

Received November 11, 2019, accepted December 16, 2019, date of publication December 24, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962003

BER Evaluation Based SCFlip Algorithm for Polar Codes Decoding

XUETING ZHANG¹, YINGZHUANG LIU¹, AND SHAOPING CHEN²

¹School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

²Hubei Key Laboratory of Intelligent Wireless Communications, South Central University for Nationalities, Wuhan 430074, China

Corresponding author: Xueting Zhang (deppy124@hust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61971452 and Grant 61231007.

ABSTRACT Successive cancellation (SC) decoding of polar codes may bring about error propagation that needs to be mitigated. In this paper, we present a new SC Flipping (SCFlip) decoder, named bit error rate (BER) evaluation based SCFlip (BER-SCFlip), which can accurately target the first error bit and correct it with a high probability. Thus, a high error correction capability and a low decoding complexity can be achieved. First, we propose a new criterion to find out the most suspicious error bit. Those non-frozen bits that have higher decoding BERs derived from log-likelihood-ratios (LLRs) after SC decoding than the corresponding expected ones estimated via Gaussian Approximation (GA), are collected into the flip-bits set. These candidate bits will be flipped one by one according to their SC decoding orderings in extra decoding attempts until the decoded codeword passes cyclic redundancy check (CRC) or a predetermined maximum number of extra attempts is reached. We then propose an extended version of BER-SCFlip, named BER-SCFlip- ω with the capability to correct up to ω error bits in each extra decoding attempt. By combining our criterion for the flip-bits selection with that of Dynamic SCFlip (D-SCFlip), the proposed BER-SCFlip- ω significantly reduces decoding complexity and latency while maintaining the superior error-correction performance close to that of D-SCFlip- ω . The simulation results show that the proposed schemes are competitive among existing SCFlip algorithms and could achieve the error-correction performance approaching that of CRC-aided SCL decoding under list size $L = 16$ while maintaining low complexity.

INDEX TERMS Polar codes, SC decoding, SCFlip decoding, Gaussian approximation.

I. INTRODUCTION

Polar codes are the first family of channel codes that are proved theoretically to achieve the capacity of binary-input discrete memory-less channels (B-DMCs) under a low-complexity successive cancellation (SC) decoding when code lengths go to infinity [1]. However, for polar codes of short and moderate lengths, the error-correction performance under SC decoding is not satisfactory. Two factors may contribute to the weakness of finite-length polar codes: partial polarization and sub-optimality of SC decoding. Accordingly, two categories of approaches were proposed in the literature. One employed modified kernels in the code construction to speed up the polarization rate [2]–[5]. The other focused on enhancing the performance of SC decoding, e.g., the SC list (SCL) decoder proposed in [6], [7] to significantly improve the error-correction performance of polar codes at finite lengths. To further enhance the performance of

SCL decoder, the authors in [6], [8] proposed a CRC-polar coding scheme, where outer CRC codes were concatenated with inner polar codes to help identify the correct path from the list of decoding paths. The CRC-polar codes under SCL decoding not only compete with LDPC and Turbo codes in terms of the error-correction performance, but also have a lower encoding and decoding complexity thanks to their recursive operations [8]. Thus, CRC-polar codes have been chosen by 3GPP as error correction codes in the control channel for the 5th generation of mobile communication standards (5G) [9]. Since the invention of CRC-polar codes and their SCL decoding, many researches have been conducted to further improve the error correction performance [10], [11], or to reduce the decoding complexity, latency and storage cost [12]–[16].

In addition to the SCL decoder, an SC Flipping (SCFlip) decoder was proposed as an alternative to improve the performance of SC decoding [17]. By trying to identify the first error bit based on the magnitudes of log-likelihood-ratio (LLR) values in SC decoding and flip it in extra

The associate editor coordinating the review of this manuscript and approving it for publication was Mingjun Dai.

decoding attempts, SCFlip decoder could achieve a better error correction performance than SC decoder at a cost of a minor increase in the decoding complexity. However, it could not compete with SCL decoding under large list sizes in the low SNR regime, for its inability to accurately target the error bits solely based on the LLRs.

To boost the performance of conventional SCFlip decoders, many approaches have been reported in the literature, either to reduce the computational complexity or to enhance the error-correction performance. To reduce the complexity, a fast SCFlip decoder was proposed in [18] with a narrowed critical set for flipping (flip-bits set) such that the efficiency of flipping and re-decoding procedures could be improved and its hardware implementation was simplified. The authors in [19] further reduced the search zone in constructing the critical set by exploiting the code tree structure. Moreover, by adjusting the critical set progressively during the decoding, the proposed Progressive SCFlip could rival SCL decoding with a lower average complexity. It was observed from simulations in [20] that the first error was distributed non-uniformly, and an adaptive threshold was employed to screen possible flipping candidates. Thus, a good trade-off between performance and complexity was achieved. The same authors proposed a partitioned SCFlip decoder to detect and correct more errors in SC decoding [21]. By dividing a codeword into many subcodes and protecting them with outer CRC codes, the partitioned SCFlip decoder could achieve a better performance with a low average complexity. The authors in [22] combined the bit flipping with the high-performance SCL decoding such that a performance gain was achieved at a cost of increased complexity. Another SCL-based flipping (SCLFlip) decoding algorithm was also proposed in [23]. With a superior bit-flip criterion, the SCLFlip decoder in [23] could not only outperform existing SCLFlip decoders but also achieve a better performance than SCL decoder while maintaining a similar complexity. In [24] and [25], Ludovic et al. presented a Dynamic SCFlip (D-SCFlip) decoder where a novel metric was designed to select the possible erroneous non-frozen bits (flip-bits) by taking into account the sequential nature of SC decoder. By dynamically updating the flip-bits set and flipping one or multiple error bits per decoding attempt, D-SCFlip could achieve a good error-correcting performance close to that of CRC-aided SCL decoder with the list size $L = 16$.

The key to successful SCFlip decoding is to accurately pinpoint error bits of SC decoding for the extra decoding attempts. The existing criteria to choose the flip-bits are mainly based on the reliability comparison among different non-frozen bits, e.g., using LLRs or other metrics. Considering that the reliability of polar codes may vary from bit to bit due to polarization, it is more reasonable to take into account the inherited reliability difference among non-frozen bits in the selection of flipping bits.

Based on this analysis, we propose a new criterion for finding out possible erroneous bits in SC decoding. For the AWGN channel, we first calculate the bit error rate (BER)

expectations for all non-frozen bits via Gaussian Approximation (GA) algorithm and then compute the estimated BERs from LLRs after SC decoding. Those non-frozen bits, whose estimated BERs after SC decoding are higher than their corresponding BER expectations, are chosen as suspicious bits and their subchannel indexes are collected as the flip-bits set. The earlier an error occurs, the more serious harm it will impose on the following decoding. Thus, we sort the indexes according to their orderings in the standard SC decoding. Based on this criterion, we propose a BER evaluation based SCFlip (BER-SCFlip) decoder that can achieve a good error-correction performance. Compared to the original SCFlip using absolute LLR values for the flip-bits selection, the scope of the flip-bits can be narrowed in our proposed BER-SCFlip such that the decoding complexity is reduced. To enhance the performance of BER-SCFlip, an extended version, named BER-SCFlip- ω , is proposed with a capability to correct up to ω errors for each extra decoding attempt. In BER-SCFlip- ω , the proposed BER evaluation criterion is combined with that of D-SCFlip in [25]. This combined strategy can significantly narrow the scope of candidate bits such that the complexity can be efficiently reduced.

To evaluate the performance of our proposed schemes, we derive lower and upper bounds of the word error rate (WER) performance. The lower bound of the WER performance is calculated using Oracle-assisted SC (OASC) decoder proposed in [17], which assumed the error bits are known by the receiver and thus enabled the accurate bit flipping. Under the CRC error-detection, the upper bound of the WER performance is also derived. Additionally, the influence of the CRC length and the maximum number of new decoding attempts on the WER performance is analyzed to optimize the design of our proposed scheme.

The rest of the paper is organized as follows. Section II introduces the notation and the basic principle of polar codes. Their SC and SCFlip decodings are also described in this section. The proposed BER-SCFlip decoder is presented in Section III. In Section IV, BER-SCFlip- ω decoder is proposed and its upper and lower bounds of the WER performance are analyzed in Section V. Simulation results are provided in Section VI and finally, conclusions are drawn in Section VII.

II. PRELIMINARY

A. POLAR CODES

Polar codes are derived from channel polarization, which is realized by performing channel combining and splitting operations on $N = 2^n$ independent and identical B-DMC channels [1]. Those k polarized subchannels $\mathcal{A} \subset \{1, 2, \dots, N\}$ with higher reliability are assigned to transmit information bits. The remaining subchannels \mathcal{A}_c are frozen to transmit a fixed value, e.g., 0, known to both the encoder and the decoder. The reliability of subchannels can be estimated by several techniques, such as Bhattacharya Parameter [1], density evolution [26], GA of density evolution [27], [28], efficient degrading and upgrading methods [29].

