Dynamic SCL Decoder with Path-Flipping for 5G Polar Codes

Dynamic SCL Decoder With Path-Flipping for 5G Polar Codes

Yifei Shen, Alexios Balatsoukas-Stimming, *Member, IEEE*, Xiaohu You, *Fellow, IEEE*, Chuan Zhang, *Senior Member, IEEE*, and Andreas Peter Burg, *Senior Member, IEEE*

***Abstract*—Since polar codes were ratified as part of the 5G standard, low-complexity polar decoders with close-to-optimum error-rate performance have received significant attention. Compared to successive cancellation (SC) decoding, both SC list and SC flip decoding can improve the error rate performance by increasing the number of considered candidate solutions. The combination of both strategies leads to SC list flip (SCLF) decoding, which can provide a tradeoff between error rate performance and area as well as energy. In this letter, we derive a new flip metric for the SCLF decoding process, based on which we propose the dynamic SCLF (D-SCLF) decoding algorithm. Moreover, we exploit the distributed CRC defined in the 5G standard to further optimize the D-SCLF decoding. Numerical results show that for the downlink control channel, our D-SCLF decoder with a list size of only four and only three additional attempts can achieve the performance of a regular list decoder with a list size of eight, leading to an overall memory and average complexity (energy) reduction.**

***Index Terms*—Polar codes, successive cancellation list flip (SCLF) decoding, path-flipping, 5G, check-remove.**

I. Introduction

FOLLOWING the invention of polar codes and their low-complexity successive cancellation (SC) decoding algorithm [1], Tal and Vardy proposed the SC list (SCL) decoding algorithm [2], which improves the error-correcting performance by keeping a list of up to L candidate bit sequences. When a cyclic redundancy check (CRC) code is concatenated with a polar code, this CRC can help the SCL decoder to select the candidate bit sequence from the list [3], which allows CRC-aided polar codes to reach a comparable performance to low-density parity-check codes and Turbo codes [4]. In 2016, polar codes were ratified as part of the 5G enhanced mobile broadband (eMBB) standard [5]. Specifically, they are adopted to protect the payload of physical uplink control channels (PUCCHs), physical downlink control channels (PDCCHs), and physical broadcast channels (PBCHs) [6].

Manuscript received September 8, 2021; revised November 9, 2021; accepted November 10, 2021. Date of publication November 22, 2021; date of current version February 17, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB2205503; in part by NSFC under Grant 62122020 and Grant 61871115; in part by Chinese Scholarship Council; in part by Huawei Technologies Company, Ltd.; in part by IEEE Circuits and Systems Society Pre-Doctoral Grants; and in part by the Jiangsu Provincial NSF under Grant BK20211512. The associate editor coordinating the review of this article and approving it for publication was L. P. Natarajan. *(Corresponding authors: Chuan Zhang; Andreas Peter Burg.)*

Yifei Shen, Xiaohu You, and Chuan Zhang are with the LEADS and the National Mobile Communications Research Laboratory, Southeast University, Nanjing 211111, China, and also with Purple Mountain Laboratories, Nanjing 210096, China (e-mail: chzhang@seu.edu.cn).

Alexios Balatsoukas-Stimming is with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands.

Andreas Peter Burg is with the Telecommunications Circuits Laboratory, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: andreas.burg@epfl.ch).

Digital Object Identifier 10.1109/LWC.2021.3129470

The error-correcting performance of the SCL decoder improves as the list size L increases, but at the cost of an increase in memory and computational complexity (energy). Although there are a series of breakthrough algorithms to reduce complexity, such as log-likelihood ratio (LLR)-based SCL decoding [7], fast simplified SCL decoding [8], and the list-pruning algorithm [9], it is still challenging to meet the low-latency and high-throughput requirements of eMBB scenarios with low area.

The SC flip (SCF) decoding algorithm [10] provides a sequential candidate exploration alternative to the parallel exploration strategy of list decoders. By guessing and flipping error-prone bits in multiple decoding attempts, state-of-the-art SCF decoders [11] achieve an area that is comparable to that of SC decoders with low average energy, but with very low worst-case throughput due to a large number of attempts.

