# CS-FSCL decoding algorithm of polar codes based on critical sets*

**YUAN Jianguo**, ZHANG Fengguo, HE Jingjie, and PANG Yu**
*School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

(Received 8 October 2023; Revised 16 November 2023)


In order to reduce the number of redundant candidate codewords generated by the fast successive cancellation list (FSCL) decoding algorithm for polar codes, a simplified FSCL decoding algorithm based on critical sets (CS-FSCL) of polar codes is proposed. The algorithm utilizes the number of information bits belonging to the CS in the special nodes, such as Rate-1 node, repetition (REP) node and single-parity-check (SPC) node, to constrain the number of the path splitting and avoid the generation of unnecessary candidate codewords, and thus the latency and computational complexity are reduced. Besides, the algorithm only flips the bits corresponding to the smaller log-likelihood ratio (LLR) values to generate the sub-maximum likelihood (sub-ML) decoding codewords and ensure the decoding performance. Simulation results show that for polar codes with the code length of 1 024, the code rates of 1/4, 1/2 and 3/4, the proposed CS-FSCL algorithm, compared with the conventional FSCL decoding algorithm, can achieve the same decoding performance, but reduce the latency and computational complexity at different list sizes. Specifically, under the list size of $L$=8, the code rates of $R$=1/2 and $R$=1/4, the latency is reduced by 33% and 13% and the computational complexity is reduced by 55% and 50%, respectively.

**Document code:** A **Article ID:** 1673-1905(2024)07-0418-6
**DOI** https://doi.org/10.1007/s11801-024-3214-2

Polar codes with its successive cancellation (SC) decoding algorithm have been proven to achieve the channel capacity[1]. To reduce the gap of error correction performance between the SC decoding and the maximum-likelihood (ML) decoding for polar codes, the successive cancellation list (SCL) decoding algorithm[2,3] and the cyclic redundancy check aided SCL (CA-SCL) decoding algorithms[4,5] were proposed by retaining the $L$ best paths in the decoding process.

However, the bit-by-bit decoding process of the SCL decoder leads to high latency and low throughput, and it produces redundant computation for the frozen bits. Hence, the fast successive cancellation list (FSCL) decoding algorithm[6] proposed in 2017 defines four special sub-codes (named as Rate-0, Rate-1, repetition (REP) and single-parity-check (SPC) nodes) and directly obtains the hard decision lists of the codewords for these nodes without calculating the log-likelihood ratio (LLR) of every source bit. It has been proven in Ref.[7] that the FSCL decoding does not compromise error correction performance compared to the conventional SCL decoding.

To further meet the low-latency decoding demands of 5G systems, more efforts have been taken. On one hand, enhance the general identification and the parallel decoding of nodes. Ref.[8] and Ref.[9] proposed five new nodes (Type-I, Type-II, Type-III, Type-IV and Type-V nodes) and their SC and SCL decoding schemes to improve the decoding speed at long code lengths. The sequence repetition (SR) node and its adaptive path splitting strategy introduced in Ref.[10] and Ref.[11] expedites the decoding speed of non-specific structure nodes and achieves the same decoding performance as the conventional FSCL decoders.

On the other hand, reduce the unnecessary path splitting. Ref.[12] proposes the path splitting selection strategy combined with the fixed search set or the dynamic search set under channel noise, reducing the complexity for Rate-1 and REP nodes with little or no performance degradation, but it requires computing the threshold value by LLRs. Ref.[13] proposes an adaptive path splitting algorithm that can directly remove the paths with large path metric (PM) values before path splitting. Ref.[14] reduces the number of path splitting for reliable bits, delivering almost the same performance as the conventional FSCL decoding for higher code rates,

* This work has been supported by the National Natural Science Foundation of China (Nos.U21A20447 and No.61971079).
** YUAN Jianguo is a professor at School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications. He received his Ph.D. degree in 2007 from Chongqing University. His research interests are mainly in the coding and decoding technologies for polar codes and LDPC codes as well as OFDM technologies. E-mail: yuanjg@cqupt.edu.cn

but not working well for lower code rates. Ref.[15] uses an offline search method to generate the minimum-combination (MC) sets, which contains bits to be flipped and is related to list sizes, reducing the decoding latency of Rate-1 node to 1, without, unfortunately, reducing the decoding latency of other nodes. In summary, this paper combines the idea of eliminating the path splitting of reliable bits as in Ref.[14] and further improves the decoding process of Rate-1, SPC, and REP nodes to reduce the latency and the computational complexity and maintain the decoding performance across various code rates for polar codes.

