A CRC-Aided SR-SCL Decoding Algorithm for Polar Codes

Li Zhang, Anqi Cao, Jian Qiao and Yejun He⋆
Guangdong Engineering Research Center of Base Station Antennas
Shenzhen Key Laboratory of Antennas and Propagation
College of Electronics and Information Engineering, Shenzhen University, 518060, China
Email: heyejun@126.com⋆

***Abstract*—In order to make a better trade-off between computational complexity and decoding performance, a cyclic redundancy check (CRC)-aided split-reduced successive cancellation list (CA-SR-SCL) algorithm is proposed in this paper. Based on the splitting rule defined in split-reduced SCL (SR-SCL) algorithm and enhanced-SR-SCL (ESR-SCL) algorithm, the splitting characteristics of the correct decoding path are calculated. In addition, a new pruning rule is proposed to timely delete the candidate paths that split frequently, and a CRC detector is appended to improve the error correction performance. Compared to the ESR-SCL algorithm, simulation results demonstrate that the proposed algorithm can achieve better performance with lower computational complexity.**

***Index Terms*—Polar codes, successive cancellation, list decoding, split-reduced, tree-pruning.**

I. Introduction

A low complexity successive cancellation (SC) decoding algorithm [1] is proposed by Arikan for polar codes. Since the excellent performance of SC decoding algorithm decreases with the shorter code length, SC list (SCL) algorithm [2] and CRC-aided SCL (CA-SCL) algorithm [3] are proposed. List decoding algorithms improve the decoding performance in short and medium codes by path splitting extension, which also brings L times the computational complexity of the SC algorithm, where L is the list size.

The SR-SCL and ESR-SCL decoding algorithms [4] are proposed to reduce the computational complexity of SCL algorithm. In SR-SCL algorithm, a novel splitting rule is defined, which performs hard decisions at high-reliability nodes instead of path splitting. In the decoding process, pruning is carried out when the number of stages without splitting for a path exceeds the threshold. One more step of the ESR-SCL algorithm is to decompose SCL decoding into SC decoding after a specified node. ESR-SCL algorithm further reduces the computational complexity of decoding and maintains a similar decoding performance to the SR-SCL algorithm.

In this paper, based on the splitting rule in [4], we count the number of splits for the correct decoding path, then propose a CRC-aided SR-SCL decoding algorithm for polar codes. During decoding, the proposed scheme counts the splitting times for each path and timely deletes the paths that split frequently. Then CRC is introduced to help decoding. The results show that the decoding performance is improved. Moreover, with the increase of signal-to-noise ratio (SNR), the computational complexity of CA-SR-SCL algorithm is lower than that of ESR-SCL algorithm.

II. Preliminaries

A. SC and SCL Decoding

Denote a polar code by $(N, K, \mathcal{A}, u^c_{\mathcal{A}})$, where $N, K, \mathcal{A}$, and $u^c_{\mathcal{A}}$ are the code length, information length, set of information bits, and value of frozen bits, respectively. For code length $N = 2^n$, code rate $R = K/N$. The received vector is $y_1^N = (y_1, y_2, ..., y_N)$. The estimation is $\hat{u}_1^N = (\hat{u}_1, \hat{u}_2, ..., \hat{u}_N)$. If u_i is a frozen bit, $\hat{u}_i = u_i$. Otherwise, the decoder computes the logarithmic likelihood ratio (LLR) of each bit channel [5] as

$$L(u_i) = log(\frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)}), \tag{1}$$

where $W_N^{(i)}$ is the channel transition probability. For SC decoding algorithm, $\hat{u}_i$ is determined by a hard-decision

$$\hat{u} = \begin{cases} 0 & \text{if } L(u_i) \geq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

For SCL decoding algorithm, both options are kept for each information bit. Therefore, each decoding path splits into two paths. The decoder then selects the best L candidates by updating the path metrics (PMs)

$$PM_l^{(i)} = \begin{cases} PM_l^{(i-1)} & \text{if } \hat{u} = \frac{1}{2}[1 - sign(L_l(u_i))], \\ PM_l^{(i-1)} + |L_l(u_i)| & \text{otherwise.} \end{cases} \tag{3}$$

where $L_l(u_i)$ is the LLR of path l. At the last stage, SCL algorithm outputs the path with the smallest PM, while CA-SCL algorithm gives priority to the path survived the CRC detector.