Both SCL and SCF decoding improve error rate performance by enlarging the number of candidate bit sequences they consider either in parallel or sequentially. Combining the two algorithms therefore can provide new and possibly better tradeoffs between time- and area-complexity for a given frame error rate (FER) performance target. Such a combined SCL flip (SCLF) algorithm was first described in [12], and an optimized flip strategy was proposed by [13] on which all subsequent SCLF decoders are based. This flip strategy considers the L (out of $2L$) path candidates that were not chosen in the original SCL decoding, based on the assumption that the original choice had accidentally eliminated the correct path from the list. This approach is equivalent to the shifted-pruning strategy in [14] for a shift value of L. The performance improvement of the SCLF decoding depends on how accurately the bit index for which the first error occurs is located. The works in [12] and [14] employ a fixed *critical set* that contains the bit indices at which the correct path is potentially eliminated, which requires a large number of decoding attempts. In practice, a *flip metric* is calculated for each bit to measure the probability that the first erroneous path selection occurs. The state-of-the-art flip metric is proposed in [15], [16] with a heuristic derivation. Based on this metric, a higher-order SCLF decoding [15] that supports multiple flip locations is further proposed, where the decoding attempts of the ω-th order (i.e., with nested ω flip locations in each attempt) are activated only after all attempts with lower orders fail the CRC. Therefore, the number of decoding attempts increases rapidly with the flip order.

In this letter, we propose a novel dynamic SCLF (D-SCLF) decoding algorithm, with the following contributions:

1) We derive the probability of the first error in each bit location by considering the effects of both frozen bits and error propagation, which results in a novel and rigorous flip metric.
2) We propose the D-SCLF decoding that dynamically adjusts the candidate flip locations in a flip set of constant size, so that each decoding attempt is always associated with the smallest flip metric among the remaining

metrics. For polar codes with distributed CRC, the proposed D-SCLF decoding can prune invalid paths during the path selection, which leads to both performance improvement and average complexity reduction.

II. Background

A. 5G Polar Codes

An (E, A) polar code represents a code whose transmission length and information length are E and A, respectively. Generally, a CRC sequence $\boldsymbol{p}$ is appended to the tail of the information sequence $\boldsymbol{a}$, thereby forming an unfrozen bit sequence $\boldsymbol{c} = [\boldsymbol{a}, \boldsymbol{p}]$. The set of unfrozen bit indices is denoted by $\mathcal{A}$. To match the recursive encoding process, the length of the bit sequence $\boldsymbol{u}$ that participates in the encoding always has a value that is a power of two, which is referred to as mother code length and denoted by N. The coded bit sequence $\boldsymbol{x}$ is calculated as $\boldsymbol{x} = \boldsymbol{u}\mathbf{G}$, where $\mathbf{G} = \mathbf{F}^{\otimes n}$, $\mathbf{F} = \left[\begin{smallmatrix}1 & 0\\ 1 & 1\end{smallmatrix}\right]$, and $n = \log_2 N$. The sequence $\boldsymbol{x}$ will then pass the sub-block interleaver and rate-matching buffer, resulting in a length-E code word ready to transmit. Depending on the physical channel, there are some other modules to re-arrange the input or coded bits in the encoder and vice versa in the decoder. For example, for PDCCH and PBCH, the input bits interleaver is placed after the CRC encoder to distribute CRC bits.

B. Successive Cancellation List Decoding

Compared with SC decoding which only selects the locally optimal bit value, SCL decoding can keep L candidate paths (candidate bit sequences) at the same time to finally select the most reliable path for which the CRC is satisfied. In SCL decoding, the reliability of each path is measured by the path metric (PM), which can be calculated as [7]

$$\mathrm{PM}_l^{(i)} \approx \begin{cases} \mathrm{PM}_l^{(i-1)}, & \text{if } \hat{u}_l^{(i)} = \texttt{hard\_decision}(\lambda_l^{(i)}),\\ \mathrm{PM}_l^{(i-1)} + |\lambda_l^{(i)}|, & \text{otherwise,}\end{cases} \tag{1}$$

where $\lambda_l^{(i)}$ represents the LLR, and the superscript i and subscript l denote the bit and path indices, respectively.

At the first $\log_2 L$ unfrozen bits, all paths are kept in the list and the set of these bits is labelled as $\mathcal{A}'$. At bit $\mathcal{A}\backslash\mathcal{A}'$, L paths are forked to $2L$ sub-paths, which forms an expanded list $\mathcal{L}^{(i)}$. Then, the bit sequences of the most reliable sub-paths are collected in the set $\mathcal{L}_{\mathrm{best}}^{(i)}$ as the candidate paths.

C. Successive Cancellation List Flip Decoding

An alternative to SCL decoding for improving the error rate performance by increasing the number of candidate bit sequences is SCF decoding [10]. When SCL and SCF decoding are combined, the search space for valid code words will be further enlarged, which also provides a middle ground between parallel and sequential attempts for a given FER performance target. The corresponding algorithm is named SCLF, which flips the path selection of the initial attempt multiple times once all paths of the original SCL decoding fail the CRC.

