mathematics

MDPI

Article

An Error Segment Bit-Flip Algorithm for Successive Cancellation List Decoding of Polar Codes

Jing Zhang [1,2,*], Hongbing Qiu [3], Ling Zhan [3] and Lili Chen [4]

[1] School of Big Data and Computer Science, Guizhou Normal University, Guiyang 550025, China
[2] School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
[3] School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China; qiuhb@guet.edu.cn (H.Q.); zhanling@guet.edu.cn (L.Z.)
[4] School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou 310018, China; chenlili@hdu.edu.cn
* Correspondence: yv@gznu.edu.cn or yv@stu.xidian.edu.cn

Abstract: Polar codes have garnered significant attention since they were proposed due to their capacity-achieving properties and their low-complexity decoding process, and they were selected as the coding scheme for the control channel in 5G mobile communication technology. A successive cancellation list (SCL) decoder, aided by a cyclic redundancy check (CRC), performs the competitive error correcting performance for polar codes compared to other sophisticated codes. The SCL-Flip decoding algorithm is proposed to enhance the error correction performance of the SCL. However, the SCL-Flip is hampered by a high decoding complexity due to the necessary selection and sorting process of path metrics (PMs). Additionally, it requires a large number of iterations to achieve a good error correction performance. In this paper, we propose the error segment bit-flip (ES-SCL-Flip) decoding algorithm, a simplified bit-flip approach for the SCL decoder based on analyzing the distributions of the first error bit. Firstly, we construct an index set with a high probability of the first erroneous bit. Second, a bit-flip criterion is introduced to prevent the costly operation of the PM selection and sorting without compromising the error correction performance. Finally, a search strategy is employed that utilizes segmented error indication to target and rectify the initial error segment sequentially. The simulation results demonstrate that the ES-SCL-Flip decoder provides an improved error correction performance with only a minimal increase in decoding complexity compared to the conventional CRC-aided SCL decoder. This algorithm attains a commendable equilibrium between the error correction performance and decoding complexity.

Keywords: 5G; polar codes; successive cancellation list decoding algorithm; bit flip

MSC: 94B35

check for updates

Citation: Zhang, J.; Qiu, H.; Zhan, L.; Chen, L. An Error Segment Bit-Flip Algorithm for Successive Cancellation List Decoding of Polar Codes. *Mathematics* **2024**, *12*, 2072. https://doi.org/10.3390/math12132072

Academic Editor: Ioannis G. Tsoulos

Received: 22 April 2024
Revised: 9 June 2024
Accepted: 28 June 2024
Published: 2 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Polar codes, proposed by Arikan, are named after the concept of channel polarization, employ the low-complexity successive cancellation (SC) decoding method, and have been demonstrated to possess a symmetric channel capacity for infinite code length [1]. However, when it comes to practical implementation, the error rate performance advantage of polar codes with SC decoding is unattainable under conditions where the code length is constrained. This limitation arises from the phenomenon of incomplete channel polarization, which impedes the full realization of the decoding potential. Consequently, various approaches have been proposed to address this issue. The successive cancellation list (SCL) decoding [2,3] concurrently evaluates a maximum of L decoding paths at each decoding stage, culminating in the selection of a single codeword from the list as the final output. Building upon this, the cyclic redundancy check (CRC)-aided SCL (CA-SCL) decoding [4,5] utilizes outer codes to recognize the proper codeword from the list. This approach has shown a competitive performance compared to the LDPC and Turbo codes,

while maintaining a low complexity in both encoding and decoding [4]. Therefore, polar codes have been adopted as the control channel coding scheme for 5G enhanced Mobile Broadband (eMBB) scenarios [6].

Both the SC and SCL decoding algorithms feature a sequential decoding structure that resembles a complete binary tree traversal process starting from the leftmost branch, leading to a high decoding latency. Therefore, several efficient decoding strategies [7–11] have been introduced to curtail the decoding latency by identifying specific nodes as constituent codes within the traditional SC and SCL decoding frameworks. Notably, these mentioned nodes include Rate-0 and Rate-1 nodes in simplified SC (SSC) decoding [7], along with the identification of Repetition (REP) nodes, Single Parity Check (SPC) nodes [8], REP-SPC nodes [9], and other Merged (MEG) nodes [10,11].

Additionally, despite the competitive performance of SCL decoding, the computational and memory complexities are $\mathcal{O}(LN \log N)$ and $\mathcal{O}(LN)$, respectively, increasing linearly with the list size. Hence, as an alternative solution, the SC-flip decoding scheme is put forth to reduce the average computational complexity [12]. It utilizes the absolute values of log-likelihood ratios (LLRs) as a metric to evaluate the channel reliability, focusing on the precise identification and rectification of the single erroneous bit. This method exhibits a comparable performance to SCL decoding with $L = 2$ when the flip number $T = 32$, significantly reducing the computational cost by half at high Eb/N0 values and halving the memory complexity across all Eb/N0 values. Ref. [13] employed the SC decoding tree to create a *critical set* (CS) capable of accurately identifying the first error bit. The dynamic SC-flip (D-SCF) decoding algorithm, which considers the properties of SC sequential decoding to choose non-frozen bits for flipping, was introduced in [14]. To mitigate the complexity of the D-SCF algorithm, the author of Ref. [15] introduced a new criterion for selecting non-frozen bits into the flip set predicated on the bit error rate (BER), computed by the Gaussian approximation (GA) exceeding the estimated BER derived from the LLR in SC decoding. By using the criterion, it is possible to effectively decrease the decoding complexity while maintaining a comparable performance as the D-SCF decoding algorithm.

Inspired by the idea of flipping, the author of [16] attempted to improve the performance of the SCL algorithm by conducting bit-flipping re-decoding after the SCL decoding failure. This method, known as SCL-flip (SCLF) decoding, modifies the critical set by analyzing the candidate path state in SCL decoding. Numerical data show that in a medium to high SNR, SCLF decoding outperforms SCL decoding in terms of error correction. The shifted pruning algorithm, as mentioned in Ref. [17], involves shifting the pruning window by a factor of L, and selects the paths from position $p + 1$ to $L + p$ at the critical bit coordinate by calculating and sorting the *path metric* (*PM*). This method underwent enhancements in [18] through the utilization of a computationally straightforward LLR-based metric and a factor Q, which can be used to estimate the location for removing the proper path. Furthermore, an enhanced shifting pruning was introduced in [19], incorporating an ordered critical set into an adaptive list size scheme, which is a higher-order SCL-flip decoding that can flip ω candidates in a single extra decoding attempt. In addition, the dynamic SCL-flip decoding algorithm was modified for its use in SCL decoding as described in [20], dynamically adjusting the candidate flip locations within a fixed-size set, resulting in improved performance and reduced average complexity with a limited list size.