Polar codes are constructed based on the symmetric capacity of polarized channels. For a polar code with the length of $N$ and $K$ information bits, the less reliable $N-K$ sub-channels are employed to transmit frozen bits (typically set to 0), while the remaining reliable $K$ sub-channels are employed to transmit information bits. Then the source sequence $\boldsymbol{u}$ is polar encoded into a codeword sequence $\boldsymbol{x}=\boldsymbol{u}\boldsymbol{F}^{\otimes n}$, where $\boldsymbol{F}^{\otimes n}$ is the $n$th Kronecker product of $\boldsymbol{F}=\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}$. The information bits can be presented directly in the source sequence $\boldsymbol{u}$ or in the codeword sequence $\boldsymbol{x}$, depending on whether non-systematic coding or systematic coding is adopted.

After the codeword sequence is transmitted over the channel, the FSCL decoder uses the LLR sequence of the received signal to decode.

A polar code binary tree with code length of $N$=16 is shown in Fig.1, where black dots denote information bits and white dots denote frozen bits. Each node in the hierarchy $S$ contains $N_{\rm v}=2^S$ bits. When $S$=0, leaf nodes denote the source sequence $\boldsymbol{u}=\{u_0,u_1,...,u_{N\text{-}1}\}$.

The parent node of length $N_{\rm v}$ passes the LLR sequence $\boldsymbol{\alpha}=\{\alpha_0,\alpha_1,...,\alpha_{N_{\rm v}\text{-}1}\}$ to the left and right children, denoted as $\boldsymbol{\alpha}^{\rm l}=\left\{\alpha_0^{\rm l},\ \alpha_1^{\rm l},...,\ \alpha_{\frac{N_{\rm v}}{2}-1}^{\rm l}\right\}$ and $\boldsymbol{\alpha}^{\rm r}=\left\{\alpha_0^{\rm r},\alpha_1^{\rm r},...,\alpha_{\frac{N_{\rm v}}{2}-1}^{\rm r}\right\}$, respectively, which can be computed as

$$\alpha_i^{\rm r}=\operatorname{sgn}(\alpha_i)\operatorname{sgn}\left(\alpha_{i+\frac{N_{\rm v}}{2}}\right)\min\left(|\alpha_i|,\left|\alpha_{i+\frac{N_{\rm v}}{2}}\right|\right), \tag{1}$$

$$\alpha_i^{\rm r}=\alpha_{i+\frac{N_{\rm v}}{2}}+(1-2\beta_0^{\rm l})\alpha_i. \tag{2}$$

The left and right children return their respective hard decision of bits, i.e., $\boldsymbol{\beta}^{\rm l}=\left\{\beta_0^{\rm l},\ \beta_1^{\rm l},...,\ \beta_{\frac{N_{\rm v}}{2}-1}^{\rm l}\right\}$ and $\boldsymbol{\beta}^{\rm r}=\left\{\beta_0^{\rm r},\beta_1^{\rm r},...,\beta_{\frac{N_{\rm v}}{2}-1}^{\rm r}\right\}$, to the parent node to get the codeword sequence $\boldsymbol{\beta}=\{\beta_0,\beta_1,...,\beta_{N_{\rm v}\text{-}1}\}$, according to

$$\beta_i=\begin{cases}\beta_i^{\rm l}\oplus\beta_i^{\rm r}, & i<N_{\rm v}/2\\ \beta_{i\text{-}N_{\rm v}/2}^{\rm r}, & \text{else}\end{cases}, \tag{3}$$

where the symbol $\oplus$ is the bitwise XOR operation.