At a flip position, the original SCLF decoder [12] only explores the paths that originally had a single forked path and selects their originally discarded forked paths. The authors of [13] and [14] independently proposed a better flip strategy, which *considers all L previously discarded path-expansion candidates* for a flip location. Namely, if i is a flip index, the SCLF decoder selects the least reliable L sub-paths from $\mathcal{L}^{(i)}$ when it revists this location in an additional decoding attempt. Denote the maximum number of additional attempts as T. Then the T flip indices form a flip set $\mathcal{S}$. While the flip set in [14] is a fixed critical set, [13] uses a better real-time calculated criterion to determine the flip indices, but it ignores the impact of frozen bits and the paths in $\mathcal{L}_{\mathrm{best}}^{(i)}$ [15]. The corresponding flip metric $M_\alpha^{(i)}$ was further optimized in [15], leading to the following equation, which is the best so far reported in the literature

$$M_\alpha^{(i)} = \ln\left(\sum_{l=0}^{L-1} e^{-\mathrm{PM}_l^{(i)}} \Big/ \left(\sum_{l=0}^{L-1} e^{-\mathrm{PM}_{l+L}^{(i)}}\right)^{\eta}\right), \tag{2}$$

where PMs are assumed to be sorted in ascending order and η is a normalization factor that mitigates the biased estimate due to the propagated errors [15]. The metric $M_\alpha^{(i)}$ of the first $\log_2 L$ unfrozen bits is set to infinity. The SCLF decoding procedure is listed in Algorithm 1, where `SCL_Decoding()` with an input set denotes standard SCL decoding during which the path selection at the bit indices given in this set is flipped.

In order to further improve the FER performance, the authors of [15] explore a second flip location, once all trials with a single flip fail the CRC. Unfortunately, to approach the FER of an SCL decoder with $L = 32$ using an SCLF decoder with only $L = 4$, even their improved SCLF decoder still requires nearly one hundred attempts [15] and consequently has an impractical worst-case decoding latency.

Algorithm 1: (Dynamic) SCLF Decoding

```
//extra steps of the proposed D-SCLF are in shadow
û ← SCL_decoding(∅);
//At each CRC bit, prune invalid selected paths
if CRC detection succeeds then
    return û;
else
    S ← bit index of T smallest M_α;// intialize S
    for t = 0 to T − 1 do
        û ← SCL_decoding(S_t);
        //For each i ∈ S_t, select paths from L^(i)\L_best^(i)
        //At each CRC bit, prune invalid selected paths
        if CRC detection succeeds then
            return û;
        else
            S(t+1:T−1) ←
            bit index of T−t−1 smallest S(t+1:T−1) ⋃ M_α(S_t);
    return û;
```

III. Proposed D-SCLF Decoding Algorithm

Although the metric in [15] results in currently the best FER performance of SCLF decoding with a single flip index, this metric was heuristically proposed. In contrast, the metric in [13] was derived by evaluating the probability that the first erroneous path selection occurs at each bit index, but inappropriately considers only the effect of unfrozen bits. Inspired by [17], we derive this probability by considering both the effects of frozen bits and error propagation, which results in a novel and more rigorous flip metric.

Based on the proposed flip metric, we propose a D-SCLF decoding algorithm. The flip set in D-SCLF decoding keeps a constant size but its contents are dynamically adjusted after each flip trial, so that the metric of the next flip locations is

always the lowest among current metrics. Moreover, the D-SCLF decoding takes the advantage of the distributed CRC in 5G PDCCH and PBCH, which can prune invalid paths early based on the parity check.

A. Optimized Flip Metric

We first begin with the order-1 SCLF decoder that performs a flip at only a single bit index in each attempt. Define $\mathcal{E}^{(i)}$ as the event that $u_0^{i-1} \in \mathcal{L}_{\text{best}}^{(i-1)}$ and that $u_0^i \notin \mathcal{L}_{\text{best}}^{(i)}$, then the probability that $\mathcal{E}^{(i)}$ occurs is