In this work, we propose a simplified SCL-flip decoding algorithm by analyzing the distributions of the first erroneous bit. We create an index set that has a high probability of containing the first erroneous bit and proposed a bit-flip criterion based on the distributions. This can increase the effectiveness of determining the bits that need to be flipped, thus avoiding the selection and sorting of PM values. Based on the bit-flip criterion, we propose an ES-SCL-Flip decoder that employs segmented error indication to target and rectify the bits in the initial error segment sequentially. The simulation results demonstrate that the proposed decoder with the proposed bit-flip criterion outperforms the CRC-aided SCL decoder with a negligible increase in decoding complexity.

The reminder of the paper is organized as follows. Section 2 offers the preliminaries on polar codes and SC, SCL, CA-SCL, SC-flip, and SCL-flip decoding. Section 3 provides a comprehensive description of the proposed simplified bit-flip SCL decoding algorithm. Section 4 discuss the numerical results, while the conclusions are derived in Section 5.

2. Preliminaries

2.1. Polar Codes

The concept of polar codes originates from channel polarization. To make polar codes, N separate physical channels are combined and split into N related virtual polarization channels. Then, K reliable bit-channels are chosen to send information bits (non-frozen bits) and CRC bits, while the remaining (N-K) bit-channels send the known bits (default is 0). The indices of reliable bit-channels are gathered in set $\mathcal{A}$, whereas set $\mathcal{A}^c$ contains the indices of the other (N-K) bit-channels. Polar encoding relies on the generator matrix $G_N = G_2^{\otimes n}$, where $G_2 \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and $(\cdot)^{\otimes n}$ denotes the n-th Kronecker power, with $n = \log_2 N$ [1]. In the following, we use a_i^j to express a vector $(a_i, a_{i+1}, \ldots, a_j), j > i$. A polar code can be encoded as $x_0^{N-1} = u_0^{N-1} G_N$, where u_0^{N-1} represents the encoding vector and it is separated from $\mathcal{A}$ and $\mathcal{A}^c$. x_0^{N-1} is the polar codeword.

2.2. SC, SCL, and CA-SCL Decoding

The initial decoding algorithm for polar codes is SC decoding, which can be better represented as a binary tree seeking that computes the bit-wise message beginning from the root node (level = n) and from the leftmost branch. At each tree level t, the logarithmic-likelihood ratios (LLRs) $\boldsymbol{\alpha} = \left\{\alpha_0, \alpha_1, \ldots, \alpha_{2^t-1}\right\}$ are obtained from its parent node, and are provided to the child nodes iteratively via the left and right functions. The LLRs of left child nodes $\boldsymbol{\alpha}^l = \left\{\alpha_0^l, \alpha_1^l, \ldots, \alpha_{2^{t-1}-1}^l\right\}$ and right child nodes $\boldsymbol{\alpha}^r = \left\{\alpha_0^r, \alpha_1^r, \ldots, \alpha_{2^{t-1}-1}^r\right\}$ are calculated as follows:

$$\alpha_j^l = \operatorname{sgn}(\alpha_j) \cdot \operatorname{sgn}(\alpha_{j+2^{t-1}}) \cdot \min\left(\left|\alpha_j\right|, \left|\alpha_{j+2^{t-1}}\right|\right) \tag{1}$$

$$\alpha_j^r = \alpha_{j+2^{t-1}} + (1 - 2\beta_j^l) \cdot \alpha_j \tag{2}$$

where $0 \le j < 2^{t-1}$. The partial sum $\boldsymbol{\beta}$ is computed through the messages from the left and right child nodes and delivered to the parent node. The computation of a partial sum β_j is as follows:

$$\beta_j = \begin{cases} \beta_j^l \oplus \beta_j^r, & if\ j \le 2^{t-1} \\ \beta_{j-2^{t-1}}^r, & otherwise \end{cases} \tag{3}$$

where $0 \le j < 2^t$. At the leaf nodes of the greedy search tree (level 0), the decoded bit $\hat{u}_i$ can be estimated as follows:

$$\hat{u}_i = \beta_0 = \begin{cases} 0, & if\ i \in \mathcal{A}^c\ or\ \alpha_0^i \ge 0 \\ 1, & otherwise \end{cases} \tag{4}$$

where $0 \le i < N$ and α_0^i represents the LLR of bit i at level 0.

SC decoding is a depth-first search algorithm, whereas SCL decoding is a breadth-first search algorithm that enables simultaneous searches along most L paths. During decoding the i-th non-frozen bit, it is essential to take account of both estimations 0 and 1. This requires expanding the list of L to $2L$ to accommodate all the candidate paths. For each path ℓ and each level t, the PM can be recursively computed as follows:

$$PM_\ell^t = \begin{cases} PM_\ell^{t-1} + \left|\alpha_0^i[\ell]\right|, & if\ \operatorname{sgn}(\alpha_0^i[\ell]) \ne (-1)^{\hat{u}_i} \\ PM_\ell^{t-1}, & otherwise \end{cases} \tag{5}$$

where $\alpha_0^i[\ell]$ denotes the LLR of bit i of ℓ path and PM is initialized as 0.

As shown in (5), the PM values take on a non-negative value, so a path with a smaller PM is considered more reliable, while the path with a less reliable bit is penalized by $|\alpha_0^i[\ell]|$. When decoding the m-th bit, the L paths with the lowest PM are selected and arranged in ascending order from PM_1^m to PM_L^m. Then, the path with the lowest PM_1^N is assigned as the estimated codeword.

In addition, the observations in [4] revealed that SCL decoding has the potential to select an incorrect codeword from the final candidate lists based on the PM evaluations. Therefore, a *cyclic redundancy check* (CRC) is implemented as an outer code for the non-frozen bits to assist the decoder in effectively detecting errors and determining the proper list among L candidates, resulting in a substantial enhancement in the error correction performance. If the CRC consists of r bits, the code rate of the polar code is $(K + r)/N$. In the following, we use PC $(N, K + r)$ to denote a polar code with code length N, non-frozen bits K, and CRC bits r.