The four special structures of nodes defined by the conventional FSCL decoder are the Rate-0, Rate-1, REP and SPC nodes, as shown in Fig.1. The FSCL decoder estimates all possible codeword sequences for these nodes directly based on their received LLR sequence. Let the binary vector $\boldsymbol{v}=\{v_0,v_1,...,v_{N_{\rm v}}\}$ denote the structure of the special nodes ($v_i$=0 represents the frozen bit and $v_i$=1 represents the information bit), the following are the structure representation of the four nodes and their decoding process.

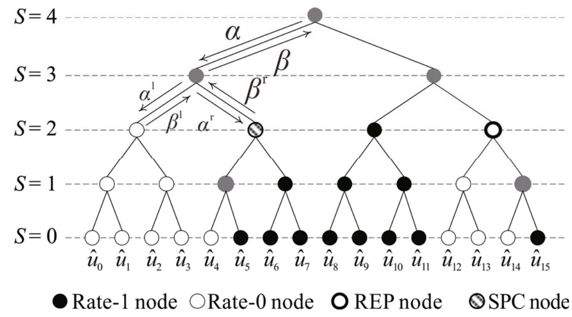



**Fig.1 Binary tree for a (16,8) polar code**

The Rate-0 node is composed entirely of frozen bits, so no path splitting occurs, but the PM still needs to be changed as

$$P{=}P_1+\sum_{i=1}^{n}\ln(1+{\rm e}^{-\alpha_i^{\rm l}}). \tag{4}$$

The REP node contains only one information bit located at the last position, so the number of path splitting is 1, generating an all-zero and an all-one codewords. The corresponding PMs are updated as

$$P_1^0{=}P_1+\sum_{i=1}^{N_{\rm v}}\ln(1+{\rm e}^{\text{-}\alpha_i^{\rm l}}),$$

$$P_1^1{=}P_1+\sum_{i=1}^{N_{\rm v}}\ln(1+{\rm e}^{\alpha_i^{\rm l}}). \tag{5}$$

All source bits of the Rate-1 node are information bits. The FSCL decoder first performs the hard decision on the received LLR sequence of each path to obtain the ML codeword $\{x_1^{\rm l},x_2^{\rm l},...,x_{N_{\rm v}}^{\rm l}\}$, according to

$$x_i^{\rm l}=\begin{cases}0, & \alpha_i^{\rm l}\geq 0\\ 1, & \text{else}\end{cases}. \tag{6}$$

Then the absolute values of LLR sequence is arranged in ascending order as $\{|\alpha_1|,|\alpha_2|,...,|\alpha_{N_{\rm v}}|\}$, and the remaining $L-1$ sub-ML codewords are obtained by flipping the codeword bits in order from the smallest to the largest absolute values of the LLR sequence. The PM of the corresponding path after each splitting is updated as

$$P_1^0=P_1,$$

$$P_1^1=P_1+|\alpha_i^1|, \tag{7}$$

where $P_1$ is the PM before each splitting, and the upper bound on the number of path splitting of Rate-1 node is identified as $\min(L-1,N_{\rm v})$ [7].

The SPC node has codewords that satisfy even-parity. Firstly, the hard decision is performed according to Eq.(6). If the codeword meets the even-check, the codeword is the ML codeword, and let $q=0$ ; otherwise, flip the bit corresponding to the smallest of the absolute LLR sequence $|\alpha_1|$ to obtain the ML codeword, and let $q=1$ . Then, the bits are flipped in ascending order of their corresponding absolute LLR values to get the remaining $L-1$ sub-ML codewords that meet the even check. The corresponding PM is updated according to Eq.(8). The upper bound on the number of the path splitting is $\min(L, N_v)$ [7].

$$P_1^0 = P_1, \quad q = q,$$
$$P_1^1 = P_1 + |\alpha_i^1| + (1-2q)|\alpha_i^1|, \quad q = \bar{q}. \tag{8}$$

Based on the above analyses, the FSCL decoder performs the path splitting and preserves two possible decoding results of "0" and "1" for each non-frozen bit even if its corresponding sub-channel is sufficiently reliable, and it will affect the decoding latency and the computational complexity by generating unnecessary path splitting and redundant candidate codewords. Therefore, retain only one result for those reliable bits can reduce the operations of path copying, sorting, and deleting and will not significantly reduce the error correction performance meanwhile.

