GSCAN-BF-Ω decoding algorithm that corrects the prior information of code bits and flipping the prior information of information bits simultaneously is carried out. Then to reduce the average decoding complexity, we propose a joint threshold scheme that takes into account the bit channel quality and the reliability of the code bit for the GSCAN-BF-2 decoding algorithm. Simulation shows the results that the proposed GSCAN-BF-1 decoding algorithm and GSCAN-BF-2 decoding algorithm can close the BLER performance for CA-SCL decoder with list size 8 and 32, separately. And the proposed GSCAN-BF-2 decoding algorithm with the joint threshold scheme can reduce computational complexity by almost 50% with only a slight BLER performance loss compared to the GSCAN-BF-2 decoding algorithm.

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, pp. 3051-3073, July 2009. Article (CrossRef Link)

[2] I. Tal and A. Vardy, "List Decoding of Polar Codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015. Article (CrossRef Link)

[3] K. Niu and K. Chen, "CRC-Aided Decoding of Polar Codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668-1671, October 2012. Article (CrossRef Link)

[4] U. U. Fayyaz and J. R. Barry, "Low-Complexity Soft-Output Decoding of Polar Codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 958-966, May 2014. Article (CrossRef Link)

[5] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447-449, June 2008. Article (CrossRef Link)

[6] C. Pillet, C. Condo and V. Bioglio, "SCAN List Decoding of Polar Codes," in *Proc. of ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, pp. 1-6, 2020. Article (CrossRef Link)

[7] L. Zhang, Y. Sun, Y. Shen, W. Song, X. You and C. Zhang, "Efficient Fast-SCAN Flip Decoder for Polar Codes," in *Proc. of 2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, Daegu, Korea, pp. 1-5, 2021. Article (CrossRef Link)

[8] C. Pillet, C. Condo and V. Bioglio, "Fast-SCAN decoding of Polar Codes," in *Proc. of 2021 11th International Symposium on Topics in Coding (ISTC)*, Montreal, QC, Canada, pp. 1-5, 2021. Article (CrossRef Link)

[9] J. Tong, H. Zhang, X. Wang, S. Dai, R. Li and J. Wang, "A Soft Cancellation Decoder for Parity-Check Polar Codes," in *Proc. of 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, London, UK, pp. 1-6, 2020. Article (CrossRef Link)

[10] M. Cai, S. Li and Z. Liu, "An Improved Simplified Soft Cancellation Decoding Algorithm for Polar Codes Based on Frozen Bit Check," in *Proc. of 2021 IEEE 21st International Conference on Communication Technology (ICCT)*, Tianjin, China, pp. 127-131, 2021. Article (CrossRef Link)

[11] Y. Wang, L. Chen, S. Qiu, L. Huang, and Z. Xing, “A Low Latency SCAN-Flip Polar Decoder for 5G Vehicular Communication,” in *Proc. of INTSYS 2018: Intelligent Transport Systems, From Research and Development to the Market Uptake*, pp. 157–170, 2019. Article (CrossRef Link)

[12] X. Wang, H. Liu, J. Li, Z. Zheng and J. He, "SCAN-BF Decoding of Polar Codes," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2507-2511, Aug. 2021. Article (CrossRef Link)

[13] Y. Shen, W. Song, Y. Ren, H. Ji, X. You and C. Zhang, "Enhanced Belief Propagation Decoder for 5G Polar Codes With Bit-Flipping," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 901-905, May 2020. Article (CrossRef Link)

[14] Y. Yu, Z. Pan, N. Liu and X. You, "Belief Propagation Bit-Flip Decoder for Polar Codes," *IEEE Access*, vol. 7, pp. 10937-10946, 2019. Article (CrossRef Link)

[15] M. Zhang, Z. Li, L. Xing, and X. Liao, “Higher-Order Belief Propagation Correction Decoder for Polar Codes,” *Entropy*, vol. 24, no. 4, p. 534, Apr. 2022. Article (CrossRef Link)

Lou Chen is a Master student in Hangzhou Dianzi University. He received the B.E. degree in Communication Engineering from Kunming University of Science and Technology, in 2021. His research interests include channel coding and wireless communication.

Guo Rui received the Ph.D. degree from the Zhejiang University, Hangzhou, China, in 2007. He is currently an associate professor with the School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China. He is a visiting scholar at Oregon State University from August 2018 to August 2019. His research interests include wireless communication and channel coding.