The encoded codeword, denoted by \mathbf{X} , is obtained by $\mathbf{X} = \mathbf{U}G_N$, where G_N is a generator matrix and \mathbf{U} is a data vector. The data vector \mathbf{U} consists of k information bits on subchannels \mathcal{A} and $N - k$ frozen bits on subchannels \mathcal{A}_c . The codeword \mathbf{X} is then modulated and sent through the channel. The data vector \mathbf{Y} is received at the decoder as an input to estimate the transmitted data $\hat{\mathbf{u}}$.

B. SC AND SCFLIP DECODING

In SC decoding, a pair of probabilities $P(u_i = 0|\hat{u}_1^{i-1}, \mathbf{Y})$ and $P(u_i = 1|\hat{u}_1^{i-1}, \mathbf{Y})$ is recursively calculated [1], where \hat{u}_1^{i-1} denotes the sequence $(\hat{u}_1, \hat{u}_2, \hat{u}_3, \dots, \hat{u}_{i-1})$. For each bit u_i , its estimate \hat{u}_i depends on both \mathbf{Y} and the previous estimates \hat{u}_1^{i-1} and can be obtained by hard decisions as

$$\hat{u}_i = h(\tilde{L}_N^i) = \begin{cases} u_i, & \text{if } i \in \mathcal{A}_c \\ (1 - \text{sign}(\tilde{L}_N^i))/2, & \text{if } i \in \mathcal{A}, \end{cases} \quad (1)$$

where \tilde{L}_N^i is the LLR of decoded symbol i computed as

$$\tilde{L}_N^i = \log\left(\frac{P(u_i = 0|\hat{u}_1^{i-1}, \mathbf{Y})}{P(u_i = 1|\hat{u}_1^{i-1}, \mathbf{Y})}\right).$$

To deal with the error propagation problem, SCFlip decoder was proposed in [17] to find out the suspicious channel-incurred error bits and flip them in extra decoding attempts, if the standard SC decoding failed. Each new decoding attempt performed the similar SC decoding except flipping one of the candidate bits, which had the lowest LLR absolute values. The computational complexity of SCFlip increases slightly compared with SC decoder in the medium and high SNR regimes and its error-correction performance could compete with CRC-aided SCL with $L = 2$. Following the line of SCFlip decoder proposed in [17], many researches have been reported in the literature for improving the error-correction performance or reducing the computational complexity [18]–[25].

III. BER-SCFLIP DECODER

It is essential for SCFlip decoder to find out the first error bit of SC decoder and correct it in the following decoding attempt. In this section, we present our proposed SCFlip decoder, BER-SCFlip decoder, with our designed method for the flip-bits selection.

A. PROPOSED CRITERION FOR A FLIP-BITS SELECTION

The flip-bits set denotes the set of unreliable bits that will be flipped in the extra decoding attempts. Instead of selecting flipping candidates based on a metric comparison among different non-frozen bits as in existing approaches, we perform this selection by comparing the reliability of non-frozen bits in SC decoding with their own reliability expectations. The indexes of those non-frozen bits whose estimated BERs derived from LLRs in SC decoding are greater than their BER expectations will be placed into the flip-bits set. The indexes collected in the set are then sorted by their SC

decoding orderings, for the error occurring at an earlier stage of decoding has a more detrimental impact on SC decoding.

GA is a low-complexity algorithm and has found applications in coding and information theory [27], [28]. We employ GA to calculate the BER expectations for non-frozen bits here. Assume that an all-zero codeword is transmitted using BPSK modulation over an AWGN channel. The channel SNR can be estimated by SNR estimation schemes at the receiver [30], [31]. We also assume that the Gaussian noise in the AWGN channel has zero mean and variance σ^2 , which can be derived from the estimated channel SNR. If the average coded symbol energy is normalized to unity, the noise variance σ^2 can be obtained based on the channel SNR γ dB as $\sigma^2 = \frac{1}{2^R \cdot 10^{\gamma/10}}$, where R is the code rate.

After the encoded codeword \mathbf{x} is transmitted through the AWGN channel with mean zero and variance σ^2 , the corresponding LLR value of received \mathbf{y} conditioned on transmitted \mathbf{x} is

$$\text{LLR}(\mathbf{y}) = \ln \frac{p(\mathbf{y}|\mathbf{x} = 0)}{p(\mathbf{y}|\mathbf{x} = 1)} = \frac{2}{\sigma^2}\mathbf{y}. \quad (2)$$

For the AWGN channel, the received LLR message L_N^i can be approximated as a Gaussian variable with mean $\mu_y = \frac{2}{\sigma^2}$ and variance $\delta_y^2 = \frac{4}{\sigma^2} = 2\mu_y$. According to [27], [28], the expectation of LLR, $E[L_N^i]$, can be updated as

$$\begin{aligned} E[L_N^{2i-1}] &= \phi^{-1} \left(1 - \left(1 - \phi \left(E \left[L_{N/2}^i \right] \right) \right)^2 \right) \\ E[L_N^{2i}] &= 2E[L_{N/2}^i], \end{aligned} \quad (3)$$

where

$$\phi(x) \approx \begin{cases} \exp(-0.4527x^{0.86} + 0.0218), & 0 < x < 10 \\ \sqrt{\frac{\pi}{x}} \exp\left(-\frac{x}{4}\right) \left(1 - \frac{10}{7x}\right), & x \geq 10. \end{cases} \quad (4)$$

Hence, we could obtain the BER expectation via GA as [27], [28]

$$\begin{aligned} P_E(i) &\triangleq P(\hat{u}_i \neq u_i | \hat{u}_1^{i-1} = u_1^{i-1}, \mathbf{Y} = \mathbf{y}) \\ &= \frac{1}{2} \text{erfc} \left(\frac{\sqrt{E[L_N^i]}}{2} \right), \end{aligned} \quad (5)$$

where erfc is the complementary error function defined as

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt. \quad (6)$$

Meanwhile, the BER of non-frozen bit u_i in SC decoding can be derived from the decoded LLR \tilde{L}_N^i as [12]

$$\begin{aligned} P_{SC}(i) &\triangleq P(\hat{u}_i \neq u_i | \hat{u}_1^{i-1}, \mathbf{Y}) \\ &= \frac{1}{\exp(|\tilde{L}_N^i|) + 1}. \end{aligned} \quad (7)$$

Note that $P_E(i)$ and $P_{SC}(i)$ are quite different. $P_E(i)$ is the metric for measuring the reliability of non-frozen bit u_i under AWGN channel while $P_{SC}(i)$ depicts the reliability of non-frozen bit u_i under SC decoding.

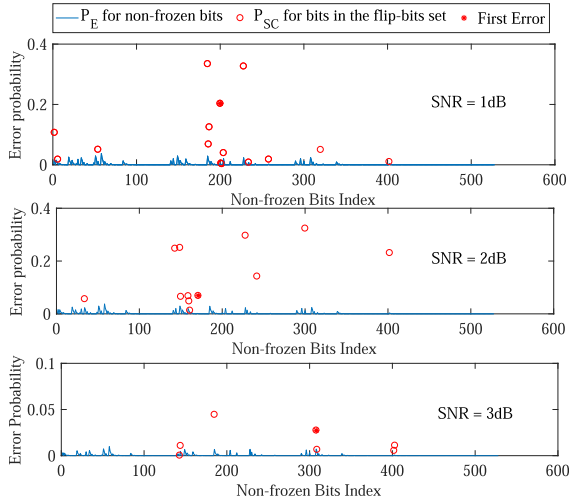


FIGURE 1. The distributions of the flip-bits set ζ for CRC-polar code (1024, 512+16).

If $P_{SC}(i) > P_E(i)$, we deduce that an error may occur in SC decoding of u_i with a high probability. Our later simulation results also validate this hypothesis. According to this analysis, we construct the flip-bits set as $\zeta = \{i \in \mathcal{A} | P_{SC}(i) > P_E(i)\}$. For example, Fig. 1 shows the distributions of a flip-bits set ζ for polar code (1024, 512) concatenated with a 16-bit outer CRC under SC decoding. The blue line denotes P_E of all non-frozen bits and red dots denote P_{SC} of bits whose indexes are selected in the flip-bits set. It is observed that the first error bit marked with a red star falls into the flip-bits set and the capacity of the set decreases when SNR increases.

