SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2021, Vol. 64 192306:1–192306:8
https://doi.org/10.1007/s11432-019-2801-6

Belief propagation list bit-flip decoder for polar codes

Yuyu YANG[1], Yaoyue HU[1], Zhiwen PAN[1,2*], Nan LIU[1] & Shenjie Xia[3]

[1]*National Mobile Communications Research Laboratory, Southeast University, Nanjing* 210096*, China;*
[2]*Purple Mountain Laboratories, Nanjing* 211100*, China;*
[3]*Huawei Technologies Co. Ltd., Shanghai* 201206*, China*

Received 5 August 2019/Accepted 18 February 2020/Published online 18 August 2021

Abstract Belief propagation (BP) decoder is a low decoding latency algorithm for polar codes. However, the block error rate (BLER) performance of the BP decoder is inferior to the cyclic redundancy check (CRC) aided successive cancellation list (CA-SCL) decoder with medium list size for polar codes. Thus, in this paper, we introduce a bit-flip method into the belief propagation list (BPL) decoder and propose a BPL bit-flip (BPLF) decoder to improve the BLER performance of BP-based decoder for polar codes. The proposed decoder is based on a CRC-aided belief propagation list (CA-BPL) decoder, and the error-prone bits are obtained from the decoded result to construct flip bits set (FBS). The bits in FBS are flipped to obtain the correct decoding result. Moreover, the simulation results show that the BLER of the proposed BPLF decoder is significantly superior to that of the CA-BPL despite having the same list size, e.g., there is 0.75 dB gain at BLER $= 10^{-4}$ with list size 64. Furthermore, the decoding complexity and latency of the proposed BPLF decoder are only slightly higher than those of CA-BPL decoder with the same list size in medium to high signal-to-noise ratio (SNR) region.

Keywords polar codes, belief propagation, bit-flip, list decoder

Citation Yang Y Y, Hu Y Y, Pan Z W, et al. Belief propagation list bit-flip decoder for polar codes. Sci China Inf Sci, 2021, 64(9): 192306, https://doi.org/10.1007/s11432-019-2801-6

1 Introduction

Polar codes are the first family of codes proven to be capable of achieving channel capacity in binary-input discrete memoryless channels (BDMC) [1]. Successive cancellation (SC) decoding and belief propagation (BP) decoding are two typical methods to decode polar codes. The SC decoding algorithm has low computational complexity, but its seriality leads to high decoding latency. Furthermore, for practical code length, the block error rate (BLER) performance of polar codes under the SC decoding is inferior to that of other state-of-the-art codes [2] owing to insufficient polarization phenomenon. To reduce the decoding latency and complexity of the conventional SC decoder, a simplified SC was proposed in [3]. Then, a cyclic redundancy check (CRC)-aided successive cancellation list (CA-SCL) decoder was introduced in [4] to further improve the BLER performance of polar codes in the practical code length region. With the help of CRC and list decoding, the CA-SCL decoder with moderate list size can achieve better BLER performance than other modern codes. Further, the SC-flip algorithm was introduced in [5] to improve the BLER performance of the SC decoder. If SC decoder fails, the bits which are likely to be the first error bits are chosen from those unreliable bits and flipped to get the correct result. For the bit-flip method, one of the most important procedures is to identify unreliable bits. Authors in [6] proposed the concept of a critical set (CS). The SCL bit-flip (SCLF) decoder was introduced in [7] to improve the BLER performance of the CA-SCL decoder. In SCLF, a revised critical set (RCS), which is based on CS while considering the decoding result of CA-SCL decoder is employed, to find the unreliable bits.