Fig.2 depicts the polarization of the Rate-1 node of length $N$=4, where the source sequence is composed of four information bits $u_1$, $u_2$, $u_3$, $u_4$, and the codeword is represented as $\boldsymbol{x}=\{x_1,x_2,x_3,x_4\}$. The channels transmitting $u_1$ and $u_2$ are reliable, while the channels transmitting $u_3$ and $u_4$ are unreliable. The number of unreliable channels is recorded as $k$ (in this case, $k$=2). Assuming that only perform the path splitting on bits corresponding to unreliable channels, without considering the list size, then the number of decoded candidate codewords can be reduced from the original $2^{N_v}=8$ to $2^k=4$ , thus achieving the purpose of complexity reduction.

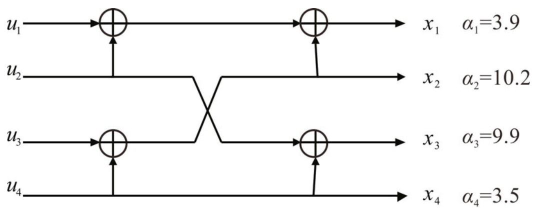



**Fig.2 Rate-1 node of length *N*=4**

After the source sequence is encoded into a node, it is worth noting that the absolute LLR value $a_i$ of the codeword bit $x_i$, which corresponds to source information bit $u_i$ transmitted by unreliable channels, may not fall within the smaller $k$ values in the absolute value of the received LLR sequence. Take Fig.2 as an example: the LLR $a_3$ of the codeword bit $x_3$ corresponding to the source bit $u_3$ is 9.9, which is larger than the LLR $a_1$ of $x_1$ and the LLR $a_4$ of $x_4$. If only the codeword bits corresponding to unreliable channels (in this case, $x_3$ and $x_4$) are flipped, the generated $2^k-1$ codewords are not sub-ML codewords and it will lead to a decoding error.

Hence, the FSCL decoding algorithm based on critical sets (CS-FSCL) proposed in this paper, regardless of whether the flipped codeword bits correspond to frozen or reliable bits, only considers the number of unreliable information bits, i.e., $k$, to limit the number of flips and it flips codeword bits corresponding to the $k$ smallest LLR values.

Furthermore, the CS containing unreliable bits is introduced to flexibly control the number $k$ of the flipped bits in this paper. The construction method is as follows: Take the polarization weight (PW) method of the 5G standard as an example for estimating channel reliability. Information bits are transmitted on sub-channels with higher PW values, and the reliable sub-channel index set is denoted as $A$. Then the parameter $\lambda$ ($0<\lambda<1$) is introduced to select the least reliable $\lambda|A|$ bits from the set $A$ to form the CS. The value of $\lambda$ is not constant and varies with different code lengths and code rates. Smaller value of $\lambda$ is preferable to minimize the decoding latency and maintain the same block error rate (*BLER*) as the conventional FSCL decoding algorithm simultaneously.

Fig.3 shows the number of redundant nodes (nodes containing information bits outside the CS and the path splitting can be eliminated) among all Rate-1, REP, and SPC nodes at $\lambda$=0.8, *SNR*=2.5 dB, with polar codes of code length of $N$=1 024, code rate $R$=3/4, 1/2, and 1/4, respectively. As observed in Fig.3, the majority of redundant nodes are Rate-1 and SPC nodes, resulting from the polarization phenomenon of polar codes. For instance, at $R$=3/4, Rate-1 nodes account for 45% and SPC nodes account for 15%. Furthermore, the proportion of redundant Rate-1 nodes increases as the code rate increases.

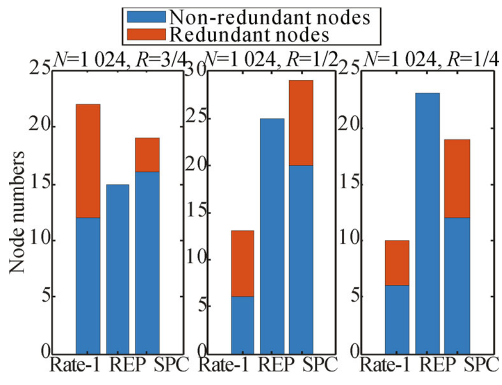



**Fig.3 Distribution of redundant nodes**

In this paper, a simplified CS-FSCL decoding algorithm of polar codes based on CSs is proposed as follows, primarily focusing on improving the decoding schemes of four special nodes: Rate-0, Rate-1, REP and SPC nodes.