$$\begin{aligned} P\Big(\mathcal{E}^{(i)}|\boldsymbol{y}\Big) &= P\Big(u_0^{i-1} \in \mathcal{L}_{\text{best}}^{(i-1)}, u_0^i \notin \mathcal{L}_{\text{best}}^{(i)}|\boldsymbol{y}\Big) \\ &= P\Big(u_0^i \notin \mathcal{L}_{\text{best}}^{(i)}|\boldsymbol{y}, u_0^{i-1} \in \mathcal{L}_{\text{best}}^{(i-1)}\Big) \\ &\quad \cdot \prod_{\substack{k<i \\ k\in\mathcal{A}\setminus\mathcal{A}'}} P\Big(u_0^k \in \mathcal{L}_{\text{best}}^{(k)}|\boldsymbol{y}, u_0^{k-1} \in \mathcal{L}_{\text{best}}^{(k-1)}\Big) \\ &= P_e^{(i)} \cdot \prod_{\substack{k<i \\ k\in\mathcal{A}\setminus\mathcal{A}'}} \Big(1 - P_e^{(k)}\Big), \end{aligned} \tag{3}$$

where $P(u_0^i \notin \mathcal{L}_{\text{best}}^{(i)}|\boldsymbol{y}, u_0^{i-1} \in \mathcal{L}_{\text{best}}^{(i-1)})$ is abbreviated by $P_e^{(i)}$. Since the decoder cannot acquire knowledge of the correct bit sequence, we can only approximate $P_e^{(i)}$ by $P(u_0^i \notin \mathcal{L}_{\text{best}}^{(i)}|\boldsymbol{y}, \mathcal{L}_{\text{best}}^{(i-1)})$, which denotes the probability of $u_0^i \notin \mathcal{L}_{\text{best}}^{(i)}$ under the condition of receiving $\boldsymbol{y}$ and currently holding the L paths related to $\mathcal{L}_{\text{best}}^{(i-1)}$. According to [7], the PM represents the negative log-probability that each path is correct. Therefore, $P_e^{(i)}$ can be formulated as

$$P_e^{(i)} = \left(\sum_{l=0}^{L-1} e^{-\text{PM}_{l+L}^{(i)}} \Big/ \sum_{l=0}^{2L-1} e^{-\text{PM}_l^{(i)}}\right). \tag{4}$$

Introducing the probability ratio of the most reliable L sub-paths to the remaining ones in the log-domain based on $\mathbb{L}^{(i)} = \ln(\sum_{l=0}^{L-1} e^{-\text{PM}_l^{(i)}} / \sum_{l=0}^{L-1} e^{-\text{PM}_{l+L}^{(i)}})$, (4) can be rewritten as

$$P_e^{(i)} = \frac{1}{1 + e^{\alpha \cdot \mathbb{L}^{(i)}}}, \tag{5}$$

where α is a factor to compensate the employed approximation for $P_e^{(i)}$ [17]. Herewith, for each $i \in \mathcal{A} \setminus \mathcal{A}'$, we define the proposed flip metric in the log-domain as:

$$M_\alpha^{(i)} = -\frac{1}{\alpha} \ln P(\mathcal{E}^{(i)}|\boldsymbol{y}) = \mathbb{L}^{(i)} + \frac{1}{\alpha} \sum_{\substack{k<i \\ k\in\mathcal{A}\setminus\mathcal{A}'}} \Big(1 + e^{-\alpha \cdot \mathbb{L}^{(k)}}\Big). \tag{6}$$

The second term in (6) involves the error probability caused by error propagation, which leads to a relatively large metric value for a large bit index. By contrast, the metric of (2) in [15] discards the second term in (6), but amplifies the value of $\mathbb{L}^{(i)}$ by adding the factor η for all bit indices. As shown in [18], a large bit index presents a relatively wide PM range, so that the flip metric for a larger index in [15] is amplified more, which explains why adding a factor η in (2) is able to mitigate the biased estimate caused by error propagation to some extent. Nevertheless, our derivation provides a more accurate metric to identify the flip locations, as we demonstrate in Section IV.

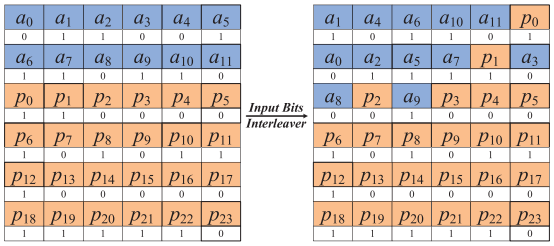

Fig. 1. The illustration of the interleaver to reorder unfrozen bits when $A = 12$, $\boldsymbol{a} = \{0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0\}$, and the channel is PDCCH, where the shift registers in CRC encoder are initialized as all-one [6].