The SC-based decoders mentioned earlier decode information sequence bit-by-bit, which cannot be calculated in parallel. Thus, they cause long decoding latency, whereas the decoding procedure of the BP

* Corresponding author (email: pzw@seu.edu.cn)

decoder for polar codes is inherently parallel [8, 9]. With higher computational complexity, BP decoder has shorter decoding latency than SC-based decoders, making it suitable for the scenarios that require high throughput. But the BLER performance of the traditional BP decoder is inferior to SC decoder. To further reduce the decoding complexity and latency of BP decoder, different early termination algorithms were designed in [10–12]. When the condition of the early stop is satisfied, the iteration of the BP decoder is terminated. To improve the BLER performance of the BP decoder, a hybrid BP-SC decoder was proposed in [13]. If the BP decoder cannot meet the early stop criterion based on a generator matrix before it reaches maximum iterations, the SC decoder is employed. In [14], the decoding errors were classified into three types, and different post-processing methods based on the perturbation of the log-likelihood ratios (LLRs) were employed for each different kind of error, making the iteration result converge to correct codeword. Inspired by the phenomenon that the BP decoder failed on a one-factor graph may obtain a correct result on another factor graph, more than one-factor graph is employed to improve the BLER performance of the BP decoder in [15, 16]. In [17], the authors went further and proposed the belief propagation list (BPL) algorithm. Several BP decoders which are based on different factor graphs are employed parallel in BPL decoder. With higher computational and hardware complexity, the BLER performance of the BPL decoder is much better compared with the BP decoder. However, the BLER performances of the BP-based decoders mentioned earlier are still not comparable to that of CA-SCL decoder with medium list size; hence, it is crucial to further improve the BLER performance of the BP-based decoder.

In this paper, we introduce the bit-flip method into the BPL decoding scheme and propose belief propagation list bit-flip (BPLF) decoder to narrow the BLER performance gap between the BP-based decoder and CA-SCL decoder. When the CA-BPL decoder fails, we construct the flip bits set (FBS) to identify unreliable bits and flip them in the next CRC-aided belief propagation list (CA-BPL) decoding attempt. The concept of FBS is proposed based on the decoding result of CA-BPL decoder, making it faster than the previous CS in finding error bits.

The rest of this paper is organized as follows. Section 2 provides the preliminaries of polar codes, BP decoding, and BPL decoding. We describe the proposed BPL bit-flip decoder for polar codes in Section 3. In Section 4, we present the simulation results of the proposed BPL bit-flip decoder for polar codes, whereas Section 5 concludes this paper.

2 Preliminaries

2.1 Polar codes

For polar codes with block length $N = 2^n$, their encoding process can be written as

$$x_1^N = u_1^N \boldsymbol{G}_N, \tag{1}$$

where $u_1^N = (u_1, u_2, \dots, u_N)$ is the binary information sequence and $x_1^N = (x_1, x_2, \dots, x_N)$ denotes the encoded codeword. The generator matrix $\boldsymbol{G}_N$ is written as $\boldsymbol{G}_N = \boldsymbol{B}_N \boldsymbol{F}^{\otimes n}$, where $\boldsymbol{B}_N$ denotes the bit-reversal permutation matrix [1] and $\boldsymbol{F} = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. Based on the polarization theorem [1], there are N polarized subchannels with different reliability. K most reliable subchannels are picked to transmit information sequence (unfrozen bits) and the others for frozen bits (which are always set to zero). $\mathcal{A}$ is the index set of K information bits in u_1^N, and the other $N-K$ indices compose the frozen bits set $\mathcal{A}^c$. $\mathcal{A}$ and $\mathcal{A}^c$ can be determined through the code construction algorithms, including Gaussian approximation (GA) [18] and polarization weight (PW) [19].

In this paper, GA is employed as the code construction algorithm. In GA, the LLRs in (3) and (4) in [18] are considered as Gaussian random variables with $D[L_N^i] = 2E[L_N^i]$, under the AWGN channel, whereas D and E are the variance and mean, respectively. From (5) and (6) in [16], the error probability of each subchannel can be calculated as follows:

$$Pe(u_i) = Q\left(\sqrt{E[L_N^i]/2}\right), \quad 1 \leqslant i \leqslant N, \tag{2}$$

where $Q(x) = \frac{1}{2\pi} \int_x^{+\infty} \mathrm{e}^{-\frac{t^2}{2}} \mathrm{d}t$. K subchannels with the smallest error probability are chosen to compose $\mathcal{A}$, and the rest composes $\mathcal{A}^c$.