Since the Rate-0 node consists of fully frozen bits, there is no possibility of path splitting. The PM requires changes according to Eq.(4).

The REP node contains only one information bit, so there are only two cases. Case 1: If the last information bit of the REP node belongs to the CS, perform path splitting and produce an all-0 codeword and an all-1 codeword, and the PM is updated according to Eq.(5). Case 2: If the last information bit does not belong to CS, a hard decision is performed based on its corresponding LLR value $\alpha_i$, and produce either an all-0 codeword (when $\alpha_i<0$) or an all-1 codeword (when $\alpha_i\geq 0$). The PM is updated as

$$P=\begin{cases} P_1+\sum_{i=1}^{N_\mathrm{v}}\ln(1+\mathrm{e}^{-\alpha_i^1}), & \alpha_i\geq 0 \\ P_1+\sum_{i=1}^{N_\mathrm{v}}\ln(1+\mathrm{e}^{\alpha_i^1}), & \text{else} \end{cases}, \tag{9}$$

where $P_1$ is the PM of the SC decoder before each path splitting.

For the Rate-1 node, the results of hard decision based on the LLR values is used as the ML decoding codeword. Then, the number of information bits in the node belonging to the CS is calculated as $k=\left|\mathrm{CS}\cap \boldsymbol{u}_1^{N_\mathrm{v}}\right|$ $(k\leq N_\mathrm{v})$. After sorting the absolute LLR values of all codeword bits in ascending order, sequentially flip the codeword bits corresponding to the smaller LLR $k$ values, i.e., $\{|\alpha_1|,|\alpha_2|,...,|\alpha_k|\}$. The PM corresponding to the $i$th splitting is calculated as

$$P_1^0=P_1,$$
$$P_1^1=P_1+\left|\alpha_i^1\right|(i=1,2,...,k). \tag{10}$$

Due to the path reduction principle of the FSCL decoding algorithm, splitting only $k$ bits corresponding to the smaller LLR values ensures that the generated codewords are the sub-ML codewords with $2^k$ smallest PM values among the original $2^{N_\mathrm{v}}$ $(N_\mathrm{v}<L)$ candidate codewords of the conventional FSCL decoders, which makes the algorithm not incur a decoding performance loss. When $2^k<L$, each decoding path will only generate $2^k$ different codewords, and when $2^k\geq L$, only $L$ different codewords will be generated. The upper limit of the number of path splitting is reduced to $\min(L-1,k)$.

For the SPC node, the proposed algorithm follows the same process as the conventional FSCL decoder to obtain the ML codeword. Then, based on the ML codeword, it selectively flips only the codeword bits corresponding to the $k$ smallest absolute LLR values ($\{|\alpha_1|,|\alpha_2|,...,|\alpha_{k+1}|\}$). The maximum number of flips is limit to $\min(L,k)$, and the PM of each path splitting is calculated as

$$P_1^0=P_1,\quad q=q,$$
$$P_1^1=P_1+\left|\alpha_1^1\right|+(1-2q)\left|\alpha_i^1\right|,\quad q=\overline{q} \text{ and } i=2,...,k+1. \tag{11}$$

In summary, the strategy of the simplified CS-FSCL decoding algorithm can be described as Tab.1.

In order to verify the superiority of the CS-FSCL decoding algorithm proposed in this paper, the complexity and *BLER* performance are simulated and analyzed in this section.