To further validate the efficiency of the proposed criterion for the flip-bits selection, the frequency of error occurrences for non-frozen bits obtained via Monte-Carlo simulations is compared with their probabilities of being selected in the flip-bits set. For our proposed scheme, the probability of a bit to be selected in our flip-bits set is

$$P(i \in \zeta) = P(P_{SC}(i) > P_E(i)) = P\left(\frac{1}{1 + \exp(-|\tilde{L}_1^i|)} > \frac{1}{2} \operatorname{erfc}\left(\frac{\sqrt{E[L_N^i]}}{2}\right)\right). \quad (8)$$

The simulation results in AWGN channel are depicted in Fig. 2, where (1024, 512+16) and (2048, 1024+16) CRC-polar codes under SNR $\gamma = 1dB$ are decoded by SC decoding. The number of samples is 10^5 . It is observed that in Fig. 2, the probability $P(i \in \zeta)$ has a similar trend with the frequency of channels-induced errors, i.e., the index of the bit with a higher frequency of error occurrences will have a higher probability to be chosen into the flip-bits set. Similar results can be observed for simulations under other SNR values. Note that for the horizontal axis, non-frozen bits are resorted according to their frequency of channels-induced errors in a descending manner and part of bits without any error occurrence is omitted for the limitation of space.

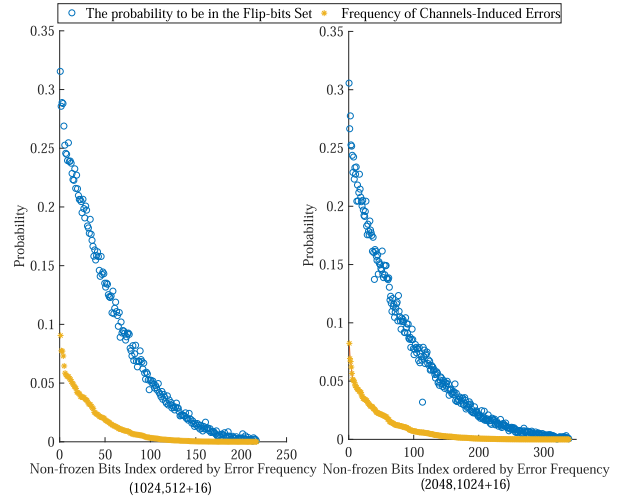


FIGURE 2. The correlation between $P(i \in \zeta)$ and frequency of channels-induced errors.

Algorithm 1 The High-Level of BER-SCFlip Decoding

Input:

- Code length: N
- Received LLR: \mathbf{y}
- Maximum capacity of the flip-bits set: T
- Channel SNR: γ

Output: The decoded bits: \hat{u}_1^N
 $[\hat{u}_1^N, \tilde{L}_1^N] \leftarrow SC \text{ decoding}(\mathbf{y}, N)$

if CRC check(\hat{u}_1^N) fails **then**

$P_E \leftarrow GA_est(\gamma)$

$\zeta \leftarrow List_Con(P_E, \tilde{L}_1^N)$

for $i = 1 : \min(T, |\zeta|)$ **do**

$[\bar{u}_1^N, \tilde{L}_1^N] \leftarrow SCF \text{ decoding}(\mathbf{y}, N, \zeta(i))$

if CRC check(\bar{u}_1^N) succeeds **then**

$\hat{u}_1^N = \bar{u}_1^N$

Break

end if

end for

end if

B. BER-SCFLIP DECODER

The proposed BER-SCFlip decoder performs the same decoding procedure as the conventional SCFlip presented in [17] except that a different criterion is employed for selection and sorting of the flip-bits. The BER-SCFlip decoder first performs SC decoding. If the decoded codeword passes the CRC check, output the decoded result and continue for the next received codeword. Otherwise, BER-SCFlip selects the flip-bits and sorts them by their SC decoding orderings. New decoding attempts proceed and each decoding attempt is performed with one bit flipped starting from the first element in the flip-bits set until the CRC check is satisfied or the maximum number of attempts is reached.

The detail description of the algorithm is illustrated in Algorithm 1. The function GA_est first provides the BER

expectations P_E via GA algorithm according to channel SNR γ dB and function $List_Con$ then generates the flip-bits set according to comparison results between P_{SC} and P_E . The elements in the set are sorted by the SC decoding ordering. The function $SCFdecoding$ denotes SC decoding with flipping bit $\zeta(i)$. To constrain the scope of the flip-bits and avoid unnecessary iterations, we set the maximum number of new decoding attempts as T , which is investigated in the later section.

IV. BER-SCFLIP- ω DECODING

A. ALGORITHM DESCRIPTION

The BER-SCFlip decoder presented in the previous section has a limited error-correction capability, for only one suspicious bit is flipped in each decoding attempt. To remove this limitation, we extend the BER-SCFlip decoder to a generalized scheme (BER-SCFlip- ω) with the capability to correct up to ω error bits per extra decoding attempt. In BER-SCFlip- ω , our proposed BER comparison criterion is combined with that of D-SCFlip presented in [25] to construct and update the flip-bits set. BER-SCFlip- ω could reduce the searching scope of the flip-bits and allow for flipping more than one bits per extra attempt. Thus, it could achieve a good performance close to that of D-SCFlip- ω and SCL decoding with reduced decoding latency and complexity.

In BER-SCFlip- ω , the flip-bits set $\xi = \{\varepsilon_1, \varepsilon_2, \dots\}$ is generalized to include one or multiple indexes of the flip-bits for each element ε_k as in D-SCFlip- ω [25] that will be flipped in extra attempts, e.g., the k -th element $\varepsilon_k = \{i_1, \dots, i_{\omega_k}\}$ contains ω_k indexes of the flip-bits. In the D-SCFlip- ω [25], a novel metric $M(\varepsilon_k)$ to measure the decoding reliability if bits in ε_k were flipped is proposed as

$$M(\varepsilon_k) = \sum_{j=1, j \in \mathcal{A} \setminus \varepsilon_k}^{j < i_{\omega_k}} \ln \left(1 + \exp(-\alpha |\tilde{L}_N^j|) \right) + \sum_{j \in \varepsilon_k} \ln \left(1 + \exp(\alpha |\tilde{L}_N^j|) \right), \quad (9)$$

where α is a coefficient adjusting the weight between decoding orderings and LLR values. Note that (9) is derived from (15) in [25] by taking $-\ln(\cdot)$ operation.

Different from D-SCFlip- ω , we employ a selection criterion for the flip-bits that combines (9) with our proposed BER evaluation condition. Based on this criterion, we develop BER-SCFlip- ω algorithm described in Algorithm 2. The description is similar to D-SCFlip- ω algorithm, except for the function $Init()$ and the function $Update()$ used to construct and update the flip-bits set ξ , respectively. The functions in Algorithm 2 are described as follows.

- *SCdecoding*: A standard SC decoding. If the CRC check is satisfied, output decoded bits \hat{u}_1^N and continue decoding for the next codeword. Otherwise, extra decoding attempts proceed with flipping bits.
- *Init*($\{\tilde{L}_N^i\}_{i \in \mathcal{A}}, P_E, \mathcal{A}, T$): The initialization strategy is described in Algorithm 3 in detail. The preliminary

Algorithm 2 The High-Level of BER-SCFlip- ω Decoding

Input:

- Code length: N
- Received LLR: \mathbf{y}
- Maximum capacity of the flip-bits set: T
- Channel SNR: γ

Output: The decoded bits: \hat{u}_1^N
 $[\hat{u}_1^N, \tilde{L}_1^N] \leftarrow SC\ decoding(\mathbf{y}, N)$

if CRC check(\hat{u}_1^N) fails **then**

$P_E \leftarrow GA_est(\gamma)$

$[\xi_1, \psi_1] \leftarrow Init(\{\tilde{L}_N^i\}_{i \in \mathcal{A}}, P_E, \mathcal{A}, T)$

for $t = 1 : T$ **do**

$[\bar{u}_1^N, \tilde{L}_{i_{\omega_t}+1}^N] \leftarrow SC(\varepsilon_t)$

if CRC check(\bar{u}_1^N) successes **then**

$\hat{u}_1^N = \bar{u}_1^N$

Break

else

$[\xi_{t+1}, \psi_{t+1}] \leftarrow update(\xi_t, \psi_t, P_E)$

end if

end for

end if

Algorithm 3 Initialization of the Flip-Bits Set $[\xi_1, \psi_1] \leftarrow Init(\{\tilde{L}_N^i\}_{i \in \mathcal{A}}, P_E, \mathcal{A}, T)$

$[\xi_0, M_{sc}] \leftarrow Init_List_Con(P_E, \{\tilde{L}_N^i\}_{i \in \mathcal{A}})$

if $|\xi_0| > T$ **then**

$[\xi_1, \psi_1] \leftarrow sort(\xi_0, M_{sc}, T)$

else

$[\xi_1, \psi_1] \leftarrow sort(\xi_0, M_{sc})$

end if

Return $[\xi_1, \psi_1]$

flip-bits set ξ_0 is constructed as $\xi_0 = \{i \in \mathcal{A} | M_{SC}(i) < M_E(i)\}$, where

$$M_E(i) = - \sum_{j=1, j \in \mathcal{A}}^{j < i} \ln(1 - P_E(j)) - \ln(P_E(i)), \quad (10)$$

$$M_{SC}(i) = \sum_{j=1, j \in \mathcal{A}}^{j < i} \ln \left(1 + \exp \left(-\alpha \left| \tilde{L}_N^j \right| \right) \right) + \ln \left(1 + \exp \left(\alpha \left| \tilde{L}_N^i \right| \right) \right). \quad (11)$$

