Path Splitting Selecting Strategy-Aided Successive Cancellation List Algorithm for Polar Codes

Chenyu Gao, Rongke Liu, *Member, IEEE*, Bin Dai, and Xu Han

***Abstract*— Successive cancellation list (SCL) decoder provides a good trade-off between error-correcting performance and complexity. To simplify the decoding algorithm with negligible performance loss, a path splitting selecting (PSS) strategy is proposed based on analyzing the reason why the SCL decoder fails. On the basis of the PSS strategy, two schemes are proposed to find erroneous information bits more accurately. Simulation results show that the proposed strategy based on the search set and decision function is a good solution to reduce the complexity with almost no performance loss.**

***Index Terms*— Polar codes, successive cancellation algorithm, successive cancellation list algorithm.**

I. Introduction

POLAR coding is a popular research direction of channel coding recently. As proved by Arıkan [1], polar codes are the first capacity-achieving channel codes under the successive cancellation (SC) algorithm. However, the performance of the SC algorithm is not ideal at finite code lengths, due to the insufficient channel polarization. The successive cancellation list (SCL) algorithm [2], which decodes multiple paths of the SC algorithm simultaneously, is proposed to improve the performance of the SC algorithm. Furthermore, a cyclic redundancy check (CRC) is added to polar codes to filter the decoding paths, which is called the CA-SCL algorithm [3].

Although the SCL algorithm can improve the decoding performance, it brings a large computational complexity. In order to make the decoding algorithm easy to implement, a series of simplified SCL decoding algorithms have been proposed successively. The concept of Rate-0 node and Rate-1 node is provided in [4], which can estimate a group of information bits simultaneously during the SC decoding process. On this basis, more special nodes [5] and optimization strategies [6], [7] based on these special nodes are proposed to further simplify the decoding algorithm. In [8], a decision-aided parallel SCL algorithm is proposed to split the decoding paths at some low reliable information bits, which can reduce the complexity. Later, in [9], the SC-Flip algorithm is proposed to improve the performance of the SC algorithm and bring a lower average complexity than the SCL algorithm, based on retrying the value of the unreliable bits to correct the errors caused by the channel noise. However, this algorithm may bring extremely high delay to some extent and is not easy to parallelize. Similar to the SC decoder, the errors are caused by the channel noise or the error propagation in the SCL decoder. As a result, if we can split the decoding paths at erroneous information bits caused by the channel noise, a better performance and a lower complexity can be expected simultaneously.

Manuscript received November 19, 2018; revised December 18, 2018; accepted January 13, 2019. Date of publication January 22, 2019; date of current version March 8, 2019. The work was supported by the National Natural Science Foundation of China (91438116, 61871009). The associate editor coordinating the review of this paper and approving it for publication was H. Pishro-Nik. *(Corresponding author: Rongke Liu.)*

The authors are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: gaochenyu@buaa.edu.cn; rongke_liu@buaa.edu.cn; daibinok@buaa.edu.cn; hanxu0225@buaa.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2019.2894150

Hence, a path splitting selecting (PSS) strategy based on two schemes is proposed in this letter to bring an efficient path splitting operation and reduce the complexity of the SCL algorithm. In addition, the PSS strategy can also be used in other simplified SCL algorithms, such as a Fast-SCL algorithm that uses Rate-0, Rate-1 and Rep nodes, which can further reduce the complexity with negligible performance loss.

II. Preliminaries

A. Polar Codes

Polar codes are linear block error correcting codes with recursive structure. A polar code can be described by (N, K), where N is the code length, K is the number of information bits. Let u_1^N be the input vector and x_1^N be the encoded vector. The polar encoding can be done by a matrix multiplication as $x_1^N = u_1^N G^{\otimes n}$, where $G^{\otimes n}$ is constructed by n-th Kronecker product of a 2×2 polarizing matrix $G = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ [1].

A polar code with code rate $R = \frac{K}{N}$ selects K most reliable bits as the information bits. The remaining $N - K$ bits are set as predetermined values, which are called as frozen bits. Since these predetermined values do not affect the error-correction performance of polar codes in symmetric channels [1], they are usually set to 0. After polar encoding, the code vector x_1^N is modulated by binary phase-shift keying (BPSK) modulation which maps $\{0, 1\}$ to $\{+1, -1\}$ and sent through the additive white Gaussian noise (AWGN) channel.