2.3. SC-Flip and SCL-Flip Decoding

The SC decoder employs a sequential decoding strategy, utilizing the received soft channel LLR $L(\hat{y}_i)$ and previously estimated bits $\hat{u}_1^{i-1}$ to facilitate the estimation of the bits $\hat{u}_1^i$. When SC decoding fails in the first attempt detected by a CRC, the decoder proceeds to perform T consecutive SC re-decoding attempts. During each attempt, the decoder flips the hard decision bit corresponding to the least reliable bit and applies it to the subsequent portion of the SC decoding tree. Furthermore, enhanced SC-flip decoding algorithms are discussed in [13–15], which present modified bit-flip metrics to evaluate multiple erroneous hard decisions and rectify them, thereby improving the error correction performance. In [13], polar codes are conceptualized as an operational procedure conducted on a subblock by subblock basis. Within this framework, the accurate decisions of the first bit, which has a code rate of 1, are pivotal, directly influencing the overall decoding performance of the polar codes. Therefore, the indices of the first bit in the rate-1 subblocks are constructed as a CS. After that, a binary flipping strategy is employed to alter the decision bit associated with the CS in additional decoding attempts, facilitating successive decoding attempts. The re-decoding process terminates either upon a successful CRC or with exhaustion of the predetermined maximum number of flip attempts. Ref. [14] employs a generalized SC-flip decoder to correct more than one erroneous hard decision by nested flipping based on the flip metrics. In [15], non-frozen bit indices with estimated BERs from LLRs in SC decoding that exceed their expected BER are put into the flip set.

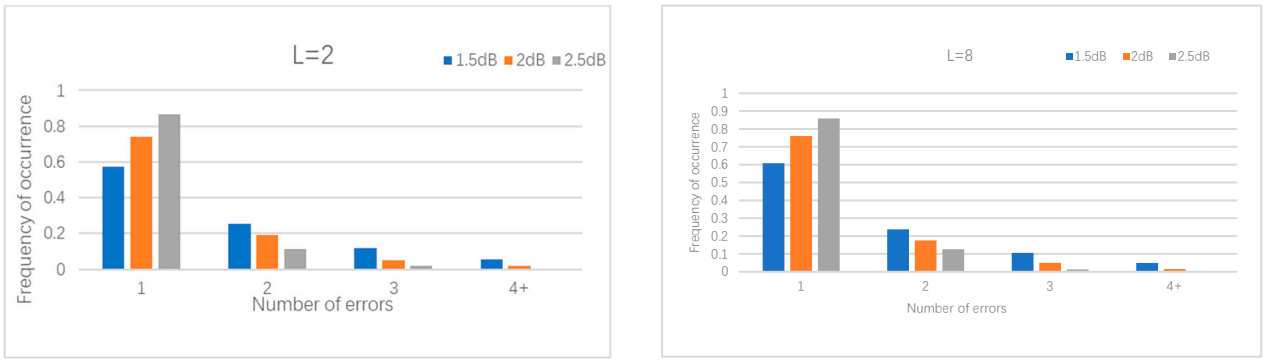
SCL-flip decoding, first introduced in [16], integrates the idea of bit flip into SCL, flipping the path selection numerous times if all paths fail the CRC in the original SCL decoding. The initially disregarded path, which expanded a single path in competition, is chosen as the surviving path for subsequent decoding. In [17], an improved flip scheme is suggested that takes into account L paths previously discarded as potential flip positions in the re-decoding attempt. The authors of [18–20] derive different flip metrics to determine the flip indices. It is demonstrated in [18] that, even if the exact elimination index does not reside within the top T candidate positions for shifting, the correct path can still be preserved if $\kappa < L$. In [19], an ordered critical set is constructed by sorting the mean of the Gaussian approximation LLR values. A larger mean LLR indicates that the bit is more reliable. In the event of an unsuccessful initial CA-SCL decoding, the L candidate paths with the lowest PM value are inverted if the error bit falls within the critical set; otherwise, the L candidate paths with the highest PM value are inverted. The procedure continues until the CRC succeeds or the maximum number of flips is reached. In [20], the author utilizes the dynamic flip set, which consists of T subsets and keeps track of all the flip indices for the t-th additional decoding attempt. Additionally, a distributed CRC is exploited to further enhance the performance of D-SCLF decoding.

3. A Simplified Bit-Flip Method for Successive Cancellation List Decoding

The flip metrics that are introduced in [18–20] are used to assess the priority of flipping positions, which rely on the probability of eliminating the proper decoding path at a specific bit location. Despite the metrics being reasonably improved with an increasing SNR, they need to be updated and sorted after each flip, which leads to significant computational and space complexity. In this study, we propose a simplified bit-flip method to improve the error correction performance of SCL-flip decoding and reduce the implementation complexity of SCL-flip decoders using a distributed CRC.

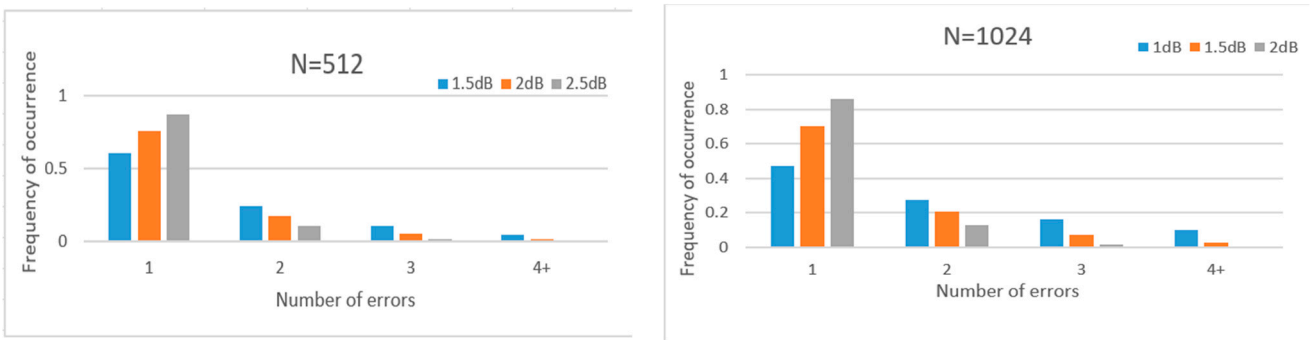
3.1. High-Probability Error Bit Index Selection

Either channel-induced noise corruption or error propagation from previously decoded bits causes the estimation error of CA-SCL decoding at each bit. The first erroneous error is always a result of channel noise, as there are no preceding mistakes, which then leads to the occurrence of subsequent estimation errors. Figure 1 displays the relative frequency of the errors caused by channel noise (given that there was at least one error) for three Eb/N0 values for polar code with $N = 512$ and $R = 1/2$ sent over an AWGN channel. It is evident that the channel typically contributes just one error most of the time, and this pattern gets increasingly more noticeable as the Eb/N0 values increase. Figure 2 displays the relative frequency of the errors resulting from channel noise with two distinct block lengths and a coding rate $R = 0.5$, with $L = 4$. It is evident that the occurrence rate of a single error event rises as the block lengths grow. Consequently, the error correction performance of the CA-SCL decoder can be greatly improved if the first error bit can be corrected.