**Tab.1 CS-FSCL decoding algorithm**

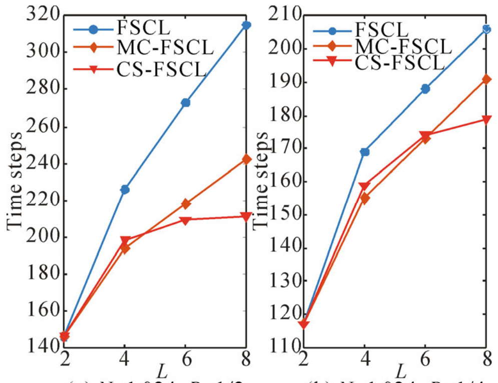
| CS-FSCL decoding algorithm |
| --- |
| Inputs: received LLRs, CS |
| Output: decoding result $\hat{\boldsymbol{u}}$ |
| for each node |
| if Rate-0 node |
| $\hat{\boldsymbol{u}}=0$ |
| PM is updated as Eq.(4) |
| else if REP node |
| if the last information bit $u_i\in\mathrm{CS}$ then |
| generate two candidate codewords $\hat{\boldsymbol{u}}=0$ and $\hat{\boldsymbol{u}}=1$, the PM is updated as Eq.(5) |
| else |
| generate one candidate codewords $\hat{\boldsymbol{u}}=0$ or $\hat{\boldsymbol{u}}=1$, the PM is updated as Eq.(9) |
| else if Rate-1 node |
| obtain the ML codeword by hard decision |
| Calculate $k$, set the number of flips to $\min(L-1,\ k)$ and get the remaining sub-ML codewords, the PM is updated as Eq.(10) |
| else SPC node |
| obtain the ML codeword, calculate $k$, set the number of flips to $\min(L-1,\ k)$ and get the remaining sub-ML codewords, the PM is updated as Eq.(11) |
| end if |
| end for |
| Output decoding candidate codewords $\hat{\boldsymbol{u}}$ |

The time step of different nodes is calculated by the approach in Ref.[7]. Tab.2 summarizes the time step of four nodes with length $N_\mathrm{v}$ of the conventional FSCL decoding algorithm in Ref.[7], the MC-FSCL decoding algorithm in Ref.[15], and the CS-FSCL decoding algorithm proposed in this paper. The time step is highly related to the number of path splitting and each split will consume one time step[7]. In the CS-FSCL decoding algorithm, the number of path splitting in the Rate-1 node and the SPC node is $\min(L-1,k)$ and $\min(L,k)$, respectively and the SPC node spends one extra time step to get the ML codeword. In fact, if the CS-FSCL decoding algorithm has the same number of unreliable channel indexes as the simplified path split (SPS)-FSCL decoding algorithm proposed in Ref.[14], the number of path splitting and the time step are also the same.

**Tab.2 Time steps of four nodes under list size of *L***

| Algorithm | FSCL-SSCL-SPC | MC-FSCL | CS-FSCL |
| --- | --- | --- | --- |
| Rate-0 | 1 | 1 | 1 |
| REP | 2 | 2 | 1 or 2 |
| Rate-1 | $\min(L-1,N_\mathrm{v})$ | 1 | $\min(L-1,k)$ |
| SPC | $\min(L,N_\mathrm{v})+1$ | $\min(L,N_\mathrm{v})+1$ | $\min(L,k)+1$ |

To further verify the algorithm's latency, this paper conducts a simulation on polar codes with code length of $N$=1 024, code rates $R$=1/2 and $R$=1/4, and CRC-16 with a generating polynomial $g_{16}(x)=x^{16}+x^{15}+x^2+1$. These polar codes are modulated by binary phase shift keying and then transmitted over additive white Gaussian noise channel. The latency of the CS-FSCL decoding algorithm, the conventional FSCL decoding algorithm, and the MC-FSCL decoding algorithm for different list lengths is depicted in Fig.4, where $\lambda$=0.3 at $R$=1/2 and $\lambda$=0.5 at $R$=1/4. From Fig.4, it can be observed that the CS-FSCL decoding algorithm has reduced latency compared with the conventional FSCL decoding algorithm across different code rates and various list sizes. When $L$=8, the CS-FSCL algorithm has the lowest latency among three algorithms with a reduction of 33% (at $R$=1/2) and 13% (at $R$=1/4) compared to the conventional FSCL decoding algorithm. Notably, the latency of the CS-FSCL decoding algorithm increases with the list size $L$, but will remain constant when the list size exceeds the node length.