B. Proposed Dynamic SCLF Decoding

Following the extension from [17] to [19], we extend the metric of (6) to support dynamic higher-order SCLF decoding. The dynamic flip set $\mathcal{S}$ has a constant size of T but is updated after each decoding attempt. The set $\mathcal{S}$ is composed of T subsets, i.e., $\mathcal{S} = \{\mathcal{S}_0, \ldots, \mathcal{S}_t, \ldots, \mathcal{S}_{T-1}\}$, in which the set $\mathcal{S}_t$ records all flip indices for the t-th additional decoding attempt. With the flip indices of $\mathcal{S}_t$, the probability that $\mathcal{E}^{(i)}$ occurs is

$$P\Big(\mathcal{E}^{(i)}|\boldsymbol{y}, \mathcal{S}_t\Big) = P_e^{(i)} \cdot \prod_{k\in\mathcal{S}_t} P_e^{(k)} \cdot \prod_{\substack{k<i \\ k\in\{\mathcal{A}\setminus\mathcal{A}'\}\setminus\mathcal{S}_t}} \Big(1 - P_e^{(k)}\Big), \tag{7}$$

Following the approximation scheme used in order-1 flip decoding, the updated metric is

$$M_\alpha^{(i)}(\mathcal{S}_t) = \mathbb{L}^{(i)} + \sum_{k\in\mathcal{S}_t} \mathbb{L}^{(k)} + \frac{1}{\alpha} \sum_{\substack{k<i \\ k\in\{\mathcal{A}\setminus\mathcal{A}'\}\setminus\mathcal{S}_t}} \Big(1 + e^{-\alpha \cdot \mathbb{L}^{(k)}}\Big) \tag{8}$$

when path selection at indices in $\mathcal{S}_t$ is flipped.

For the 5G PDCCH and PBCH, the input bits interleaver reorders the unfrozen bits by $c_i' = c_{\Pi(i)}$, where Π is the interleaving pattern specified by the 5G standard [6]. As illustrated in Fig. 1, some of the parity bits are then distributed among information bits, which enables early termination when none of the candidate paths in the list can meet the parity constraint. Moreover, the distributed CRC can help to improve the accuracy of the path selection [20]. Specifically, the check-select (CS) strategy directly calculates the CRC bit value based on the parity constraint instead of decoding it as part of the path expansion. The alternative check-remove (CR) strategy, discards the invalid paths that do not satisfy current parity among the selected L sub-paths. In the following, we explain how to exploit the distributed CRC to further improve the performance of D-SCLF decoding.

The CS strategy is not applicable to SCLF decoding, because all L CRC-satisfying paths cannot activate the flip procedure. By contrast, the CR strategy can be applied: when meeting a parity bit, the invalid sub-paths can be removed to leave space for the descendants of parity-satisfying sub-paths. In this way, all the maintained candidates at the end of each decoding attempt satisfy the CRC unless this attempt is terminated early. In that case, the candidate flip locations exclude all indices after the termination point, which leads to a more accurate flip set $\mathcal{S}$. It should be noted that in SCL decoding, the path number within $\mathcal{A} \setminus \mathcal{A}'$ is no longer constant as L due to the CR strategy. To avoid revisiting one candidate path when the path selection is flipped at bit index i, we select the paths that correspond to the bit sequences beyond the set $\mathcal{L}_{\text{best}}^{(i)}$ instead of the L least reliable ones, which also leads to a reduction of computations in each decoding attempt.

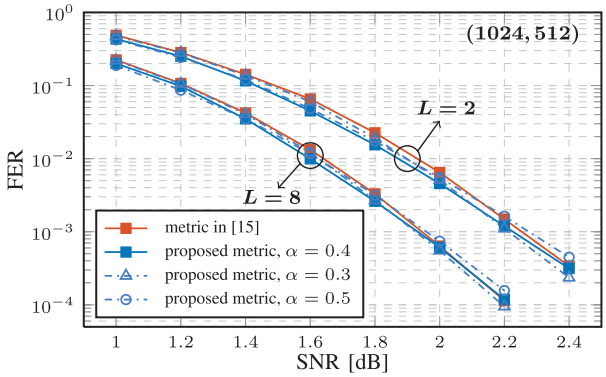

Fig. 2. FER performance of an SCLF decoder with order one and $T = 10$ using different metrics to decode a (1024, 512) code for PUCCH.