(**a**) (**b**)

Figure 1. Relative frequency of occurrence for errors caused by channel noise for different Eb/N0 values, where $N = 512$ and $R = 1/2$: (**a**) $L = 2$; and (**b**) $L = 8$.



(**a**) (**b**)

Figure 2. Relative frequency of occurrence for errors caused by channel noise for different Eb/N0 values, where $L = 4$ and $R = 1/2$: (**a**) $N = 512$; and (**b**) $N = 1024$.

The SCL-flip decoding allows for utilizing the distribution of the first error bit to increase the effectiveness of the bit-flipping selection procedure. Table 1 displays the first error bit distribution with varying Eb/N0 (dB) values obtained from simulating 5×10^6 frames, where a polar code of (512, 256 + 16) was constructed under Gaussian approximation and transmitted over an AWGN channel. It is worth noting that, when the threshold is set to 0.001, the rate of the first error bit collected under different Eb/N0 (dB) values can exceed 99%, and almost all first error bits are detected. Figure 3 illustrates the cumulative probability of the first error bit for a code length of $N = 512$, a code rate of $R = 1/2$, and various Eb/N0 (dB) values. It is evident that at different Eb/N0 values, when the threshold is 0.001, the rate of the first error bit consistently approaches 100%. Based on this, we can define the set of high-probability error bits $\mathcal{FS}$ as follows:

$$\frac{P_{CASCL}(i)|_{i\in\mathcal{FS}}}{\sum_{i\in A} P_{CASCL}(i)} \geq 0.001 \tag{6}$$

where $P_{CASCL}(i) \triangleq P(\hat{u}_i \neq u_i | \hat{u}_1^{i-1} = u_1^{i-1}, y_1^N)$ is the probability that the i-th bit is the first error bit after decoding.

Table 1. Distribution of first bit error with different $E_b/N_0(dB)$ values.

E_b/N_0 (dB)	Threshold	First Error Bit Distribution (The Interval Length Is 64 Bits)							
		1–64	**65–128**	**129–192**	**193–256**	**257–320**	**321–384**	**385–448**	**449–512**
1	>1%	1	6	6	4	4	1	0	0
	>0.5%	1	7	6	6	6	3	1	0
	>0.1%	1	7	10	11	10	7	5	0
2	>1%	1	5	5	5	5	3	3	0
	>0.5%	1	6	6	6	6	5	4	0
	>0.1%	1	7	9	10	9	10	8	0
3	>1%	1	4	4	5	4	4	3	0
	>0.5%	1	6	5	7	5	6	4	0
	>0.1%	1	6	8	8	8	10	9	0

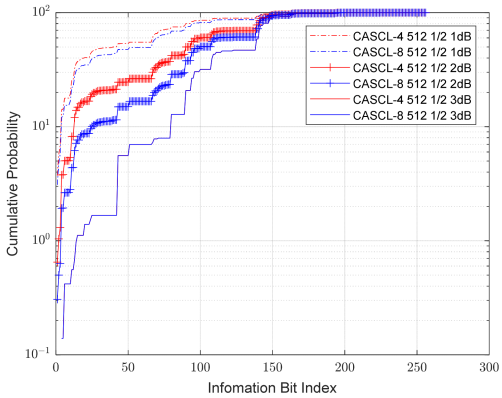


Figure 3. The cumulative error probability of the first error bit.

The channel polarization theory states that, even though the polarization is not complete, many bits are still transmitted in relatively reliable channels, which therefore have significantly lower error probabilities compared to other non-frozen bits. Thus, the selection process of a set with a high bit error rate involves excluding more dependable non-frozen bits from the set. When considering a set with a high bit error rate for flipping, it can actually reduce its size with a minimal impact on the accuracy of the flipping process.

3.2. Bit-Flip Criterion

When SCL decoding occurs at the i-th decoding step in the initial CA-SCL decoding, for any $j \in \mathcal{A}$, the input path list of L candidates is denoted as $\mathcal{L}^{j-1}$. Then, the SCL decoder evaluates the potential values of u_j as either 0 or 1 for every candidate. Only the L most dependable candidates evaluated by their PM can be reserved in a new list $\mathcal{L}^j$ for the j-th decoding step. Given flawless CRC detection, the SCL decoding is expected to be successful if the input sequence u_1^N is among the final candidates, i.e., $u_1^N \in \mathcal{L}^N$.

The event of the first hard decision error at the non-frozen bit u_i, $\forall\ i \in \mathcal{FS}$ can be defined as E_i, which implies the correct decoding trajectory is removed from the reserved candidate, i.e., $\hat{u}_1^{i-1} \in \mathcal{L}^{i-1}$ and $\hat{u}_1^i \notin \mathcal{L}^i$. Thus, for any $j \in \mathcal{A}$, let $\mathcal{L}^j[i]_0 = \left\{ \overset{\wedge}{\mathbf{u}}_1^j[i]_0,\ \overset{\wedge}{\mathbf{u}}_2^j[i]_0, \ldots, \overset{\wedge}{\mathbf{u}}_L^j[i]_0 \right\}$ represent the list of paths for $u_j = 0$, and $\mathcal{L}^j[i]_1 = \left\{ \overset{\wedge}{\mathbf{u}}_1^j[i]_1, \overset{\wedge}{\mathbf{u}}_2^j[i]_1, \ldots, \overset{\wedge}{\mathbf{u}}_L^j[i]_1 \right\}$ denote the list of paths for $u_j = 1$ at the j-th decoding step. Furthermore, let the PM of $\overset{\wedge}{\mathbf{u}}_l^j[i]_m$ be denoted as $\mathrm{PM}_l^j[i]_m$, with $m \in \{0,1\}$ and $l \in \{1,2,\ldots,L\}$. Furthermore, $\hat{u}_l^j[i]_{1,m}^k = \left\{ \hat{u}_l^j[i]_{1,m},\ \hat{u}_l^j[i]_{2,m}, \ldots, \hat{u}_l^j[i]_{k,m} \right\}$ is the vector consisting of the initial k bits of the l-th decoding path at the j-th decoding step, whose PM is stated in $\mathrm{PM}_l^j[i]_{k,m}$. That implies $\overset{\wedge}{\mathbf{u}}_l^j[i]_m \in \mathcal{L}^j[i]_m$. Note that if $j < k$, $\hat{u}_l^j[i]_{1,m}^k$ are null vectors, $\mathrm{PM}_l^j[i]_{k,m}$ can be calculated as follows [21]:

$$\mathrm{PM}_l^j[i]_{k,m} = \sum_{j=1}^{j} \ln\left(1 + e^{-(1-2\hat{u}_l^j[i]_{k,m})\cdot\alpha_l^j[i]_{k,m}}\right) \tag{7}$$

where $\alpha_l^j[i]_{k,m}$ is the decoding LLR of bit $\hat{u}_l^j[i]_{1,m}^k$, $m \in \{0,1\}$ and given by y_1^N and decoding trajectory $\hat{u}_l^j[i]_{1,m}^{k-1}$:

$$\alpha_l^j[i]_{k,m} = \log\left(\frac{\Pr\left(u_l^j[i]_{k,m} = 0 \middle| y_1^N, u_l^j[i]_{1,m}^{k-1}\right)}{\Pr\left(u_l^j[i]_{k,m} = 1 \middle| y_1^N, u_l^j[i]_{1,m}^{k-1}\right)}\right) \tag{8}$$

Therefore, for any u_i ($\forall\ i \in \mathcal{FS}$), we propose a bit-flip criterion for the SCL decoder that if E_i occurs, all candidates in $\mathcal{L}^i$ should be selected as the paths that correspond to $\overline{u}_i$ in a subsequent re-decoding attempt, where $\overline{u}_i$ is the flip of u_i. This bit-flip criterion ensures that if E_i occurs in the initial CA-SCL decoding, the proper path is preserved in $\mathcal{L}^i$, i.e.,

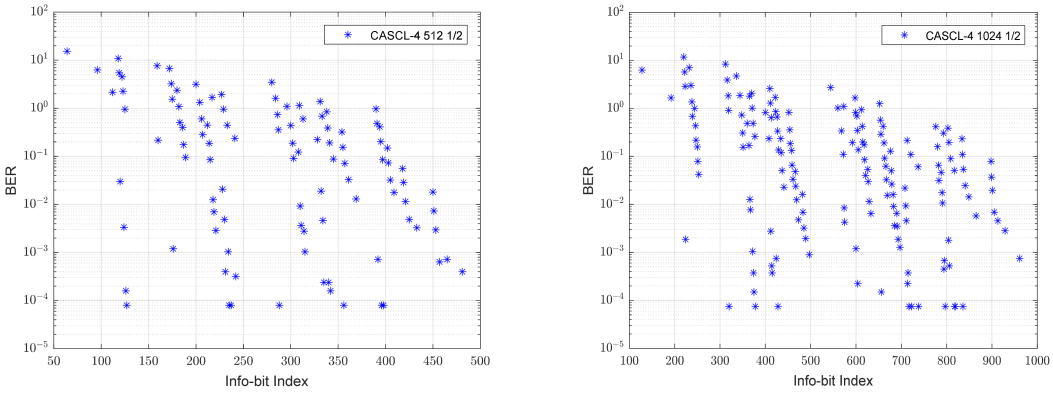
$$\begin{cases} \mathcal{L}^i = \mathcal{L}^j[i]_1,\ \mathrm{PM}_l^j[i]_m = PM_l^j[i]_1\ if\ \hat{u}_i = 0 \\ \mathcal{L}^i = \mathcal{L}^j[i]_0,\ \mathrm{PM}_l^j[i]_m = \mathrm{PM}_l^j[i]_0 \quad if \quad \hat{u}_i = 1 \end{cases} \tag{9}$$

$$\begin{cases} \alpha_l^j[i]_{1,m}^k = \left\{\alpha_l^j[i]_{1,0}, \alpha_l^j[i]_{2,0}, \ldots, \alpha_l^j[i]_{k,0}\right\} \quad if \quad \hat{u}_i = 0 \\ \alpha_l^j[i]_{1,m}^k = \left\{\alpha_l^j[i]_{1,1}, \alpha_l^j[i]_{2,1}, \ldots, \alpha_l^j[i]_{k,1}\right\}\ if\ \hat{u}_i = 1 \end{cases} \tag{10}$$

3.3. Codeword Segmentation

Table 1 indicates that there is only one first error bit in the first interval of the 64-bit spaced message sequence, while the last interval has no first error bit, and a nearly equal number of first error bits are observed in the middle passages. This aids in reducing the flipping time in re-decoding attempts by promptly identifying the first error bit segment. As a result, we employ the distributed CRC bits used in SC-flip [21] to determine the erroneous segments containing the first error bit. However, unlike the research in [21], the distributed CRC bits discussed in this research solely function as a parity check and are not utilized for pruning during the decoding process.

In this case, the distributed positioning of CRC bits and the precise segmentation of the codewords play a crucial role in the decoding process, greatly decreasing the average number of iterations executed and improving the error correction performance. Figure 4 illustrates the bit error rate (BER) of the first erroneous bit under CA-SCL decoding for the polar code (512, 256 + 16) and (1024, 512 + 16), respectively. As shown in the picture, the BER of the first erroneous bit exhibits a segmented pattern characterized by a gradual decrease from high to low, followed by a transition to the next segment. Hence, by placing the CRC bits on the most reliable bits within each message segment, errors inside the message segment can be detected.



(**a**) (**b**)

Figure 4. Bit error rate of first erroneous bit under conventional CA-SCL decoding: (**a**) a polar code of (512, 256 + 16); and (**b**) a polar code of (1024, 512 + 16).

Consider the example of a polar code (512, 256 + 16). Assuming that the total length of CRC bits used in the encoding is 16, we can divide the message sequence into five segments with indices of 1~128, 129~192, 193~256, 257~336, and 337~512, respectively, based on Figure 4 and Table 1. The 57 information bits in $\mathcal{FS}$ are identified according to Equation (6) and are also distributed across these five segments, with each segment containing 8, 10, 12, 12, and 15, respectively. For each of the first four segments, the CRC-2 is generated for the non-frozen sequence in the $\mathcal{FS}$ of that segment and placed at the end of the respective segment. Subsequently, the message sequences and parity bits of the first four segments are combined with the sequence of the last segment to generate the CRC parity bits for all sequences, totaling 8 bits. Finally, these are fed into the polar encoder for encoding. Figure 5 illustrates the coding framework for the first segment, and the same operations applied to the rest of the segments. The blue boxes represent the information bits in $\mathcal{FS}$; the yellow boxes represent the CRC bits; the green boxes represent the non-frozen bits that are not part of $\mathcal{FS}$; and the other boxes represent the frozen bits.