B. Successive Cancellation List Algorithm

To improve the error-correction performance of the SC algorithm for polar codes, the SCL algorithm selects the decoding result with the highest global reliability by decoding multiple paths simultaneously. Instead of doing hard decision directly, the SCL algorithm splits decoding paths at each information bit to consider its both possible values 0 and 1. To control the exponential increase in the complexity, the algorithm restricts the number of decoding paths to L. When the number of decoding paths exceeds L, the SCL algorithm prunes the decoding paths based on the path metric (PM) [10].

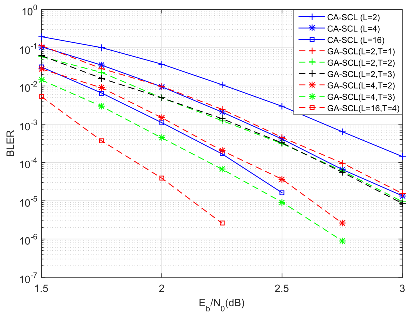

Fig. 1. BLER performance of two algorithms for a (1024,512) polar code.

III. Path Splitting Selecting Strategy Aided Successive Cancellation List Algorithm

A. The Cause for the Errors of SCL Decoding

The SCL algorithm decodes several paths of the SC algorithm simultaneously. Hence, similar to the SC decoder, the errors are also caused by the channel noise or the error propagation in the SCL decoder. In the SC decoder, if we can correct the erroneous bits caused by the channel noise, the error propagation after those bits will be corrected together. Analogously, if we split the decoding paths at the erroneous bits caused by the channel noise, there will be no error propagation on the right path.

In order to verify the above conclusion, we provide the error correcting performance of a genie-aided SCL (GA-SCL) decoder in the ideal situation. First, we do SC decoding to find out the first T erroneous bits caused by the channel noise. If the SC decoder fails, the first erroneous bit must be caused by the channel noise. If we correct it in the process of decoding, the followed erroneous bit will be the second erroneous bit caused by the channel noise in the primitive SC decoding process. In this way, the following erroneous bits caused by the channel noise can be obtained in turn. Then we just split decoding paths at these T bits in the SCL decoder. Note that if there are just $T^* < T$ erroneous bits caused by the channel noise, we just split the decoding paths T^* times.

Fig. 1 shows the block error rate (BLER) performance of two algorithms. As we can see, the GA-SCL algorithm performs better than the CA-SCL algorithm but with less number of path splitting. It means that just splitting decoding paths at the erroneous bits caused by the channel noise can improve the performance of the SCL algorithm and reduce the complexity simultaneously. On the other hand, splitting decoding paths at non-erroneous bits will affect the performance. Because the more decoding paths we split, the less likely we are to retain the correct one when pruning. As a result, the purpose of this letter is to locate the erroneous bits caused by the channel noise accurately and improve the efficiency of path splitting operation in the SCL algorithm. We will introduce two schemes in the following section.

B. The Path Splitting Selecting Strategy Based on the Search Set

A polar code can be divided into multiple sub-blocks. The sub-block which consists of only information bits is called Rate-1 node [4] and the sub-block is a Rep node if and only if its last bit is an information bit [5]. A full binary tree for $N = 2^4$ is shown in Fig. 2, in which the information bits and frozen bits are expressed by black and white leaf nodes, respectively. There are five Rate-1 nodes, which are denoted by their corresponding root nodes $\{A, B, C, D, E\}$ and are colored in black. However, when we consider the Rep nodes, the Rate-1 nodes $\{B, E\}$ are replaced by the Rep nodes $\{F, G\}$ which are colored in red. According to the simulation results in [11], a critical set built by all the first bit of each Rate-1 node can contain more than 99% of all erroneous information bits caused by the channel noise.

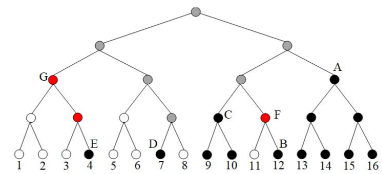

Fig. 2. Full binary tree for a (16,9) polar code.

To improve the efficiency of path splitting operations of the SCL algorithm, we propose our first path splitting selecting strategy based on a search set S which is also constructed by all the first bit of each Rate-1 node when L is small. However, when L increases, the SCL algorithm can keep more paths. Thus, we add the second bit of each Rate-1 node to the search set S when L is large. The path splitting selecting strategy based on the search set (SS) under the SCL algorithm, referred to as the PSS-SS-SCL algorithm, is proposed by splitting the decoding paths at all information bits in the search set.