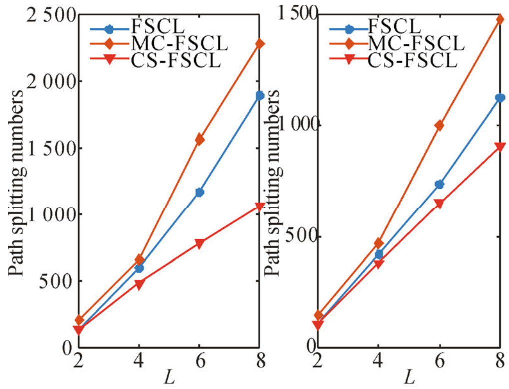



(a) $N$=1 024, $R$=1/2 (b) $N$=1 024, $R$=1/4

**Fig.4 Time steps of the CS-FSCL, the conventional FSCL and the MC-FSCL for different list lengths**

Additionally, the computational complexity is also affected by the number of path copying, sorting and deleting during the path splitting. Therefore, an experiment on the number of path splitting for the three algorithms is also conducted in this paper as shown in Fig.5. It is evident that the computational complexity of the CS-FSCL is the least, with a reduction of 53% (at $R$=1/2) and 38% (at $R$=1/4) when the list size is 8, compared to the conventional FSCL decoding algorithm, and a reduction of 44% (at $R$=1/2) and 18% (at $R$=1/4) compared to the MC-FSCL decoding algorithm.

In this paper, a simulation comparison analysis on the error correction performance is also performed for the CS-FSCL decoding algorithm of different code rates such as $R$=1/2 ($\lambda$=0.3), $R$=1/4 ($\lambda$=0.5) and $R$=3/4 ($\lambda$=0.4) and different list sizes, as shown in Fig.6 and Fig.7. Other simulation conditions are the same as referred above. As observed in Fig.6, the CS-FSCL decoding algorithm maintains the same decoding performance as the conventional FSCL decoding algorithm, when list size is $L$=4 and $L$=8, and code rate is $R$=1/2. From Fig.7, it can be seen that the CS-FSCL decoding algorithm has no performance loss compared with the conventional FSCL decoding algorithm at both lower and higher code rates. In addition, the CS-FSCL decoding algorithm achieves a better error correction performance than the SPS-FSCL decoding algorithm, while their computational complexity is the same. Especially, when $L$=8, $R$=1/4 and $BLER$=$10^{-4}$, the CS-FSCL decoding algorithm improves by 0.13 dB compared to the SPS-FSCL decoding algorithm.




(a) $N$=1 024, $R$=1/2 (b) $N$=1 024, $R$=1/4

**Fig.5 Number of path splitting for the CS-FSCL, the conventional FSCL, and the MC-FSCL at code rates of *R*=1/2 and 1/4, respectively**

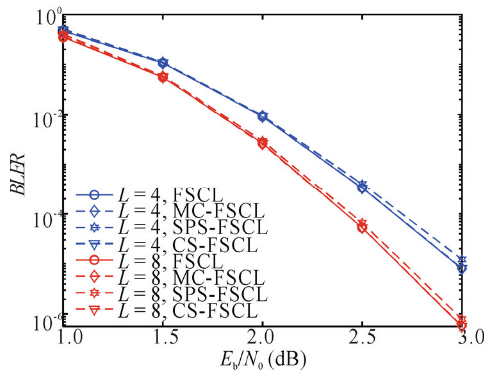



**Fig.6 *BLER* performance comparison for the CS-FSCL, the conventional FSCL, the SPS-FSCL, and the MC-FSCL with list sizes of *L*=4 and 8, respectively**

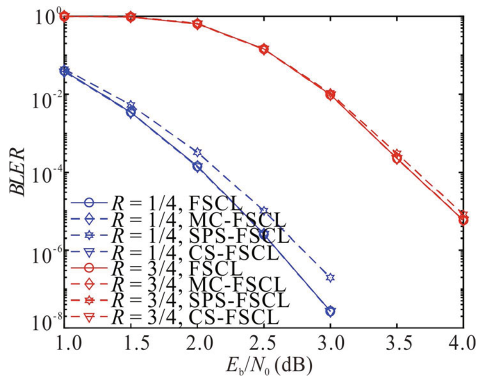



**Fig.7 *BLER* performance comparison for the CS-FSCL, the conventional FSCL, the SPS-FSCL, and the MC-FSCL at code rates of *R*=1/4 and 3/4, respectively**

The proposed CS-FSCL decoding algorithm in this paper utilizes the number of the information bits belonging to the CS in the special nodes to constrain the number of the path splitting and flips the codeword bits corresponding to the smaller LLR values to obtain the ML codewords and sub-ML codewords. Simulation results show that the proposed CS-FSCL decoding algorithm, compared with the conventional FSCL decoding algorithm, can reduce the computational complexity without any degradation in the error correction performance.