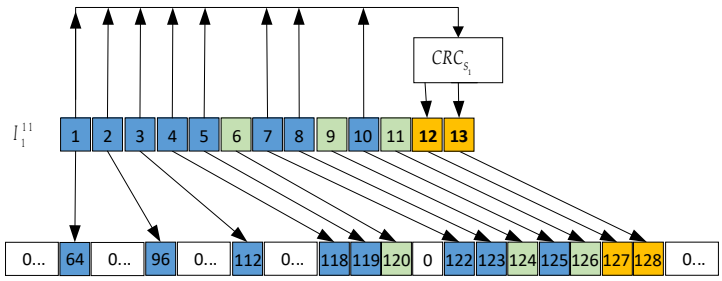


Figure 5. Polar coding diagram based on first error bit distribution.

3.4. Proposed SCL-Flip Decoding Algorithm

In the coding framework described above, the proposed SCL-Flip decoder based on error segment (ES-SCL-Flip) is summarized in Algorithm 1. In Algorithm 1, we refer to the

initial CA-SCL decoding as SCL(0) and the final output path with the lowest *PM* as $\hat{u}_1^N[0]$. An additional decoding attempt corresponding to flip is SCL-Flip(t). The indices of the last CRC bit in each of the first four segments are denoted as *stop_index*. The main decoding operation procedure is as follows. If the initial CA-SCL decoding fails, the output is $\hat{u}_1^N[0]$ while constructing the flip set and the maximum flip number T with Algorithm 2. The flip set is the subset of $\mathcal{FS}$, and the SCL-Flip(t) is performed for $\mathcal{L}^i$, with $i \in$ FlipIndexSet, $t \in \{1, 2, \ldots, |\text{FlipIndexSet}|\}$, where |FlipIndexSet| represents the number of elements in the FlipIndexSet. If at least one path passes the distributed CRC after completing the extra decoding of each segment, the decoder proceeds to the following segments. However, if a segment does not contain any path that passes the distributed CRC, the decoder stops this decoding and proceeds to the next round of re-decoding, known as an Early Stop Mechanism, and follows the above process either until at least one path $\mathcal{L}^i$ in all segments passes all CRC bits, the decoding is successful, and the final path is the output, or the maximum number of extra decoding attempts is reached. Once the maximum number of flips has been reached and no path is found where all the CRC bits pass successfully, the decoding process is considered a failure.

Algorithm 1: Proposed SCL-Flip Decoding

Input: $y_1^N, \mathcal{A}, L, CRC_1^S, stop_index$
Output: $\hat{u}_1^N$
$\hat{u}_1^N \leftarrow \text{CA} - \text{SCL}(y_1^N, \mathcal{A})$
if $CRC_1^S(\hat{u}_1^N)$ pass **then**
Return $\hat{u}_1^N$
else
Return $\hat{u}_1^N[0]$, **Algorithm 2 do**
for $t = 1 \rightarrow T$ **do**
for $i = 1 \rightarrow N$ **do**
$\hat{u}_i \leftarrow \text{SCL}(y_1^N, \mathcal{A}, \textbf{FlipIndexSet})$
if $i =$ **FlipIndexSet**(t)
$\mathcal{L}^i \leftarrow$ *select the paths with* $\hat{u}_l^i[t] = 1 - \hat{u}^i[0]$
else
if $i \in stop_index$ and $CRC_1^{s-1}(\hat{u}_1^i)$ fails
Terminate
end
end
end
if $CRC_1^s(\hat{u}_1^N)$ pass **then**
Return $\hat{u}_1^N$
end
end
end

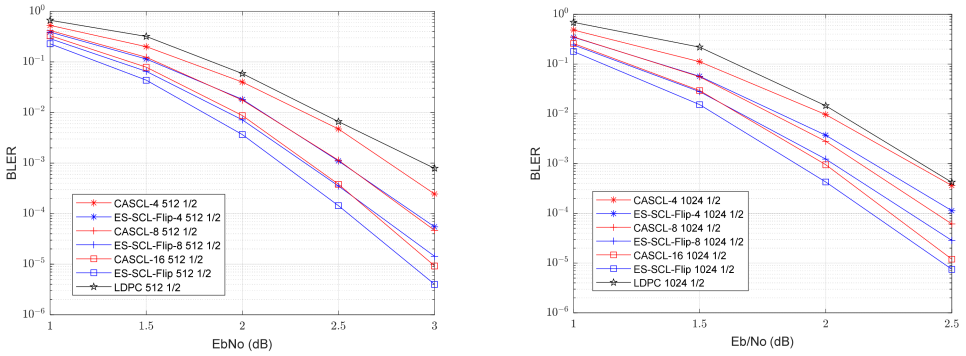
Algorithm 2: FlipIndexSet

Input: $\hat{u}_1^N[0]$, S
Output: FlipIndexSet, T
for $s = 1 \rightarrow S - 1$ **do**
if $CRC(\left\{\hat{u}_j[0], j \in \mathcal{FS}_s\right\})$ **fails then**
FlipIndexSet $\leftarrow \mathcal{FS}_s$
Break
else
if $s = S - 1$ **then**
FlipIndexSet $\leftarrow \mathcal{FS}_S$
end
end
end
Return FlipIndexSet and $T =$ |FlipIndexSet|

4. Simulation Results and Complexity Analysis

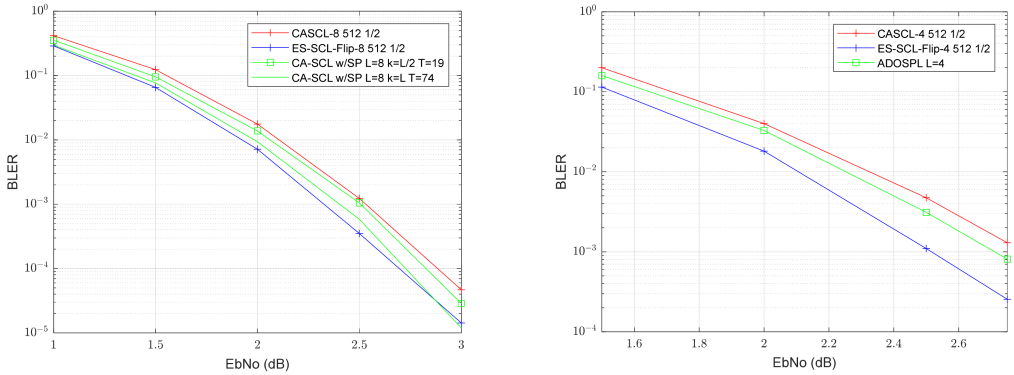
4.1. Simulation Results and Numerical Analysis