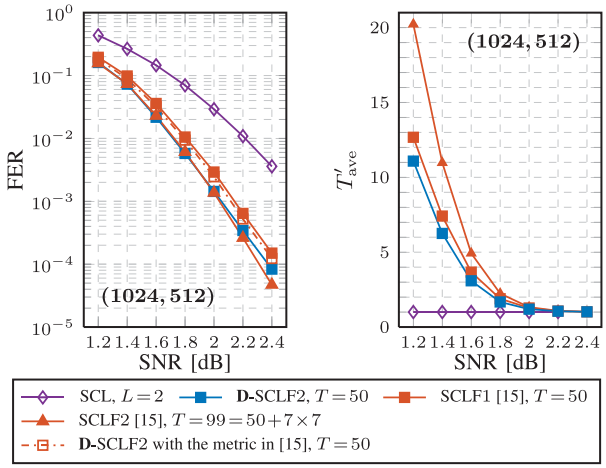

Fig. 3. FER performance and average attempt number of different SCLF decoders with $L = 2$.

The steps of our D-SCLF decoding are listed in Algorithm 1, where the additional part with respect to SCLF decoding is marked with a shadow.

IV. Experimental Results

In this section, we provide the FER performance results of the proposed D-SCLF decoder. In the simulations, the code words are modulated using binary shift-keying modulation and transmitted over additive white Gaussian noise channels. In the following figures, the SCLF decoding with a maximum flip order ω is denoted as SCLFω for brevity.

Fig. 2 compares the FER performance of SCLF decoding using our proposed metric and the metric that employs $\eta = 1.2$ in [15]. As discussed in [19], the optimal value of α decreases with the SNR and can be determined by Monte-Carlo simulations. Comparing the FER curves using $\alpha = 0.3, 0.4$, and 0.5 in Fig. 2, we can observe that the SCLF decoder with $\alpha = 0.4$ requires the lowest SNRs in the FER region of 10^{-3} to 10^{-2} and outperforms the SCLF decoder using the metric in [15] by 0.05 dB at FER$= 10^{-2}$ when $L = 2$. In practice, a constant value of α is preferred for every SNR and code. Since our decoder is not sensitive to the choice of α, we fix $\alpha = 0.4$.

Fig. 3 shows the FER performance and average attempt number (denoted by T'_{ave}) of our D-SCLF decoder compared to the state-of-the-art flip metric based SCLF decoder [15]. We follow the same parameters as [15], i.e., $T = 50$ for (1024, 512) polar codes, but the CRC length is 11, consistent with the PUCCH. Compared with the SCLF1 decoder in [15], our D-SCLF2 decoder enjoys a 0.1 dB gain at an FER of 10^{-3}, because it can execute either order-1 or order-2 flips during these 50 attempts. In [15], order-2 flip decoding is also discussed, but only after all order-1 flip decoding attempts fail the CRC. It can be seen that a total of up to 99 additional attempts in [15] can lead to a similar performance to our D-SCLF decoder with only up to 50 additional attempts, at the expense of much higher average and worst-case computational complexity. It is noted that the metric in [15] can also be adapted to the dynamic flip decoding, which is the sum of the η-normalized $\mathbb{L}$ value of each flip location. Since our metric considers the error propagation more carefully, the D-SCLF2 decoder using our metric is 0.1 dB better than using the metric in [15] at an FER of 10^{-3}.

For PDCCH, there are five coded block sizes E that are typically used in practice: 108, 216, 432, 864, and 1728, and the value range of A is from 12 to 140 [21]. Fig. 4 compares the FER performance of the proposed D-SCLF decoder and the state-of-the-art SCLF decoder [15], given $E = 432$ and $A = 12$, 16, and 100. Due to the early termination, we use the cumulative number of paths (CNP) over $\mathcal{A}$ to measure the computational complexity, instead of the number of attempts. In Fig. 4, all D-SCLF decoders are based on decoding with $L = 4$. In this case, the number of unfrozen bits is 36 for $A = 12$. Hence, the CNP in standard SCL decoding is $2 + 4 \times 35 = 142$. Due to the short information length in PDCCH, the SCLF decoder requires fewer decoding attempts to obtain an obvious performance gain than in PUCCH. With a limited number of flip trials, the gain brought by the proposed flip metric is small. However, the consideration of distributed CRC provides advantages in terms of both FER performance and average complexity, especially for small A. When $A = 12$, $L = 4$, and $T = 15$, our D-SCLF2 decoder outperforms the SCLF1 decoder [15] by 0.5 dB at an FER of 10^{-3}, and its CNP is 21.5% less than that of the SCLF1 decoder [15] at -6.5 dB.