B. SR-SCL and ESR-SCL decoding

In the SCL algorithm, path splitting occurs for each unfrozen bit. In the SR-SCL algorithm, a new splitting rule [4] is defined as

$$\hat{u}_i = \begin{cases} 0 & L_l(u_i) > \log\frac{1-P_e(u_i)}{P_e(u_i)}, \\ 1 & L_l(u_i) < -\log\frac{1-P_e(u_i)}{P_e(u_i)}, \\ \text{split} & \text{otherwise.} \end{cases} \tag{4}$$

978-1-7281-0716-5/19/$31.00 ©2019 IEEE

where $P_e(u_i)$ is the error probability of u_i. As formula (4) shows, splitting is avoided when the reliability of decoding the unfrozen bit is high enough.

For list pruning, SR-SCL decoding algorithm introduces a counter $\omega_l[i]$ to count the number of stages that the path l survives without splitting. When the number of paths exceeds L, if $\omega_l[i]$ reaches the predefined threshold ω, SR-SCL algorithm prunes the path whose counter is less than ω. On this basis, ESR-SCL algorithm degrades to SC decoding after the index $N-K/4+1$, which achieves lower computational complexity without degrading the performance of SR-SCL algorithm.

III. Proposed CRC-Aided SR-SCL Decoding Algorithm

In order to improve the performance of SR-SCL decoding, a CRC-aided scheme is appended to the proposed algorithm. In this section, we first calculate the splitting characteristics of the correct decoding path, then propose a CA-SR-SCL decoding algorithm.

A. Splitting Characteristics of The Correct Decoding Path

On the basis of the splitting rule, if the LLR of u_i satisfies $|L(u_i)| \leqslant \log\frac{1-P_e(u_i)}{P_e(u_i)}$, the correct decoding path splits once. Fig. 1 shows the relative frequency of the number of path splitting for four different E_b/N_0 values. The distribution of the number of splits approximates a normal distribution. As the E_b/N_0 increases, the numbers become more and more concentrated towards 0.

Because the correct path rarely splits and the incorrect path frequently splits [4], the proposed scheme deletes frequently splitting paths to reduce the computational complexity. Based on Fig. 1, if the cumulative number of splitting for each candidate path is counted and the path with a higher number is deleted at each stage, the performance loss caused by pruning can be completely avoided by setting the number threshold to a certain value.

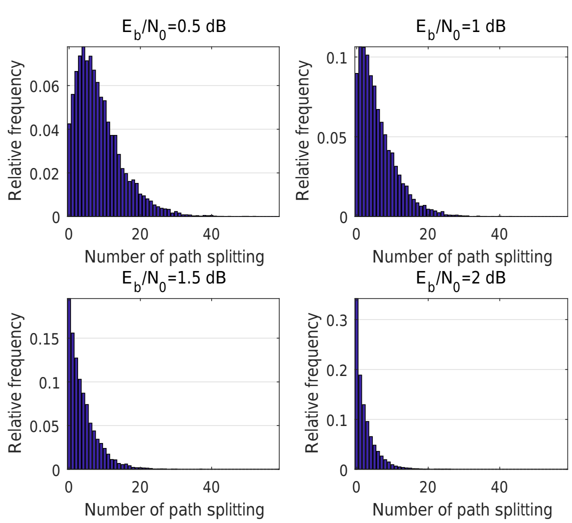

Fig. 1. The relative frequency of the number of path splitting for the correct decoding path with N = 256, R = 0.5.

B. CA-SR-SCL Decoding Algorithm

In this section, a CRC-aided split-reduced decoding scheme is proposed to improve the performance of SR-SCL. As shown in Fig.1, the threshold for the cumulative number of splits should be set at more than 15 to minimize the loss of performance. However, it is unnecessary to retain the paths with the number of splits for 15 or more when the correct decoding path splits only once. For further reduce the decoding complexity, the proposed CA-SR-SCL algorithm adopts the relative number of splitting for pruning.