## Ethics declarations

## Conflicts of interest

The authors declare no conflict of interest.

## References

[1] CHEN F T, ZHAO H M, SHI D, et al. Decoding algorithm of polarization code with low time complexity[J]. Journal of Chongqing University of Posts and Telecommunications (natural science edition), 2021, 33(4): 571-576. (in Chinese)

[2] MILOSLAVSKAYA V, VUCETIC B, LI Y H, et al. Recursive design of precoded polar codes for SCL decoding[J]. IEEE transactions on communications, 2021, 69(12): 7945-7959.

[3] LIU W, DUAN H G. Adaptive successive cancellation list bit-flip decoding of polar codes[J]. Journal of Chongqing University of Posts and Telecommunications (natural science edition), 2021, 33(1): 87-93. (in Chinese)

[4] SHEN Y F, BALATSOUKAS-STIMMING A, YOU X H, et al. Dynamic SCL decoder with path-flipping for 5G polar codes[J]. IEEE wireless communications letters, 2022, 11(2): 391-395.

[5] YUAN J G, ZHANG R, ZHANG F G, et al. Novel coding algorithm of the CRC-aided PC-polar codes[J]. Journal of Chongqing University of Posts and Telecommunications (natural science edition), 2022, 34(6): 929-934. (in Chinese)

[6] SARKIS G, GIARD P, VARDY A, et al. Fast list decoders for polar codes[J]. IEEE journal on selected areas in communications, 2016, 34(2): 318-328.

[7] HASHEMI S A, CONDO C, GROSS W J. Fast and flexible successive-cancellation list decoders for polar codes[J]. IEEE transactions on signal processing, 2017, 65(21): 5756-5769.

[8] HANIF M, ARDAKANI M. Fast successive-cancellation decoding of polar codes: identification and decoding of new nodes[J]. IEEE communications letters, 2017, 21(11): 2360-2363.

[9] ARDAKANI M H, HANIF M, ARDAKANI M. Fast successive-cancellation-based decoders of polar codes[J]. IEEE communications letters, 2019, 67(7): 4562-4574.

[10] ZHENG H T, HASHEMI S A, BALATSOUKAS S A, et al. Threshold-based fast successive-cancellation decoding of polar codes[J]. IEEE transactions on communications, 2021, 69(6): 3541-3555.

[11] SHEN Y F, KRISTENSEN A T, BALATSOUKAS S A, et al. Fast sequence repetition node-based successive cancellation list decoding for polar codes[C]//IEEE 9th International Conference on Communications, July 27-29, 2022, Seoul, Korea. New York: IEEE, 2022: 116-122.

[12] GAO C, LIU R, DAI B, et al. Path splitting selecting strategy-aided successive cancellation list algorithm for polar codes[J]. IEEE communications letters, 2019, 23(3): 422-425.

[13] WANG L, ZHANG Z, HU H. Adaptive fast simplified successive cancellation list polar decoding based on path selecting[C]//2020 IEEE/CIC International Conference on Communications in China, August 9-11, 2020, Chongqing, China. New York: IEEE, 2020: 959-963.

[14] PENG Y L, WU H H, CHEN P P, et al. Simplified path split strategy for SCL decoding algorithm with lower latency[C]//2022 Wireless Telecommunications Symposium, April 6-8, 2022, Pomona, CA, USA. New York: IEEE, 2022: 1-6.

[15] ZHAO Y, YIN Z, WU Z, et al. Minimum-combinations set-based rate-1 decoder for fast list decoding of polar codes[J]. IEEE communications letters, 2021, 25(10): 3185-3189.