The metric $M_E(i)$ denotes the estimated reliability of bit i via GA algorithm. Meanwhile, the metric $M_{SC}(i)$ denotes the decoding reliability after SC decoding if bit i was flipped. Note that (11) is simplified from (9) by considering only one element in ε_k . If $M_{SC}(i) < M_E(i)$, we deduce that the first error may occur in bit i with a high probability. If the number of bits in ξ_0 exceeds T , the set ξ_0 only retains T elements with the lowest metrics $M_{SC}(i)$. Then the bits in the set ξ_0 are sorted by their metrics $M_{SC}(i)$ in an ascending manner

Algorithm 4 Update of the Flip-Bits Set $[\xi_{t+1}, \psi_{t+1}] \leftarrow \text{update}(\xi_t, \psi_t, P_E)$

```

 $\zeta(i_{w_t}) \leftarrow \text{List\_Con}(P_E, \bar{L}_{i_{w_t}+1}^N)$ 
for  $i = i_{w_t} + 1, \dots, N$  and  $i \in \zeta(i_{w_t})$  do
   $\varepsilon' = \varepsilon_t \cup \{i\}$ 
   $m = M(\varepsilon')$ 
  if  $m < M(\varepsilon_T)$  then
     $[\xi_{t+1}, \psi_{t+1}] \leftarrow \text{Insert\_flip}(\psi(\varepsilon'), \varepsilon')$ 
  end if
end for
Return  $[\xi_{t+1}, \psi_{t+1}]$ 

```

to generate an initialized set $\xi_1 = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_\ell\} = \{i_1, i_2, \dots, i_\ell\}$, where $\ell = \min(|\xi_0|, T)$. Each element in ξ_1 contains an index of a bit, i.e., $\varepsilon_k = \{i_k\}$. The corresponding metric values are collected into a set $\psi_1 = \{M_{SC}(\varepsilon_1), M_{SC}(\varepsilon_2), \dots, M_{SC}(\varepsilon_\ell)\}$ for $M_{SC}(\varepsilon_i) \leq M_{SC}(\varepsilon_j)$, $1 \leq i \leq j \leq \ell$.

- $SC(\varepsilon_t)$: The t -th extra SC decoding that performs the same operation as the conventional SC decoding except those decoded bits with indexes in the set ε_t are flipped. If the CRC check is satisfied, output the decoded codeword \hat{u}_1^N and continue the decoding of the next codeword. Otherwise, update the flip-bits set ξ_t to ξ_{t+1} and the corresponding metrics set ψ_t to ψ_{t+1} . Then continue the following decoding attempt $SC(\varepsilon_{t+1})$.
- $\text{update}(\xi_t, \psi_t, P_E)$: The update function, described in Algorithm 4, calculates $M(\varepsilon')$ using the LLR values after unsuccessful $SC(\varepsilon_t)$ decoding for flipping elements $\varepsilon' = \varepsilon_t \cup \{i\}$, where $i \in \zeta(i_{w_t})$ and $\zeta(i_{w_t}) = \{i \in \mathcal{A} | P_{SC}(i) > P_E(i), i > i_{w_t}\}$ (recall i_{w_t} is the last bit in the flipping element ε_t). If $M(\varepsilon') < M(\varepsilon_T)$, where ε_T is the last element in ξ_t , insert ε' and $M(\varepsilon')$ into proper positions of ξ_t and ψ_t respectively according to their metric values. Since $M(\varepsilon_t) < M(\varepsilon')$, ε' will be inserted into a position behind t . Otherwise, $\xi_{t+1} = \xi_t$ and $\psi_{t+1} = \psi_t$. In this way, the updated sets ξ_{t+1} and ψ_{t+1} are obtained.

The main difference between the proposed BER-SCFlip- ω and D-SCFlip- ω lies in the strategies to initialize and update the flip-bits set. The initial construction of the flip-bits set in D-SCFlip- ω needs to sort all information bits by their metrics. In our proposed scheme, we only need to sort those bits satisfying $M_{SC}(i) < M_E(i)$. Thus, the number of bits to be sorted can be dramatically reduced. The update function in D-SCFlip- ω , calculates $M(\varepsilon)$ for all flipping elements $\varepsilon = \varepsilon_t \cup \{i\}$, where $i \in \mathcal{A}$ and $i > i_{w_t}$, which is a heavy computational burden in practice. In contrast, the update function in BER-SCFlip- ω calculates $M(\varepsilon')$ for a limited number of flipping elements $\varepsilon' = \varepsilon_t \cup \{i\}$, where $i \in \zeta(i_{w_t}) = \{i \in \mathcal{A} | P_{SC}(i) > P_E(i), i > i_{w_t}\}$. By introducing the constraint $P_{SC}(i) > P_E(i)$, the number of new elements whose metrics to be measured and sorted at each update is significantly

reduced so that the computational burden can be mitigated as well.

Therefore, the proposed BER-SCFlip- ω has a much lower sorting and metric computation complexity compared to D-SCFlip- ω , which will be analyzed in the following section.

B. THE OPTIMIZATION OF PARAMETER α

The parameter α in (9) and (11) is used to adjust the weight between the decoding ordering and LLR values in the criterion for selecting the flip-bits. When $\alpha = 1$, the selection of the flip-bits set mainly depends on LLR values. When $\alpha = 0$, the selection of the flip-bits set only depends on the position of the bit in the decoding ordering.

In [25], the optimal parameter α_{opt} was first obtained via Monte-Carlo simulations. For simplification, α_{opt} was then calculated by modelling it approximately as a quadratic function of iWER-0.

The optimal parameter α_{opt} in our scheme determined by Monte-Carlo simulations is different from that obtained in [25], for that the different criteria will affect the optimal value of α_{opt} . In this section, we may also obtain the optimal α_{opt} by modelling it as a function of some key parameters determining iWER-0, such as code length, rate and SNR.

Assume that the code rate of polar codes remains the same, e.g., $1/2$, the influence of code lengths on α_{opt} is first investigated. As Fig. 3(a) shows, for codes with different lengths ($N = 512, 1024, 2048$), there does not exist a big gap among the optimized α_{opt} . Similar results can be observed from simulations under other rates. Thus, we can neglect the impact of code lengths on the parameter α_{opt} . When the code lengths are fixed, e.g., $N = 1024$, the influence of different rates on α_{opt} can be visualized in Fig. 3(b).

It is observed that higher rate codes tend to have a relatively higher optimized parameter α_{opt} . Meanwhile, the optimized parameter α_{opt} tends to decrease with increased SNR and converges to a floor value in an exponential manner. Thus, we present an approximate formula for the optimized $\bar{\alpha}_{opt}(R, \gamma)$ as

$$\bar{\alpha}_{opt}(R, \gamma) \triangleq \min \left(a \cdot e^{f(R, \gamma)} + b \cdot e^{g(\gamma)} + c, 1 \right). \quad (12)$$

We obtain many data points depicting the correlation of $\alpha_{opt}^{(i)}$ with SNR $\gamma^{(i)}$ dB and rate $R^{(i)}$ as $\left\{ \left(\gamma^{(i)}, R^{(i)} \right); \alpha_{opt}^{(i)} \right\}$ via Monte-Carlo simulations. The parameters a, b, c and linear functions $f(R, \gamma), g(\gamma)$ are obtained via minimizing the fitting errors. The results are $a = 0.0015$, $b = 10$, $c = 0.35$ and $f(R, \gamma) = 18.4208R - 2.3026\gamma$, $g(\gamma) = -3.1775\gamma$. Thus, $\bar{\alpha}_{opt}(R, \gamma)$ is expressed as

$$\bar{\alpha}_{opt}(R, \gamma) \triangleq \min(0.0015 \cdot e^{(18.4208R - 2.3026\gamma)} + 10e^{-3.1775\gamma} + 0.35, 1). \quad (13)$$

The equation (13) could be used to facilitate the parameter optimization and the algorithm design.