The PSS-SS strategy can also be used in the simplified SCL algorithms to further reduce the complexity. We introduce the Rate-0, Rate-1 and Rep nodes to the SCL algorithm, which is similar to [6], and call it the Fast-SCL (FSCL) algorithm. Assume that the j-th Rate-1 node and v-th Rep node of l-th path consist of N_o and N_r bits, respectively. For the j-th Rate-1 node, the information bits $u_{j_1,l}^{j_{N_o},l}$ are decoded simultaneously by the log-likelihood ratio (LLR) $\alpha_{x_{j_1,l}}^{x_{j_{N_o},l}}$ of the encoded bits $x_{j_1,l}^{j_{N_o},l}$. According to the hardware-friendly version of f and g operations [6], the absolute value of LLR $|\alpha_{u_{j_1,l}}|$ of $u_{j_1,l}$ for the j-th Rate-1 node can be expressed as

$$|\alpha_{u_{j_1,l}}| = \min_{j_1 \leqslant i \leqslant j_{N_o}} \{|\alpha_{x_{i,l}}|\}. \quad (1)$$

Similarly, the v-th Rep node is also decoded by the LLRs of its encoded bits. The LLR $\alpha_{u_{v_{N_r},l}}$ of $u_{v_{N_r},l}$ for the v-th Rep node can be calculated as

$$\alpha_{u_{v_{N_r},l}} = \sum_{i=v_1}^{v_{N_r}} \alpha_{x_{i,l}}. \quad (2)$$

To generate the search set for the FSCL algorithm, we firstly add the only information bit $u_{v_{N_r},l}$ of the v-th Rep node to S, since the path splitting at the v-th Rep node is to consider the two judgments of $u_{v_{N_r},l}$. Secondly, we replace the first information bit $u_{j_1,l}$ of j-th Rate-1 node by the most unreliable encoded bit $x_{j_p,l}$ which has the minimum absolute value of LLR $|\alpha_{x_{j_p,l}}|$ among $x_{j_1,l}^{j_{N_o},l}$. Note that the location of the encoded bit $x_{j_p,l}$ may be different in each path. When L

Algorithm 1 The PSS-SS-(F)SCL Algorithm

1: Initialization: Build search set S and implement the (F)SCL decoding.
2: While decoding a node, if it belongs to S, then split the decoding paths. Otherwise, do hard decision directly.
3: Select the final decoding path by CRC and output the decoding result $\hat{u}_1^N$.

is large, we should consider two most unreliable encoded bits of each Rate-1 node. Hence, the second unreliable encoded bit $x_{j_q,l}$ of j-th Rate-1 node, which has the second minimum absolute value of LLR $|\alpha_{x_{j_q,l}}|$ among $x_{j_1,l}^{j_{N_o},l}$, is added to the search set. Both PSS-SS strategy based algorithms can be described as Algorithm 1.

C. The Path Splitting Selecting Strategy Based on the Search Set and Decision Function

In practical applications, we cannot really identify the erroneous information bits caused by the channel noise but select them by probability. Since the search set is fixed, it cannot adapt to the changeable channel conditions, the PSS-SS can be further improved by adding a decision function (DF). The improved method uses the DF based on the LLRs to reselect the information bits from the search set S, which is called the PSS-DS scheme. The PSS-DS scheme based SCL algorithm can be shortened as the PSS-DS-SCL algorithm.

If the absolute values of the LLRs of all paths for the i-th bit are small, the probability of two judgments (0 or 1) for this bit is close, which means that the LLRs are unreliable. In this case, we should split the decoding paths at this information bit. We only need to judge whether the bit is reliable or not by the maximum absolute value of the LLRs of all paths. Hence, the DF for the SCL algorithm is

$$D\left(u_i\right) = \max_{1\leqslant l\leqslant m} \left\{|\alpha_{u_{i,l}}|\right\}, \tag{3}$$

where m represents the number of current decoding paths. If $D\left(u_i\right) \leqslant \theta$, the PSS-DS-SCL algorithm splits the decoding paths at u_i. Otherwise, we output the hard decision directly.