2.2 Belief propagation decoding

The BP decoding is a message-passing decoding method based on the factor graph. In this paper, two types of messages are passed in the form of LLR, i.e., the left-to-right message R and the right-to-left message L. Let $\boldsymbol{L}$ and $\boldsymbol{R}$ denote two matrices with N rows and $n+1$ columns that store L and R messages, respectively. $\boldsymbol{L}$ and $\boldsymbol{R}$ are initialized as

$$\boldsymbol{L}_{i,j} = \begin{cases} \mathrm{LLR}_i, & j = n+1, \\ 0, & j \neq n+1, \end{cases} \tag{3}$$

$$\boldsymbol{R}_{i,j} = \begin{cases} +\infty, & j = 1,\ i \in \mathcal{A}^c, \\ 0, & \text{others}, \end{cases} \tag{4}$$

where LLR_i means the LLR of the i-th received bits and the first column elements of rows corresponding to the frozen bits in $\boldsymbol{R}$ are initialized to $+\infty$. Every iteration of the BP decoding consists of two procedures, i.e., the updating of $\boldsymbol{R}$ matrix and updating of $\boldsymbol{L}$ matrix. In this paper, the updating rules are the same as (3) in [20]. After iterations, the estimation of information sequence and codeword $\hat{u}_1^N = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N)$ and $\hat{x}_1^N = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)$ can be obtained as

$$\hat{u}_i = \begin{cases} 1, & \boldsymbol{L}_{i,1} + \boldsymbol{R}_{i,1} < 0, \\ 0, & \boldsymbol{L}_{i,1} + \boldsymbol{R}_{i,1} \geqslant 0, \end{cases} \tag{5}$$

$$\hat{x}_i = \begin{cases} 1, & \boldsymbol{L}_{i,n+1} + \boldsymbol{R}_{i,n+1} < 0, \\ 0, & \boldsymbol{L}_{i,n+1} + \boldsymbol{R}_{i,n+1} \geqslant 0. \end{cases} \tag{6}$$

The moment to stop the iteration is determined by the early stop criterion. In this paper, once the equation

$$\hat{x}_1^N = \hat{u}_1^N \boldsymbol{G}_N \tag{7}$$

is satisfied, or the iteration reaches the maximum iteration times, the iteration stops.

2.3 BPL decoding

As shown in Figure 2 in [16], for polar codes with length $N = 2^n$, there are n steps of a factor graph, so there are $n!$ different permuted factor graphs based on the generator matrix $\boldsymbol{G}_N$. The BP decoding based on the original factor graph may not obtain the correct decoding result but maybe success when decoding on another factor graph. The authors in [17] proposed the BPL decoding for polar codes. In the BPL decoding, there are L BP decoders based on different factor graphs executing in parallel. Moreover, based on the observation that cyclic shifts of the original factor graph are better than the randomly permuted factor graphs, the $n-1$ cyclic shifts are always within L factor graphs. Among L decoding results, the valid results are those satisfying (7). Based on the observation that the BPL decoding can benefit only slightly from adding CRC code, the BPL in [17] does not employ the CRC to choose the final result of BPL decoder, which is used in CA-SCL decoder [4]. The final result of BPL is chosen from the valid results with its codeword estimation $\hat{x}_1^N$ closest to the channel output $y_1^N = (y_1, y_2, \ldots, y_N)$ in terms of Euclidean distance. The BPL decoding can achieve better BLER performance than BP decoding at the cost of higher complexity (L times the complexity of BP approximately) and hardware requirements, but still not be comparable to CA-SCL decoding.

3 Belief propagation list bit-flip decoder for polar codes

In this section, we propose a BPLF decoder to further improve the BLER performance of BPL decoder. First, the concept of FBS is proposed to identify the information bits that may be estimated incorrectly with high probability by taking the result of the CA-BPL decoder into account. Then, the BPLF decoder based on FBS is proposed.

3.1 Construction of FBS

The FBS is the ordered set of the indices of information bits, which are the most likely to be decoded incorrectly and will be flipped to try to obtain the correct decoding result. The size of the FBS is denoted by T $(0 < T \leqslant K)$, and the value of T can be predetermined. FBS is constructed based on the result of CA-BPL decoding. Let $\mathrm{rec}_{p,1}$ and $\mathrm{rec}_{p,2}$ $(1 \leqslant p \leqslant K)$ denote the number of BP decoders in CA-BPL decoder whose p-th unfrozen bit is decoded to 0 and 1, respectively. Define

$$\mathrm{ratio}_p = \frac{\min(\mathrm{rec}_{p,1}, \mathrm{rec}_{p,2})}{\max(\mathrm{rec}_{p,1}, \mathrm{rec}_{p,2})}, \quad 1 \leqslant p \leqslant K \tag{8}$$

as the metric to reveal the difference of L BP decoders in decoding p-th unfrozen bit. The indices of unfrozen bits are sorted in descending order of ratio_p, and the first T bit indices are chosen to form the ordered set FBS.