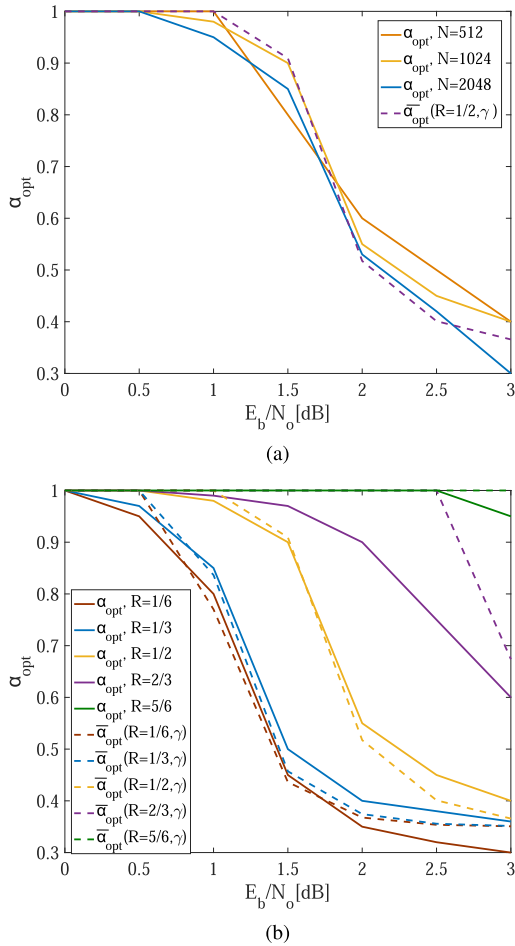


FIGURE 3. The optimized parameter α_{opt} and $\bar{\alpha}_{opt}(R, \gamma)$ with different lengths N , rates R at various SNRs.

V. THE PERFORMANCE ANALYSIS

In this section, we evaluate the WER performance of the proposed SCFlip for the decoding of CRC-polar codes by deriving its lower and upper bounds. The lower bound of SCFlip decoders has been obtained in [25] by calculating the WER of OASC decoder, the perfect SCFlip decoder. So we focus on calculating its upper bound in the following.

SCFlip decoding is a sequential version of SCL decoding. Instead of expanding paths and finding the most probable path among many parallel choices in SCL, SCFlip decoder flips the most unreliable bits sequentially once the decoded result fails the CRC check. If the estimated codeword passes the CRC check, the decoding will be announced success. However, a successful CRC check does not always mean a correct decoding since an erroneous codeword may pass the CRC check if it is beyond the CRC check capability (Undetected Error). Similarly, an unsuccessful CRC check does not always mean an incorrect decoding, for that the decoding error of redundant bits added by CRC coding may fail the CRC check despite all information bits being decoded successfully (False Alarm). The CRC check result determines the start and end of the decoding process. Thus, the WER performance of the

proposed SCFlip decoding depends on the probability of the CRC check failure, including the probability of the False Alarm P_{FA} and Undetected Error P_{UE} .

Let P_{SC} denote the probability of SC decoding failure, t^* be the number of additional decoding attempts before outputting a correct codeword. We write T to denote the maximum number of extra decoding attempts. The WER of proposed SCFlip, P_{BSCF} , can be approximately expressed as

$$P_{BSCF} = P_{SC} - P_{SC} (1 - P_{FA}) \sum_{t=1}^T P(t^* = t) (1 - P_{UE})^{t-1}, \quad (14)$$

where $P(t^* = t)$, the probability of correct decoding at the t -th additional attempt, can be obtained by Monte-Carlo simulations, while P_{FA} and P_{UE} , the probabilities related to CRC, can be approximately estimated by the method in [32]. Assume the error probability of the information bit $p \in [0, 0.5]$, the AWGN channel with p can be approximately converted into a discrete binary symmetry channel(BSC) with $p = 1 - \int_{-\infty}^{\frac{1}{\delta}} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$, where δ is the variance of the channel noise.

The CRC check has a False Alarm only when any bit in r CRC redundancy bits is decoded incorrectly yet all information bits are correctly estimated. Thus, P_{FA} can be expressed as

$$P_{FA} = \sum_{i=1}^r \binom{r}{i} p^i (1-p)^{k+r-i}. \quad (15)$$

According to [32], the occurrence probability of the undetected error can be

$$P_{UE} = \sum_{i=1}^{k+r} A_i p^i (1-p)^{k+r-i}, \quad (16)$$

where $A_i = |\{c|c \in C, w(c) = i\}|$ denotes the number of codes c with weight i in the codebook C and k is the information length of CRC-polar codes. The weight distribution A_i is obtained by enumerating all short length codewords or SCL with an extremely large list for relatively long length codewords [32]. Using $P_{UE} \leq (2^k - 1) 2^{-k-r} \leq 2^{-r}$, we get the upper bound of P_{BSCF} according to [32], [33].

$$P_{BSCF} \leq P_{SC} - P_{SC} (1 - P_{FA}) \sum_{t=1}^T P(t^* = t) (1 - 2^{-r})^{t-1}. \quad (17)$$

$$P_{OASC} \leq P_{BSCF} \leq P_{SC} \times \left[1 - (1 - P_{FA}) \sum_{t=1}^T P(t^* = t) (1 - 2^{-r})^{t-1} \right] \quad (18)$$

Thus, we can get the lower bound and upper bound of the WER for our proposed SCFlip decoding in (18), where P_{OASC} is the error probability of the perfect OASC decoding.

TABLE 1. The CRC codes and corresponding generator polynomials.

CRC codes	Generator polynomials
CRC-4	$x^4 + x^3 + x^2 + x + 1$
CRC-8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-24	$x^{24} + x^{23} + x^6 + x^5 + x + 1$

From (17), we can observe that a long-length CRC code may lead to a low upper bound of the WER for our proposed scheme. But the inordinate long length may cause the negative effect of the rate loss. Thus, to evaluate the effect of the CRC check (redundancy) length r on the decoding performance, we perform the simulations of BER-SCFlip- ω ($\omega \rightarrow \infty$) decoding for various CRC redundancy bits under (1024, 512) CRC-Polar codes. The maximum number of new decoding attempts T is 100. Note that both the CRC code rate $R_c = \frac{k}{k+r}$ and the polar code rate $R_p = \frac{k+r}{N}$ will change with r but the overall code rate R remains fixed. The generator polynomials of different CRC encoders are listed in Table 1 and the simulation result is shown in Fig. 4.

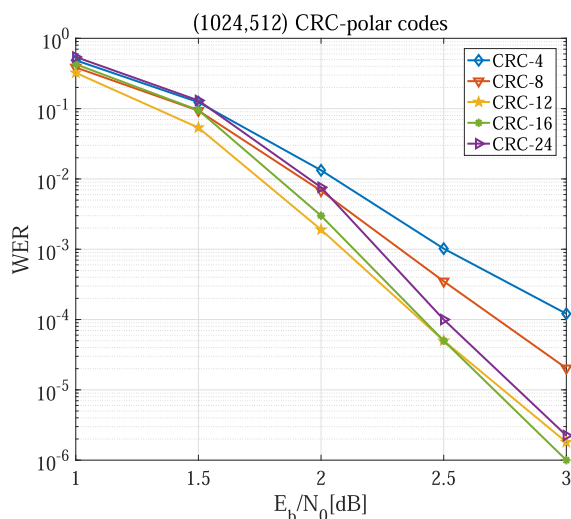


FIGURE 4. The WER performance of BER-SCFlip- ω decoding with various CRC polynomials for (1024,512) CRC-polar codes.

It is observed that the protection of polar codes by a relatively longer CRC code could provide significant improvement in the WER performance, e.g., polar codes concatenated with CRC-12 achieve a 0.4dB gain at the WER of 10^{-3} over those with CRC-4. However, excessively long-length redundancy bits, e.g., $r = 24$, give rise to a degraded WER performance. The reason is that the high polar code rate R_p degrades the error correction capability of polar codes. Thus, a moderate CRC code length is recommended, e.g., CRC-12 or CRC-16 for CRC-Polar codes of length 1024 and rate 1/2.

VI. THE SIMULATIONS AND ANALYSIS

In this section, our proposed schemes are compared via simulations with some existing SCFlip decoding algorithms

in terms of the WER performance and the computational complexity to validate the advantages of the proposed BER-SCFlip decoder and BER-SCFlip- ω decoder.

A. EVALUATION OF BER-SCFLIP DECODER

First, we compare BER-SCFlip decoder with other existing SCFlip decoding algorithms where only one bit will be flipped in one decoding attempt, including original LLR-SCFlip decoder in [17], D-SCFlip decoder in [25], Progressive SCFlip decoder in [19] and Improved SCFlip decoding in [20]. The comparisons are given in terms of the accuracy of targeting the first error bit, the WER performance and the computational complexity. For the fairness of comparison, the simulations are performed under the same code lengths and rates for all decoders: (1024,512) and (2048,1024) polar codes with CRC-16. The maximum number of additional decoding attempts is set to be $T = 10$.

We first perform extensive Monte Carlo simulations of 10^6 samples in experiments to obtain the statistical results about the probability of correctly targeting the first error bit and the average position of the estimated first error bit in the flip-bits set. The results are listed in Table 2. It is observed that all SCFlip decoders have good error-correction performance with reduced additional decoding attempts as SNR increases. It is also shown that the proposed SCFlip decoder can target the first error bit more precisely and more timely than all other SCFlip decoders at low SNR ($\gamma < 2dB$). At high SNR ($\gamma \geq 2dB$), the proposed decoder performs better than other decoders except for D-SCFlip, which can target the first error bit a little bit earlier than the proposed decoder.