In Fig. 5, we fix the targeted FER to 10^{-3} and present the required SNRs for all typical configurations of (E, A) polar codes in PDCCH. Compared to the baseline SCL decoder with $L = 4$, our D-SCLF decoder with $T = 3$ can achieve 0.2$\sim$0.7 dB gains, which is comparable to the SCL decoder with $L = 8$ in terms of error-correcting performance. Taking the CNP of the SCL decoder with $L = 4$ as benchmark and normalizing the CNP of the D-SCLF decoder, we find that when $T = 3$, the average complexity of the D-SCLF decoder is reduced by 5%$\sim$19%, because invalid paths are removed during each decoding attempt. In contrast, the SCLF decoder in [15] needs to process more paths than the SCL decoder with $L = 4$. The results show the potential of the D-SCLF decoder in PDCCH, where the area and energy are the most critical cost factors [22]. This is because the proposed D-SCLF decoder based on $L = 4$ has nearly half the LLR storage and the average number of computations less than the SCL decoder with $L = 8$, at the cost of up to only three additional attempts.

V. Conclusion

In this letter, we propose the D-SCLF decoding algorithm with a rigorous derivation of the flip metric, which also considers the distributed CRC for 5G downlink channels. When we focus on an FER of 10^{-3}, the conclusions are as follows. For PUCCH (1024, 512) polar codes, the D-SCLF2 decoder ($T = 50$) using the proposed metric is 0.1 dB better than using the state-of-the-art metric. For all possible polar codes in PDCCH, the performance of our D-SCLF decoder with $L = 4$ and only $T = 3$ is comparable to that of the SCL decoder with $L = 8$, with lower average complexity (energy).

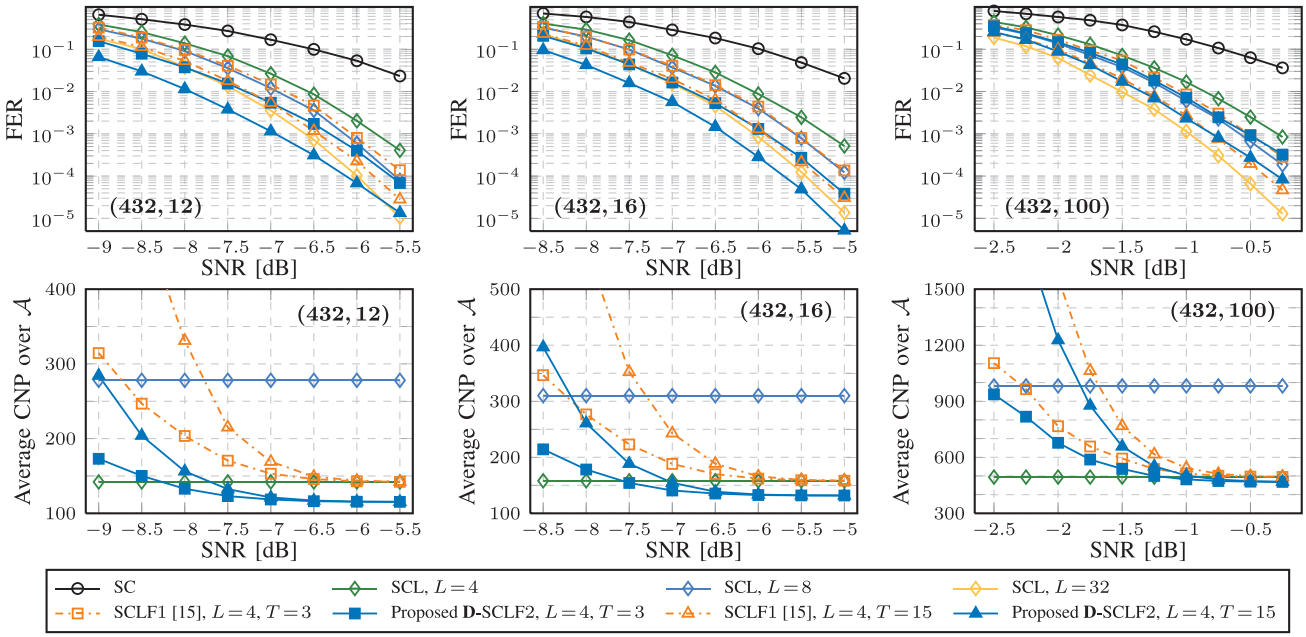

Fig. 4. FER performance and average CNP of the SCL decoders, the SCLF1 decoders [15] and the proposed D-SCLF2 decoders.