Similarly, the PSS-DS strategy can also be used in the FSCL algorithm and the DF needs to be calculated by the LLRs of the encoded bits of each Rate-1 and Rep node. The DF for the v-th Rep node can be expressed as

$$D\left(u_{v_{N_r}}\right) = \max_{1\leqslant l\leqslant m}\{|\alpha_{u_{v_{N_r},l}}|\} = \max_{1\leqslant l\leqslant m}\{|\sum_{i=v_1}^{v_{N_r}} \alpha_{x_{i,l}}|\}, \tag{4}$$

If $D\left(u_{v_{N_r}}\right) \leqslant \theta$, the PSS-DS-FSCL algorithm splits the decoding paths at this Rep node. Otherwise, we do hard decision directly and update the PMs by

$$\mathrm{PM}_{v_{N_r},l} = \mathrm{PM}_{(v_1-1),l} + \sum_{i=v_1}^{v_{N_r}} \frac{(1-sgn(\alpha_{u_{v_{N_r},l}}\alpha_{x_{i,l}}))}{2}|\alpha_{x_{i,l}}|. \tag{5}$$

Moreover, the DF of the j-th Rate-1 node when L is small is

$$D(x_{j_{p^*}}) = \max_{1\leqslant l\leqslant m}\left\{|\alpha_{x_{j_p,l}}|\right\}, \tag{6}$$

Algorithm 2 The PSS-DS-(F)SCL Algorithm

1: Initialization: Build search set S and implement the (F)SCL decoding.
2: While decoding a node, if the node is in the set S, calculate the DF by the LLRs. If the value is smaller than the threshold θ, then split the decoding paths. Otherwise, do hard decision directly.
3: Select the final decoding path by CRC and output the decoding result $\hat{u}_1^N$.

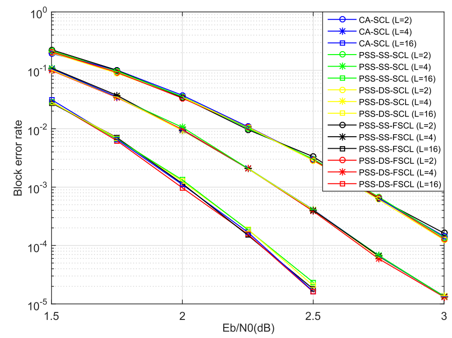

Fig. 3. BLER performance of several algorithms for a (1024,512) polar code.

where $x_{j_p,l}$ is the most unreliable encoded bit of the j-th Rate-1 node at the l-th path. When L is large, we extend the set S by the second most unreliable encoded bit $x_{j_q,l}$ of each Rate-1 node. So we also give the DF of these bits as

$$D(x_{j_{q^*}}) = \max_{1\leqslant l\leqslant m}\left\{|\alpha_{x_{j_q,l}}|\right\} \tag{7}$$

If $D(x_{j_{p^*}}) \leqslant \theta$, the algorithm splits the decoding paths at the encoded bit $x_{j_p,l}$ for the l-th path, where $1 \leqslant l \leqslant m$. Analogously, if $D(x_{j_{q^*}}) \leqslant \theta$, the algorithm splits the decoding paths at the encoded bit $x_{j_q,l}$ for the l-th path. Both PSS-DS strategy based algorithms can be described as Algorithm 2.

IV. Results and Discussion

A. The BLER Performance Comparison

In this section, we compare the BLER performance of several decoding algorithms. Polar codes are concatenated with 16 bits CRC and are construct by Gaussian approximation method [12]. The CRC bits are not considered as information bits when calculating the code rate by $R = \frac{K}{N}$.

Fig. 3 shows the BLER performance of the algorithms for a (1024,512) polar code. The PSS-SS-SCL and PSS-SS-FSCL algorithms perform almost the same as the CA-SCL algorithm with $L = 2, 4$ and have a tiny performance degradation when $L = 16$. The PSS-DS-SCL and PSS-DS-FSCL algorithms perform almost the same as the CA-SCL for all L values. The results prove that the key to the SCL algorithm is not just the path splitting times, but also how to locate the erroneous information bits caused by the channel noise accurately. Table I shows the threshold values θ which are optimized by heuristic experiment search. Since we mainly compare the performance of the algorithms at BLER $= 10^{-5}$, we just show the results at $\frac{E_b}{N_0} \leq 3.00$ dB for $L = 2, 4$ and $\frac{E_b}{N_0} \leq 2.50$ dB for $L = 16$.

TABLE I
THE THRESHOLD VALUES θ OF PSS-DS ALGORITHMS

E_b/N_0	$L = 2$		$L = 4$		$L = 16$	
1.50	23	27	24	19	27	20
1.75	29	29	23	31	33	26
2.00	20	16	39	27	28	36
2.25	30	20	35	28	32	35
2.50	23	19.5	26	26	33	36
2.75	41.5	29	29	28		
3.00	33	38	29	29		

The threshold of the PSS-DS-SCL algorithm is on the left and the threshold of the PSS-DS-FSCL algorithm is on the right for each L.