We then give a WER comparison among advanced SCFlip decoding algorithms in Fig. 5. It is observed that the proposed BER-SCFlip and D-SCFlip outperform others and could approach the performance of OASC decoding. At low SNR ($\gamma \leq 1dB$), where more than one errors will occur with a high probability due to the strong channel noise, the performance gain is minor when only one bit can be flipped in additional decoding attempts.

To evaluate the computational burden incurred by additional decoding attempts, we calculate the average number of new attempts before successful decoding, which is listed for different SCFlip decoders in Table 3. It is observed that the proposed scheme has fewer additional decoding attempts and thus a lower complexity except for D-SCFlip, which shows a minor advantage over the proposed decoder under high SNR ($\gamma > 2dB$). Considering that the proposed BER-SCFlip employs a lower complexity criterion for targeting the error bits than D-SCFlip, the overall computational complexity of the proposed BER-SCFlip is lower than D-SCFlip, which will be elaborated in the later complexity analysis.

B. COMPLEXITY ANALYSIS FOR BER-SCFLIP DECODER

The computational burden includes both additional decoding attempts and the computational complexity for selecting unreliable bits in the construction of the flip-bits set. In Table 3, it is shown that the complexity difference in the

TABLE 2. The accuracy and average position of the estimated first error.

SNR(dB)	(1024,512+16) CRC-polar codes						(2048,1024+16) CRC-polar codes					
	The accuracy			The average position			The accuracy			The average position		
	1	2	3	1	2	3	1	2	3	1	2	3
LLR-SCFlip [17]	0.70	0.88	0.98	5.93	4.09	2.19	0.56	0.83	0.97	6.88	4.66	2.25
D-SCFlip [25]	0.47	0.96	1.00	7.13	2.15	1.34	0.40	0.98	1.00	7.50	2.91	1.30
Progressive SCFlip [19]	0.47	0.95	0.99	6.37	2.88	2.12	0.40	0.96	0.98	6.90	2.97	2.14
Improved SCFlip [20]	0.56	0.95	0.99	5.77	3.76	2.24	0.56	0.88	0.99	7.22	3.54	1.59
Proposed BER-SCFlip	0.83	0.95	1.00	5.01	2.18	1.84	0.79	0.97	1.00	5.68	2.99	1.48

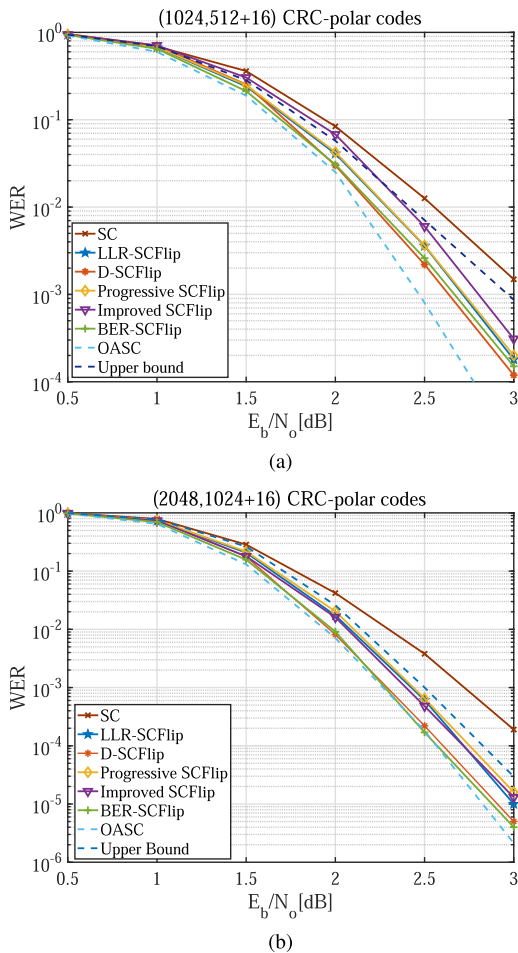


FIGURE 5. The WER comparison for advanced SCFlip decoding algorithms.

number of extra attempts is minor since their probabilities of starting a new attempt are close. Thus, we turn to the complexity analysis in terms of different strategies for flip-bits selection in the following.

Denote the length of non-frozen bits by K . In the conventional SCFlip, LLR’s absolute values are used as metrics for the flip-bits selection, thus no additional computation is needed for the metric calculation and the sorting complexity is $O(K \log K)$. In D-SCFlip, in addition to the sorting complexity $O(K \log K)$, K multiplications and $\frac{(K+3)K}{2}$ additions are needed to compute metrics for each new decoding attempt, excluding the complexity of logarithm and exponential computations due to their low-complexity operations of

TABLE 3. The average number of additional decoding attempts.

SNR(dB)	(1024,512+16) CRC-polar codes			(2048,1024+16) CRC-polar codes		
	1	2	3	1	2	3
LLR-SCFlip [17]	7.358	0.746	0.010	7.797	0.335	0.0010
D-SCFlip [25]	7.617	0.538	0.006	7.950	0.176	0.0006
Progressive SCFlip [19]	7.412	0.775	0.010	7.880	0.355	0.0010
Improved SCFlip [20]	7.602	0.827	0.018	7.872	0.278	0.0007
Proposed BER-SCFlip	7.174	0.522	0.008	7.566	0.141	0.0006

the look-up table. In contrast, Progressive SCFlip decoding has a much narrower scope of the critical set, so the sorting complexity can be reduced to $O(|S| \log |S|)$, where $|S|$ denotes the capacity of the critical set S . Besides, it needs to build a tree structure and search the tree to construct the critical set S with at least $O(N)$ complexity. Improved SCFlip decoding needs massive simulations to construct the critical set before the decoding. Thus, its pre-decoding complexity is $O(MN \log N)$, where M is the number of samples in the pre-experiment. Also, the comparison of LLR values with the determined threshold costs $O(K)$ computational complexity. The proposed BER-SCFlip needs to compute BERs from decoded LLR values for the flip-bits selection, resulting in K multiplications and K additions. The complexity of calculating the BER expectation via GA algorithm is $O(N)$ [28]. Also, the BER comparison of non-frozen bits costs $O(K)$ complexity. Besides, the sorting complexity can be omitted, since the set is sorted by the decoding ordering and no extra operation is needed.

In summary, the complexity comparison in the construction of the flip-bits set is listed in Table 4, where we can observe that our proposed BER-SCFlip has a relatively lower complexity in the construction of the flip-bits set.

C. THE EVALUATION OF BER-SCFLIP- ω ALGORITHM

BER-SCFlip- ω allows more than one channel-incurred errors to be corrected in one new decoding attempt with dynamic updating of the flip-bits set, which is obtained by incorporating the proposed criterion with D-SCFlip- ω . Thus, BER-SCFlip- ω could achieve a strong error-correction performance with reduced complexity. We provide simulations to compare the WER performance of the proposed BER-SCFlip- ω , D-SCFlip- ω [25],

TABLE 4. The complexity in the construction of flip-bits set.

	Sorting	Metric Computation	Comparison	Overall
LLR-SCFlip [17]	$O(K \log K)$	-	-	$O(K \log K)$
D-SCFlip [25]	$O(K \log K)$	$O(K^2)$	-	$O(K^2)$
Progressive SCFlip [19]	$O(S \log S)$	$O(N)$	-	$O(N + S \log S)$
Improved SCFlip [20]	-	$O(MN \log N)$	$O(K)$	$O(MN \log N)$
Proposed BER-SCFlip	-	$O(N)$	$O(K)$	$O(N)$

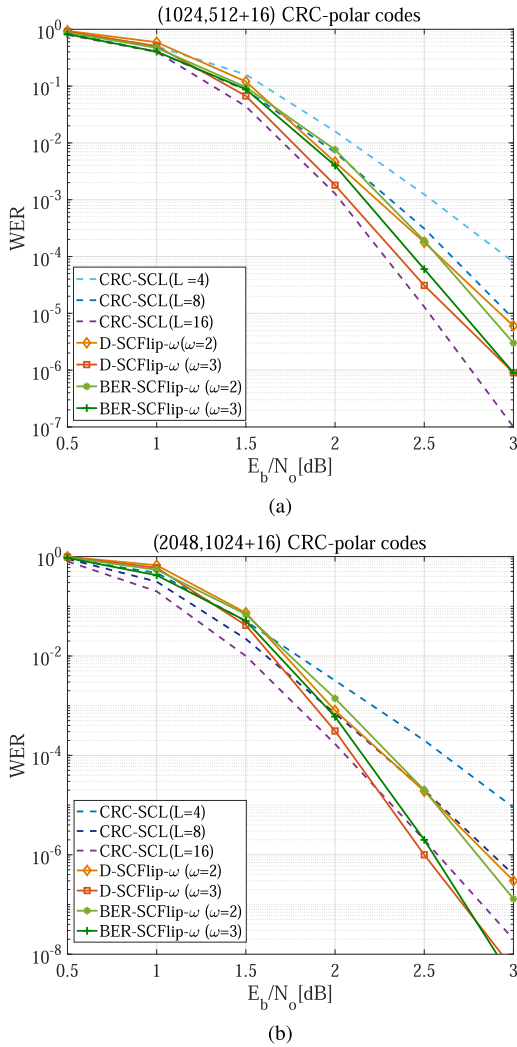


FIGURE 6. The WER comparison among SCFlip decoders and SCL decoders.

and CRC-SCL decoder [6]. All simulations are performed under AWGN channel. CRC-polar codes (1024,512+16) and (2048,1024+16) are used for simulations, where CRC-16 in Table 1 is used. We construct polar codes using GA method presented in [28]. The maximum number of new decoding attempts T is set to be 100 and 300 for $\omega = 2$ and $\omega = 3$, respectively.