To verify the error correction performance of the ES-SCL-Flip decoding, we compare the block error rate (*BLER*) of the proposed ES-SCL-Flip decoder to the CA-SCL, LDPC, and existing shifted pruning (SP) list decoders [18,19] in Figures 6 and 7. Since the maximum number of extra decoding attempts by the ES-SCL-Flip decoder is equivalent to the maximum number of elements in the segmented $\mathcal{FS}$, we set the maximum number of flips in the simulation to the same value for fairness. For the ES-SCL-Flip decoder, the CRC-2 polynomial in the first four segments is $p(x) = x^2 + 1$, and the CRC-8 polynomial in the last segment is $p(x) = x^8 + +x^6 + x^3 + x^2 + 1$. The CRC-16 polynomial in the CA-SCL decoder and the other two SP decoders is $p(x) = x^{16} + x^{15} + x^2 + 1$.



(**a**) (**b**)

Figure 6. The *BLER* performance of different decoders: (**a**) performance comparison of PC(512, 256 + 16) for different decoders with L = {4, 8, 16} and the counterpart in the LDPC; and (**b**) performance comparison of PC(1024, 512 + 16) for different decoders with L = {4, 8, 16} and the counterpart in the LDPC.



(**a**) (**b**)

Figure 7. The *BLER* performance of different decoders: (**a**) performance comparison of PC(512, 256 + 16) for different decoders with L = 4; and (**b**) performance comparison of PC(1024, 512 + 16) for different decoders with L = 4.

As observed in Figure 6, the *BLER*s of both decoders decrease as L increases. Under the same list size, the ES-SCL-Flip decoder shows obvious advantages in error correction performance compared to the CA-SCL. For polar code (512, 256 + 16) at $BLER = 10^{-3}$, the ES-SCL-Flip decoding algorithm shows 0.3 dB improvement compared to the CA-SCL when L = 4. At the same time, when L = 4, the *BLER* curve of the ES-SCL-Flip decoder closely resembles that of the CA-SCL of L = 8, and the ES-SCL-Flip decoder with L = 8 nearly matches the CA-SCL with L = 16. For polar code (1024, 512 + 16) at $BLER = 10^{-5}$, the ES-SCL-Flip decoding algorithm improves by 0.2 dB compared with the CA-SCL when L =

16. When $E_b/N_0 \leq 1.5$ dB, the *BLER* curveofthe ES-SCL-Flip decoder with $L = 4$ is almost the same as that of the CA-SCL with $L = 8$. On the other hand, when $E_b/N_0 > 1.5$ dB, the *BLER* curve of the ES-SCL-Flip decoder with $L = 4$ is slightly worse than that of the CA-SCL with $L = 8$. Additionally, when $E_b/N_0 \leq 1.5$ dB, the *BLER* performance of the ES-SCL-Flip decoder with $L = 8$ is nearly equivalent to that of the CA-SCL with $L = 16$, whereas when $E_b/N_0 > 1.5$ dB, the differences in the *BLER* performance between the ES-SCL-Flip decoder with $L = 8$ and the CA-SCL decoder with $L = 16$ increase with the increase in the E_b/N_0 values.

In Figure 7a, it can be observed that when list size $L = 8$, the ES-SCL-Flip decoder outperforms the CA-SCL-SP decoder. But when $E_b/N_0 = 3$ dB , the performance gain of the ES-SCL-Flip decoder appears less remarkable than that of CA-SCL-SP decoder with $T = 74$. This is because the ES-SCL-Flip decoder is limited to a maximum of 15 additional decoding attempts, whereas the CA-SCL-SP decoder benefits from a substantially higher number of shifted attempts, reaching up to 74, thereby allowing for more decoding attempts. As observed in Figure 7b, when $L = 4$, the ES-SCL-Flip decoder shows obvious advantages in the error correction performance compared to the ADOSPL [19]. For polar code (512, 256 + 16) at $BLER = 10^{-3}$, the ES-SCL-Flip decoding algorithm shows a 0.25 dB improvement compared to the CA-SCL when $L = 4$.

4.2. Complexity Analysis

The average computational complexity of the ES-SCL-Flip decoding algorithm primarily consists of the FlipIndexSet determination complexity and the additional decoding complexity. Since the complexity of the FlipIndexSet determination is an off-line operation, this paper focuses on the additional decoding complexity, which can be expressed as $T_{average} \cdot \mathcal{O}(LN \log N)$, where $T_{average}$ is the average number of re-decoding attempts. In the decoding process, the maximum number of flips is basically determined by Algorithm 1, which is no more than 15. Hence, $T_{average} \leq 15$.

Furthermore, assuming the average complexity of the SC decoding algorithm for polar codes when $N = 512$ and $R = 1/2$ is 1, Figure 8 shows the average normalized complexity of the ES-SCL-Flip decoder for $L = \{4, 8, 16\}$. It can be observed that the ES-SCL-Flip decoder exhibits significant computational complexity at a low SNR. As the SNR increases, the normalized complexity of the ES-SCL-Flip decoder rapidly converges to that of the CA-SCL decoder, which is equal to the number of lists.

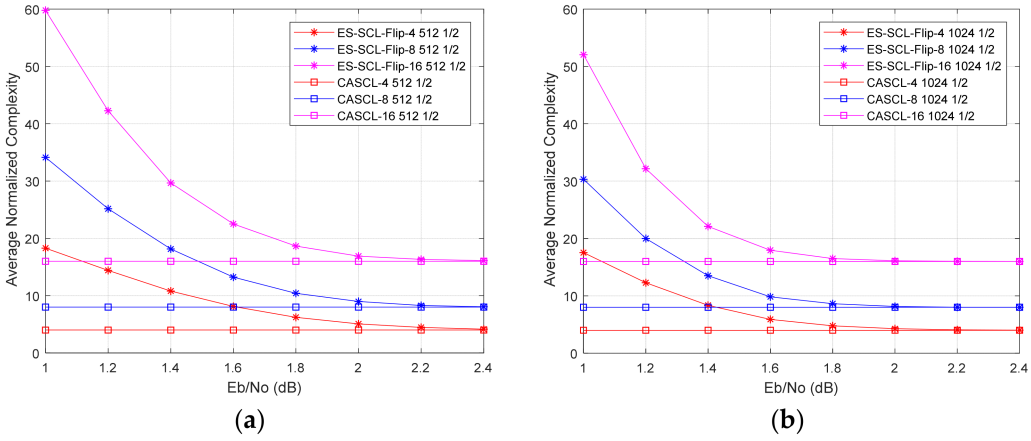


Figure 8. Average normalized computational complexity of the ES-SCL-Flip decoder of polar codes with $L = \{4, 8, 16\}$ and different code lengths: (**a**) average normalized computational complexity of PC(512, 256 + 16); and (**b**) average normalized computational complexity of PC(1024, 512 + 16).