CA-SR-SCL algorithm follows the splitting rule of formula (4). Counter ω_l is introduced to count the cumulative splitting number of path l. If path l survives without split at stage i, the counter holds its value at stage $i-1$. If it splits into two paths $l1$ and $l2$, $\omega_{l1} = \omega_{l2} = \omega_l + 1$. The relative number of path splitting for path l at stage i is defined as

$$S_l[i] = \omega_l - \underset{l=1,2\cdots M}{min} \omega_l \tag{5}$$

where M is the path number after splitting at stage i. After the path splitting is completed, the candidate path with $S_l[i] \geqslant T$ is discarded. In this way, the proposed scheme can achieve lower computational complexity by timely deleting the frequently splitting paths.

Algorithm 1: CA-SR-SCL Decoding

Input: $y_1^N, \mathcal{A}, L, Pe(u_1^N)$ and threshold T
initialization;
$M \longleftarrow 1$;
for $i \in \mathcal{A}$ **do** /* calculations for $i \notin \mathcal{A}$ are omitted */
  for $l \in [0, M-1]$ **do**
    compute $L_l(u_i)$;
    if $|L_l(u_i)| > log\frac{1-Pe(u_i)}{Pe(u_i)}$ **then**
      $\hat{u}_i[l] \longleftarrow 0$ or 1 according to Eq. (4);
    else
      path l splits into two paths l_1 and l_2;
      $\omega_{l1} \longleftarrow \omega_l + 1$;
      $\omega_{l2} \longleftarrow \omega_l + 1$;
    update the PMs based on Eq. (3);
  compute $S_l[i]$ and discard the path with $S_l[i] \geqslant T$;
  $M \longleftarrow$ current path number;
  if $M > L$ **then**
    keep the L candidate paths with smallest PMs;
    $M \longleftarrow L$;
$mark \longleftarrow$ the first l of passCRC($\hat{u}_1^N[l]$);
if $mark$ = *false* **then**
  declare a decoding failure;
else
  return $\hat{u}_1^N[mark]$;

As shown in Algorithm 1, when $S_l[i]$ reaches the threshold T, path l is deleted. If the number of remaining paths is greater than L, L paths with the smallest PMs are selected for the next stage. After the last bit is decoded, CRC-aided path selection is performed. If no path survives, the algorithm declares the decoding failure. Fig. 2 shows the decoding procedure of the CA-SR-SCL algorithm. The splitting paths are also frequently deleted in time.

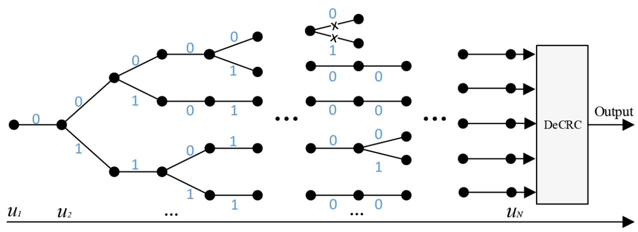

Fig. 2. Decoding procedure of CA-SR-SCL decoding.

IV. Experimental results and performance analysis

In this section, we compare the proposed CA-SR-SCL decoding algorithm with ESR-SCL decoding and conventional decoding algorithms in decoding performance and computational complexity with N = 256, R = 0.5, L = 8. The cyclic generator polynomial $g(x) = x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$ [6].

Fig. 3 shows the block error rate (BLER) performance of proposed CA-SR-SCL decoding with different T and ESR-SCL decoding with $\omega = 30$ and $\omega = 60$. As observed in [4], when ω increases, although the performance of ESR-SCL decoding improves, it is still not better than that of SCL decoding. From Fig. 3, the performance of proposed CA-SR-SCL decoding improves with T increases, and the performance of CA-SR-SCL decoding is much better than that of SCL but worse than that of CA-SCL decoding. In addition, when T = 6 and ω = 60, CA-SR-SCL decoding and ESR-SCL decoding almost achieve their optimal performance, respectively.