1) THE PERFORMANCE COMPARISON

The WER performance comparison results are presented in Fig.6. It is observed from Fig.6 that for $\omega = 2$, the proposed

BER-SCFlip- ω can achieve a better WER performance than D-SCFlip- ω for both CRC-polar codes (1024,512+16) and (2048,1024+16) under high SNR ($\gamma > 2.5dB$). Moreover, They both surpass CRC-SCL decoder with $L = 8$. When ω increases to 3, their performance could approach that of the CRC-SCL with $L = 16$ for $N = 1024$ and even surpass it for $N = 2048$ at high SNR ($\gamma > 2.5dB$). In addition, in contrast with D-SCFlip- ω , the effect of ω values on the performance gain is less obvious for our scheme when $N = 1024$, while for $N = 2048$ codes, the gap is filled.

2) THE COMPLEXITY COMPARISON

The computational complexity of BER-SCFlip- ω and D-SCFlip- ω includes not only the additional decoding complexity but also the computation complexity in the construction of the flip-bits set.

The average number of additional decoding attempts, T_{ave} , is listed in Table 5 for the comparison between BER-SCFlip- ω and D-SCFlip- ω decoders. It is observed that the proposed BER-SCFlip- ω decoder could achieve successful decoding with much fewer average additional attempts than D-SCFlip- ω , especially at low SNR ($\gamma \leq 2dB$), e.g., the gap is over 10 for $\omega = 2$ and over 40 for $\omega = 3$ at SNR $\gamma = 1dB$. At high SNR ($\gamma > 2dB$), the average number of extra attempts for the successful decoding is similar for both decoders.

The computational complexity for the construction of the flip-bits set mainly includes the metric computation and sorting complexity. Denote the number of non-frozen bits by K . In the initialization of the flip-bits set, the number of elements whose metrics to be calculated is K for both decoders. Thus, their metric calculation complexity is both $O(K^2)$ ¹ as Table 6 shows. Before sorting, the metric comparison in BER-SCFlip- ω will cost extra $O(K)$ complexity. However, with the metric comparison, the number of elements to be sorted is reduced to $|\xi_0|$, while D-SCFlip- ω needs to sort K elements in the initial step. Therefore, the sorting complexity of BER-SCFlip- ω and D-SCFlip- ω will be $O(|\xi_0| \log |\xi_0|)$ and $O(K \log K)$, respectively.

Denote the number of non-frozen bits decoded after bit i_{ω_t} by τ . Recall that bit i_{ω_t} is the last bit in ϵ_t in Algorithm 4. When updating the flip-bits set in our algorithm, the BER-based comparison with metric computation

¹ Note that the metric calculation in BER-SCFlip- ω includes the calculation of metrics M_E and M_{SC} , and the BER expectation via GA algorithm P_E . The metric calculation complexity remains $O(K^2)$.

TABLE 5. The average number of additional decoding attempts T_{ave} .

codes	(1024,512+16) codes						(2048,1024+16) codes					
	$\omega = 2$			$\omega = 3$			$\omega = 2$			$\omega = 3$		
SNR(dB)	1	2	3	1	2	3	1	2	3	1	2	3
D-SCFlip- ω [25]	64.710	1.464	0.006	177.481	1.813	0.006	72.550	0.367	0.0006	201.313	0.339	0.0004
Proposed BER-SCFlip- ω	50.888	1.115	0.005	128.515	1.714	0.006	55.915	0.325	0.0006	142.445	0.317	0.0004

TABLE 6. The complexity in the construction and updating of flip-bits set.

		Metric Computation	Comparison	Sorting
Initial-ization	D-SCFlip- ω	$O(K^2)$	-	$O(K \log K)$
	BER-SCFlip- ω	$O(K^2)$	$O(K)$	$O(\xi_0 \log \xi_0)$
Update	D-SCFlip- ω	$O(\tau^2)$	-	$O(\tau \log \tau)$
	BER-SCFlip- ω	$O(\tau)$	$O(\tau)$	$O(\nu \log \nu)$

TABLE 7. The average number of elements to be processed when SNR $\gamma = 2dB, \omega = 3$.

	Initialization		Update			
	K	$ \xi_0 $	$\bar{\tau}_1$	$\bar{\nu}_1$	$\bar{\tau}_2$	$\bar{\nu}_2$
(1024,512+16) CRC-polar codes	528	189	410	34	395	37
(2048,1024+16) CRC-polar codes	1040	457	747	56	782	74

complexity $O(\tau)$ and comparison complexity $O(\tau)$ can further reduce the number of elements to be sorted from τ to ν . In contrast, the metric computation and sorting in D-SCFlip- ω are based on τ elements. The complexity comparison in the updating of the flip-bits set is given in Table 6. It can be observed from Table 6 that the proposed BER-SCFlip- ω has a lower computation burden for constructing and updating the flip-bits set than D-SCFlip- ω .

To further verify the efficiency of the metric and BER-based comparison, the average number $|\xi_0|$ is collected, as well as $\bar{\nu}_i$ where i denotes the number of bits in ε_i . Table 7 presents $|\xi_0|$ and $\bar{\nu}_i$ for both CRC-polar codes (1024,512+16) and (2048,1024+16) when SNR $\gamma = 2dB$ and $\omega = 3$. The average number $\bar{\tau}_i (i = |\varepsilon_i|)$ is also given for comparison. As Table 7 shows, the introduction of the metric and BER-based comparison could bring a significant reduction in the number of elements to be calculated and sorted. Similar results can be observed for simulations under other SNR values.

Taking into account that the proposed BER-SCFlip- ω has a lower number of additional decoding attempts than D-SCFlip- ω , the former has a lower overall complexity and time delay than the later without much loss in the error-correction performance.

To evaluate the constraint T , the maximum number of new decoding attempts, on the WER performance, we conduct simulations to compare the WER of BER-SCFlip- $\omega (\omega \rightarrow \infty)$ with CRC-SCL decoding algorithm under $L = 4, 8, 16$ for reference. Fig. 7 shows the WER performance curves for (1024,512+16) and (2048, 1024+16) CRC-polar codes

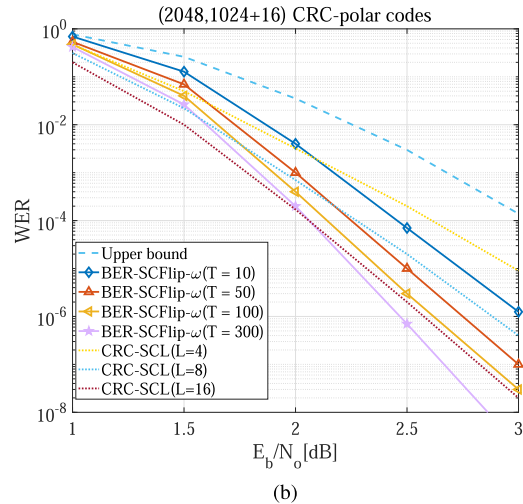
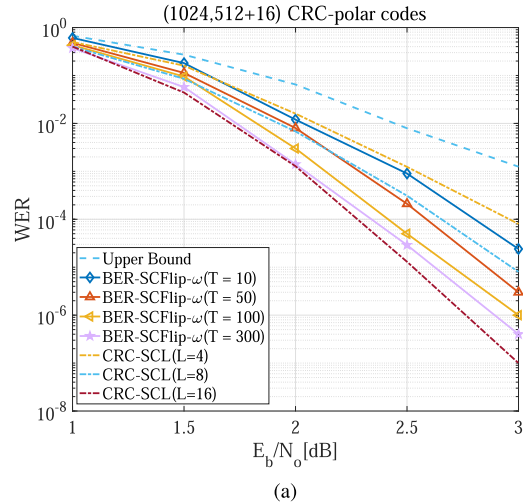


FIGURE 7. The WER performance of BER-SCFlip- ω decoding with various T and CRC-SCL decoding with different L .

using BER-SCFlip- ω decoding with different T values, and CRC-SCL decoding with various list sizes. We can observe that BER-SCFlip- ω decoding with $T = 10$ outperforms CRC-SCL decoding with $L = 4$, while BER-SCFlip- ω with $T = 50$ could surpass CRC-SCL decoding with $L = 8$. The performance of BER-SCFlip- ω decoding with $T = 300$ approaches CRC-SCL with $L = 16$ for $N = 1024$ and could compete with it for $N = 2048$.