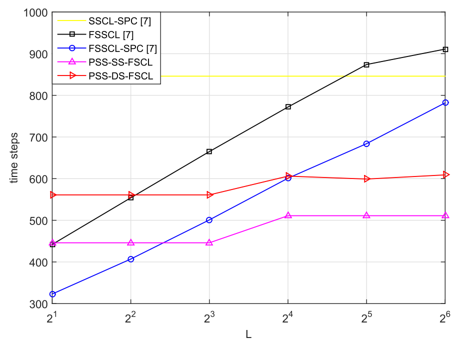

Fig. 4. The time steps of several algorithms for a (1024,512) polar code.

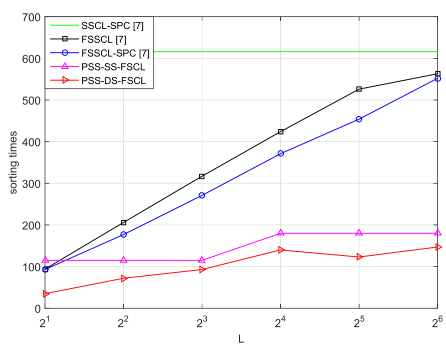

Fig. 5. The sorting times of several algorithms for a (1024,512) polar code.

B. Complexity Comparison

Fig. 4 and Fig. 5 show the time steps and sorting complexity of different algorithms based on the coding structure under $\frac{E_b}{N_0} = 2.25$ dB, respectively. The PSS-SS-FSCL and PSS-DS-FSCL algorithms spend 1, $1 + \max\{N_p, 1\}$ clock cycles on a Rate-1 node and 2, 3 clock cycles on a Rep node, respectively. N_p is the average path splitting times of a Rate-1 node, which is obtained by the Monte Carlo simulation. The time steps are calculated by the sum of the clock cycles required by each node. When L is small, the total number of path splitting is relatively small and spending a clock cycle on the DF brings higher time steps to the PSS-DS-FSCL algorithm. Nevertheless, the DF can reduce the number of path splitting, which keeps the time steps at a relatively low level when L is large. However, the time steps are calculated under the assumption of unlimited resources, which may not be realized in practice. For a more comprehensive comparison, we also compare the sorting complexity of each algorithm based on the sorting times, as sorting the PM values during pruning takes up a large part of the complexity in list decoding algorithms. Each time we do path splitting, if the number of decoding paths exceeds L, we need to sort and prune. So the sorting times approximately equals to the number of path splitting which is obtained by the Monte Carlo simulation. Fig. 5 shows that the PSS-DS-FSCL algorithm has the lowest sorting complexity compared with other simplified SCL algorithms. The reason is that the DF of the PSS-DS-FSCL algorithm can greatly improve the efficiency of path splitting operation.

V. CONCLUSION

In this letter, we firstly study the reasons why the SCL decoder fails. Then we proposed the PSS strategy based on two schemes to improve the efficiency of the path splitting operation. Both PSS-SS and PSS-DS strategies reduce the number of path splitting. In addition, the PSS-DS-SCL algorithm and the PSS-DS-FSCL algorithm can reduce the complexity with almost no performance degradation. Hence, the proposed method can be used to decode the polar code in practical applications, which points out a new direction for the research on the simplified SCL algorithm.

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] B. Li, H. Shen, and D. Tse, “An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check,” *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2044–2047, Dec. 2012.
[4] A. Alamdar-Yazdi and F. R. Kschischang, “A simplified successive-cancellation decoder for polar codes,” *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011.
[5] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, “Fast polar decoders: Algorithm and implementation,” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, May 2014.
[6] S. A. Hashemi, C. Condo, and W. J. Gross, “A fast polar code list decoder architecture based on sphere decoding,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2368–2380, Dec. 2016.
[7] S. A. Hashemi, C. Condo, and W. J. Gross, “Fast and flexible successive-cancellation list decoders for polar codes,” *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, Nov. 2017.
[8] B. Li, H. Shen, and K. Chen, “A decision aided parallel SC-list decoder for polar codes,” *CoRR*, vol. abs/1506.02955, Jun. 2015. [Online]. Available: http://arxiv.org/abs/1506.02955
[9] 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.
[10] 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.
[11] Z. Zhang, K. Qin, L. Zhang, H. Zhang, and G. T. Chen, “Progressive bit-flipping decoding of polar codes over layered critical sets,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.
[12] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.