4.3. Decoding Delay

The delay in the ES-SCL-Flip decoding depends on a series of additional decoding attempts and the average number of re-decoding attempts is $T_{average}$, which are performed

in case the CA-SCL decoder fails. As a result, the average decoding delay of the ES-SCL-Flip decoding is a factor of the average normalized complexity. At high E_b/N_0 values, the average decoding latency for both the CA-SCL and the proposed ES-SCL-Flip decoder is nearly identical. This indicates that the proposed ES-SCL-Flip decoder can enhance the performance of the CA-SCL decoder without introducing any additional decoding complexity.

5. Conclusions

In this paper, we investigated a simplified bit-flip SCL decoding algorithm aimed at enhancing the performance and reducing the computational complexity of the CA-SCL decoder. Initially, we focused on the probability distribution of the first error bit that occurs in CA-SCL decoding. Based on the analysis, we provided the bit-flip criterion for the bit-flip SCL decoder. Finally, we presented the polar coding and related ES-SCL-Flip decoding algorithm. The simulation results demonstrated that the proposed ES-SCL-Flip decoder can improve the error correction performance of the CA-SCL decoder with minimal additional decoding complexity, particularly at high E_b/N_0 values.

Author Contributions: Conceptualization, J.Z. and H.Q.; methodology, J.Z. and L.C.; software, J.Z. and L.Z.; validation, L.Z. and L.C.; and writing—original draft, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SC	Successive cancellation decoding
SCL	Successive cancellation list decoding
SCL-Flip	Successive cancellation list flip decoding
CRC	Cyclic redundancy check
CA-SCL	CRC-aided successive cancellation list decoding
PM	Path metric
LLR	Log-likelihood ratio
$\mathcal{A}$	Information bit indices
$\mathcal{A}^c$	Frozen bit indices
BLER	Block error rate

References

1. Arikan, E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073. [CrossRef]
2. Tal, I.; Vardy, A. List Decoding of Polar Codes. In Proceedings of the 2011 IEEE International Symposium on Information Theory, St. Petersburg, Russia, 31 July–5 August 2011; pp. 1–5.
3. Chen, K.; Niu, K.; Lin, J.R. List successive cancellation decoding of polar codes. *Electron. Lett.* **2012**, *48*, 500–501. [CrossRef]
4. Niu, K.; Chen, K. CRC-Aided Decoding of Polar Codes. *IEEE Commun. Lett.* **2012**, *16*, 1668–1671. [CrossRef]
5. Tal, I.; Vardy, A. List Decoding of Polar Codes. *IEEE Trans. Inf. Theory* **2015**, *61*, 2213–2226. [CrossRef]
6. *Evaluation on Channel Coding Candidates for eMBB Control Channel*; 3GPP TSG RAN WG1 87, R1–1611109; ZTE Microelectron: Reno, NV, USA, 2016.
7. Alamdar-Yazdi, A.; Kschischang, F.R. A Simplified Successive-Cancellation Decoder for Polar Codes. *IEEE Commun. Lett.* **2011**, *15*, 1378–1380. [CrossRef]
8. Sarkis, G.; Giard, P.; Vardy, A.; Thibeault, C.; Gross, W.J. Fast Polar Decoders: Algorithm and Implementation. *IEEE J. Sel. Areas Commun.* **2014**, *32*, 946–957. [CrossRef]
9. Sarkis, G.; Tal, I.; Giard, P.; Vardy, A.; Thibeault, C.; Gross, W.J. Flexible and Low-Complexity Encoding and Decoding of Systematic Polar Codes. *IEEE Trans. Commun.* **2016**, *64*, 2732–2745. [CrossRef]

10. Hanif, M.; Ardakani, M. Fast Successive-Cancellation Decoding of Polar Codes: Identification and Decoding of New Nodes. *IEEE Commun. Lett.* **2017**, *21*, 2360–2363. [CrossRef]
11. Condo, C.; Bioglio, V.; Land, I. Generalized Fast Decoding of Polar Codes. In Proceedings of the IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 1–6.
12. Afisiadis, O.; Balatsoukas-Stimming, A.; Burg, A. A low-complexity improved successive cancellation decoder for polar codes. In Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2–5 November 2014; pp. 2116–2120.
13. Zhang, Z.; Qin, K.; Zhang, L.; Zhang, H.; Chen, G.T. Progressive Bit-Flipping Decoding of Polar Codes over Layered Critical Sets. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
14. Chandesris, L.; Savin, V.; Declercq, D. Dynamic-SCFlip Decoding of Polar Codes. *IEEE Trans. Commun.* **2018**, *66*, 2333–2345. [CrossRef]
15. Zhang, X.; Liu, Y.; Chen, S. BER Evaluation Based SCFlip Algorithm for Polar Codes Decoding. *IEEE Access* **2020**, *8*, 3042–3054. [CrossRef]
16. Yongrun, Y.; Zhiwen, P.; Nan, L.; Xiaohu, Y. Successive Cancellation List Bit-flip Decoder for Polar Codes. In Proceedings of the 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 18–20 October 2018; pp. 1–6.
17. Rowshan, M.; Viterbo, E. Improved List Decoding of Polar Codes by Shifted-pruning. In Proceedings of the 2019 IEEE Information Theory Workshop (ITW), Visby, Gotland, Sweden, 25–28 August 2019; pp. 1–5.
18. Rowshan, M.; Viterbo, E. Shifted pruning for path recovery in list decoding of polar codes. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 27–30 January 2021; pp. 1179–1184.
19. Lv, Y.; Yin, H.; Wang, Y. An Adaptive Ordered Shifted-Pruning List Decoder for Polar Codes. *IEEE Access* **2020**, *8*, 225181–225190. [CrossRef]
20. Shen, Y.; Balatsoukas-Stimming, A.; You, X.; Zhang, C.; Burg, A.P. Dynamic SCL Decoder With Path-Flipping for 5G Polar Codes. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 391–395. [CrossRef]
21. Condo, C.; Ercan, F.; Gross, W.J. Improved successive cancellation flip decoding of polar codes based on error distribution. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; pp. 19–24.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.