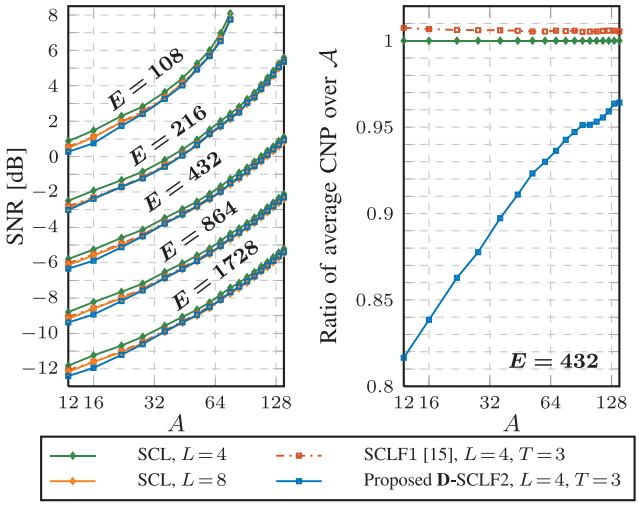

Fig. 5. Required SNRs and CNP ratios of SCL and SCLF decoders for all typical (E, A) configurations in PDCCH, where the target FER is 10^{-3}.

REFERENCES

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[3] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.

[4] A. Balatsoukas-Stimming, P. Giard, and A. Burg, "Comparison of polar decoders with existing low-density parity-check and turbo decoders," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop (WCNCW)*, San Francisco, CA, USA, 2017, pp. 1–6.

[5] *Chairman's Notes of Agenda Item 7.1.5 Channel Coding and Modulation*, document 3GPP TSG RAN WG1 meeting #87, R1-1613710, 3GPP, Reno, NV, USA, Nov. 2016.

[6] *5G NR: Multiplexing and Channel Coding*, 3GPP Standard TS 38.212 version 15.2.0, Jul. 2018.

[7] 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.

[8] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast list decoders for polar codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 318–328, Feb. 2016.

[9] Z. Zhang, L. Zhang, X. Wang, C. Zhong, and H. V. Poor, "A split-reduced successive cancellation list decoder for polar codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 292–302, Feb. 2016.

[10] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *Proc. IEEE Asilomar Conf. Signal Syst. Comput. (ACSSC)*, Pacific Grove, CA, USA, 2014, pp. 2116–2120.

[11] F. Ercan, T. Tonnellier, N. Doan, and W. J. Gross, "Practical dynamic SC-Flip polar decoders: Algorithm and implementation," *IEEE Trans. Signal Process.*, vol. 68, pp. 5441–5456, Sep. 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9195767

[12] Y. Yongrun, P. Zhiwen, L. Nan, and Y. Xiaohu, "Successive cancellation list bit-flip decoder for polar codes," in *Proc. IEEE Int. Wireless Commun. Signal Process. (WCSP)*, Hangzhou, China, 2018, pp. 1–6.

[13] 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.

[14] M. Rowshan and E. Viterbo, "Improved list decoding of polar codes by shifted-pruning," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, 2019, pp. 1–5.

[15] Y.-H. Pan, C.-H. Wang, and Y.-L. Ueng, "Generalized SCL-flip decoding of polar codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, 2020, pp. 1–6.

[16] C.-H. Wang, Y.-H. Pan, Y.-H. Lin, and Y.-L. Ueng, "Post-processing for CRC-aided successive cancellation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 24, no. 7, pp. 1395–1399, Jul. 2020.

[17] L. Chandesris, V. Savin, and D. Declercq, "An improved SCFlip decoder for polar codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[18] M. Rowshan and E. Viterbo, "Stepped list decoding for polar codes," in *Proc. Int. Symp. Turbo Codes (ISTC)*, Hong Kong, 2018, pp. 1–5.

[19] L. Chandesris, V. Savin, and D. Declercq, "Dynamic-SCFlip decoding of polar codes," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2333–2345, Jun. 2018.

[20] C. Pillet, V. Bioglio, and C. Condo, "On list decoding of 5G-NR polar codes," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Seoul, South Korea, 2020, pp. 1–6.

[21] Z. B. K. Egilmez, L. Xiang, R. G. Maunder, and L. Hanzo, "The development, operation and performance of the 5G polar codes," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 96–122, 1st Quart., 2020.

[22] S. Scholl, S. Weithoffer, and N. Wehn, "Advanced iterative channel coding schemes: When Shannon meets moore," in *Proc. IEEE Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Brest, France, 2016, pp. 406–411.