Comparing to SCL decoding complexity $O(LN \log N)$, the proposed BER-SCFlip- ω has an approximate complexity $O(T_{ave}N \log N)$, since the computational complexity in

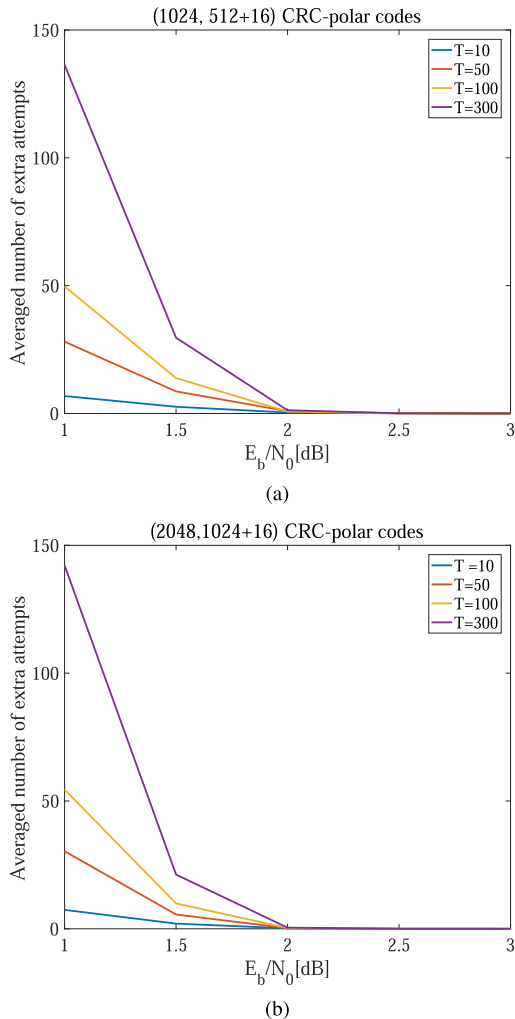


FIGURE 8. The average number of extra decoding attempts T_{ave} for BER-SCFlip- ω decoding with various T .

the construction of the flip-bits set can be negligible with respect to SC decoding complexity. From Fig. 8, we observe that T_{ave} , the average number of extra decoding attempts, can be automatically adapted to SNR variations. At very low SNR ($\gamma < 1dB$), BER-SCFlip- ω has a higher complexity than SCL decoding. However, At high SNR ($\gamma > 2dB$), T_{ave} approaches 1 and BER-SCFlip- ω has nearly the same complexity as SC decoding, which is much lower than SCL decoding. Thus, the proposed criterion can be applied to SCL decoding for the path selection, which may speed up the decoding by narrowing the selection scope. We will investigate this issue in our future research.

VII. CONCLUSION

A BER evaluation based SCFlip decoder for polar codes is presented. By estimating the BER expectations of non-frozen bits via GA and comparing them with their corresponding BERs derived from LLR values in SC decoding, the error bits of SC decoding can be pinpointed accurately. Additionally, by combining our proposed criterion with that presented in D-SCFlip [25] for the flip-bits sets initialization and update,

an extended BER-SCFlip- ω is proposed to correct multiple bits per decoding attempt such that a better error-correction performance is achieved. Thanks to the constraint we introduce, the scope of candidate bits in the constructing and updating the flip-bits set is significantly reduced and thus a low complexity, storage cost and decoding latency can be achieved.

REFERENCES

- [1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
- [2] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 254–266, Feb. 2016.
- [3] N. Presman, O. Shapira, and S. Litsyn, "Mixed-kernels constructions of polar codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 239–253, Feb. 2016.
- [4] F. Gabry, V. Bioglio, I. Land, and J.-C. Belfiore, "Multi-kernel construction of polar codes," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 761–765.
- [5] K. Qin and Z. Zhang, "Polar coding for arbitrary channels based on Gaussian decomposition," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.
- [6] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- [7] K. Chen, K. Niu, and J. Lin, "Improved successive cancellation decoding of polar codes," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3100–3107, Aug. 2013.
- [8] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.
- [9] *Technical Specification Group Radio Access Network*, document TS 38.212, V.15.0.0, 3GPP, Spreading and Modulation, 2017.
- [10] T. Wang, D. Qu, and T. Jiang, "Parity-check-concatenated polar codes," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2342–2345, Dec. 2016.
- [11] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "From polar to Reed-Müller codes: A technique to improve the finite-length performance," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3084–3091, Sep. 2014.
- [12] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.
- [13] H. Zhou, C. Zhang, W. Song, S. Xu, and X. You, "Segmented CRC-aided SC list polar decoding," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.
- [14] C. Xiong, J. Lin, and Z. Yan, "A multimode area-efficient SCL polar decoder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 12, pp. 3499–3512, Dec. 2016.
- [15] D. Kim and I.-C. Park, "A fast successive cancellation list decoder for polar codes with an early stopping criterion," *IEEE Trans. Signal Process.*, vol. 66, no. 18, pp. 4971–4979, Sep. 2018.
- [16] S. A. Hashemi, M. Mondelli, S. H. Hassani, C. Condo, R. L. Urbanke, and W. J. Gross, "Decoder partitioning: Towards practical list decoding of polar codes," *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 3749–3759, Sep. 2018.
- [17] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2014, pp. 2116–2120.
- [18] P. Giard and A. Burg, "Fast-SSC-flip decoding of polar codes," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2018, pp. 73–77.
- [19] Z. Zhang, K. Qin, L. Zhang, and G. T. Chen, "Progressive bit-flipping decoding of polar codes: A critical-set based tree search approach," *IEEE Access*, vol. 6, pp. 57738–57750, 2018.
- [20] F. Ercan, C. Condo, and W. J. Gross, "Improved bit-flipping algorithm for successive cancellation decoding of polar codes," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 61–72, Jan. 2019.

- [21] F. Ercan, C. Condo, S. A. Hashemi, and W. J. Gross, "Partitioned successive-cancellation flip decoding of polar codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
- [22] Y. Yongrun, P. Zhiwen, L. Nan, and Y. Xiaohu, "Successive cancellation list bit-flip decoder for polar codes," in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2018, pp. 1–6.
- [23] F. Cheng, A. Liu, Y. Zhang, and J. Ren, "Bit-flip algorithm for successive cancellation list decoder of polar codes," *IEEE Access*, vol. 7, pp. 58346–58352, 2019.
- [24] L. Chandesaris, V. Savin, and D. Declercq, "An improved SCFlip decoder for polar codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
- [25] L. Chandesaris, V. Savin, and D. Declercq, "Dynamic-SCFlip decoding of polar codes," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2333–2345, Jun. 2018.
- [26] R. Mori and T. Tanaka, "Performance of polar codes with the construction using density evolution," *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 519–521, Jul. 2009.
- [27] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.
- [28] D. Wu, Y. Li, and Y. Sun, "Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation," *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1099–1102, Jul. 2014.
- [29] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.
- [30] K. Liang, B. Feng, J. Jiao, S. Wu, and Q. Zhang, "An efficient channel estimation scheme based on frozen bits of polar codes over fading channel," in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2018, pp. 1–6.
- [31] Y. Li, R. Liu, and R. Wang, "A low-complexity SNR estimation algorithm based on frozen bits of polar codes," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2354–2357, Dec. 2016.
- [32] M. Xu, P. Chen, B. Bai, and S. Tong, "Distance spectrum and optimized design of concatenated polar codes," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6.
- [33] T. Murata and H. Ochiai, "On design of CRC codes for polar codes with successive cancellation list decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1868–1872.



XUETING ZHANG received the B.S. degree in communication engineering and the M.S. degree in communication and information system from South Central University for Nationalities, Wuhan, China, in 2008 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Department of Electronics and Information Engineering, Huazhong University of Science and Technology (HUST), Wuhan. Her research interests include channel coding theory, information theory, and wireless networks.



YINGZHUANG LIU received the Ph.D. degree in communication and information engineering from HUST, China, in 2000. Before, he has worked at HUST from 2002, he was a Postdoctoral Researcher with Paris University XI, France. He is currently a Professor with the Department of Electronics and Information Engineering, Huazhong University of Science and Technology (HUST). His research field is in the area of mobile communication (3G, LTE, and IMT-A).



SHAOPING CHEN received the B.S. and M.S. degrees in electrical engineering from Wuhan University, Wuhan, China, in 1987 and 1990, respectively, and the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology, Wuhan, in 2004. He is currently a Professor with the Hubei Key Laboratory of Intelligent Wireless Communications, South Central University for Nationalities, Wuhan. His research interests include the communications and signal processing, including transceiver design, detection and estimation, and wireless and optical communication systems.

...