The computational decoding complexity is described by the average list size. Fig. 4 compares the average list size of CA-SR-SCL decoding with that of ESR-SCL decoding. As T increases, the computational decoding complexity of CA-SR-SCL algorithm increases. When E_b/N_0 is greater than 1.5 dB, if T = 4, the average list size of proposed CA-SR-SCL algorithm is smaller than that of ESR-SCL algorithm with $\omega = 30$; if T = 5, the size is smaller than that of ESR-SCL algorithm with $\omega = 60$ even 30 when E_b/N_0 reaches 2.5. When E_b/N_0 is greater than 2 dB, the average list size of CA-SR-SCL algorithm with T = 6 is smaller than that of ESR-SCL algorithm with $\omega = 60$. Based on the above observation, the proposed algorithm can achieve better decoding performance with lower computational complexity.

V. Conclusions

In this paper, the splitting characteristics of the correct decoding path are calculated, and based on observations of these characteristics, a CRC-aided SR-SCL decoding algorithm is proposed. A new pruning rule and CRC-aided scheme are appended to the proposed algorithm. Simulation results show that the proposed CA-SR-SCL algorithm improves the decoding performance with lower computational complexity.

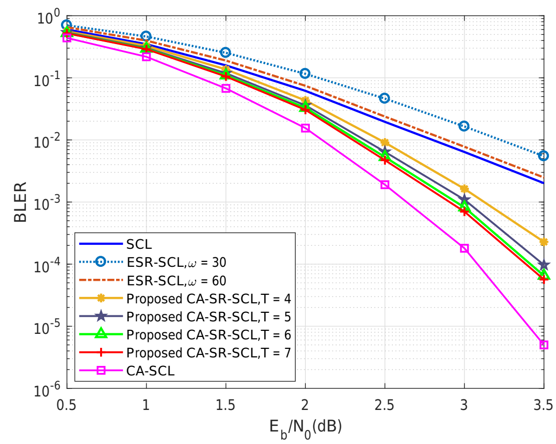

Fig. 3. Performance comparison of different decoding algorithms.

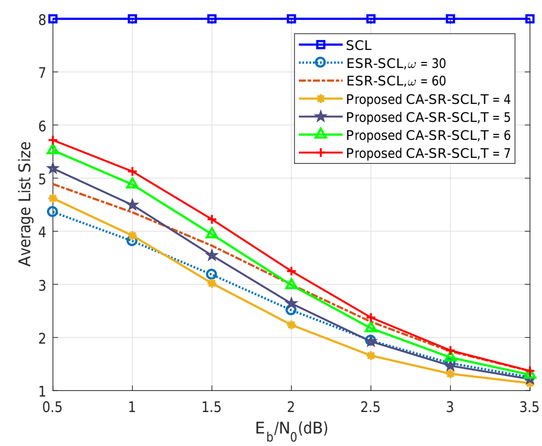

Fig. 4. Average list size of different decoding algorithms.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants 61372077, 61801299, and 61871433, in part by the Shenzhen Science and Technology Programs under Grants ZDSYS 201507031550105, JCYJ 20170302150411789, JCYJ 20170302142515949, GCZX 2017040715180580, and GJHZ 20180418190529516, in part by the Guangzhou Science and Technology Program under Grant 201707010490.

References

[1] E. Arikan, "Channel Polarization: A Method for Costructing Capacity-Achieving Codes for Symmetic 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 Communications Letters*, vol. 16, no. 10, pp. 1668-1671, Oct. 2012.
[4] Z. Zhang, L. Zhang, X. Wang, C. Zhong and H. V. Poor, "A Split-Reduced Successive Cancellation List Decoder for Polar Codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 292-302, Feb. 2016.
[5] A.Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5165-5179, Oct. 2015.
[6] 3GPP, Multiplexing and channel coding (FDD), Release 9, 3GPP TS 25.212, 2009.