Larger ratio_p indicates that the estimations of p-th unfrozen bit of the BP decoders with different factor graphs are more different. Hence, information bits with larger ratio_p are considered to be decoded incorrectly with higher probability and should be flipped to get the correct result. The motivation that we propose FBS to identify the bit indices, which should be flipped instead of using CS [6] directly lies in two aspects. First, the CS is proposed for SC-based decoders to find the first error bit. Owing to its sequential nature, it is crucial to find the first error bit in the SC-based decoders to avoid severe error propagation. Although the BP-based decoders are inherently parallel, their error propagation is different from the SC-based decoders. Second, the CS only considers the code construction of polar codes, and there is no sorting procedure. Whereas the construction of the FBS is related to the actual noise of channel, decoding results of different BP decoders can be utilized fully. Thus, the sorting procedure in constructing the FBS allows it to find the incorrect bit more quickly.

3.2 Proposed BPLF decoder

Figure 1 shows the flow chart of the proposed BPLF decoding algorithm.

The CA-BPL decoder is first employed. Then, after L BP decoders with different factor graphs execute in parallel, if there are valid results from BP decoders that can pass the CRC check, the final result is chosen from these valid results. The codeword estimation $\hat{x}_1^N$ is the closest to the channel output $y_1^N = (y_1, y_2, \ldots, y_N)$ in terms of Euclidean distance, and the BPLF decoder can be considered success. Otherwise, CA-BPL decoding with one-bit flipping is employed to try to find the correct decoding result. After FBS is constructed based on the CA-BPL decoding result, as in Subsection 3.1, parameter t is employed to count the number of bit-flips and is initialized as $t = 1$. Let index_t, $1 \leqslant t \leqslant T$ denote the t-th element in FBS. The estimation of index_t-th bit in information sequence is flipped in the next CA-BPL decoding turn to try to obtain the correct decoding result. The procedure of one-bit-flipping for all L BP decoders is

$$\boldsymbol{R}_{\mathrm{index}_t,1} = \begin{cases} +\infty, & \hat{u}_{\mathrm{index}_t} = 1, \\ -\infty, & \hat{u}_{\mathrm{index}_t} = 0. \end{cases} \tag{9}$$

For each BP decoder, if $\hat{u}_{\mathrm{index}_t} = 1$, $\boldsymbol{R}_{\mathrm{index}_t,1}$ is initialized to $+\infty$, and according to (5), $\hat{u}_{\mathrm{index}_t}$ will be flipped to zero and vice versa. After the one-bit-flipping procedure, the CA-BPL decoding is employed. Once there exist valid results in the CA-BPL decoding with one-bit flipping, the BPLF decoder is considered as a success. Otherwise, we set $t = t + 1$ and continue a new round of CA-BPL decoding with one-bit flipping to get the correct decoding result. If $t > T$, which means every bit index in the FBS has been flipped and still cannot get the correct result, the BPLF decoding fails.

In the BPLF decoder, the differences between the CA-BPL and the original BPL decoding that we should mention are two-fold. First, the method selects the factor graphs that we need. In [16], the authors found that the permuted factor graphs that fix the front k $(0 \leqslant k < n)$ steps and only permute the back $n - k$ steps can achieve better BLER performance than randomly select factor graphs and cyclic shifts factor graphs. Based on this observation, we narrow the search space and choose L permuted factor graphs with the highest probability to decode successfully from $(n - k)!$ factor graphs. The GA [18] is employed to calculate the probability to decode successfully in each factor graph. Secondly, in the original BPL decoder, the valid results are those satisfying (7); whereas, in this paper, the valid results should also pass the CRC check. Furthermore, because there is one CA-BPL decoding procedure and at

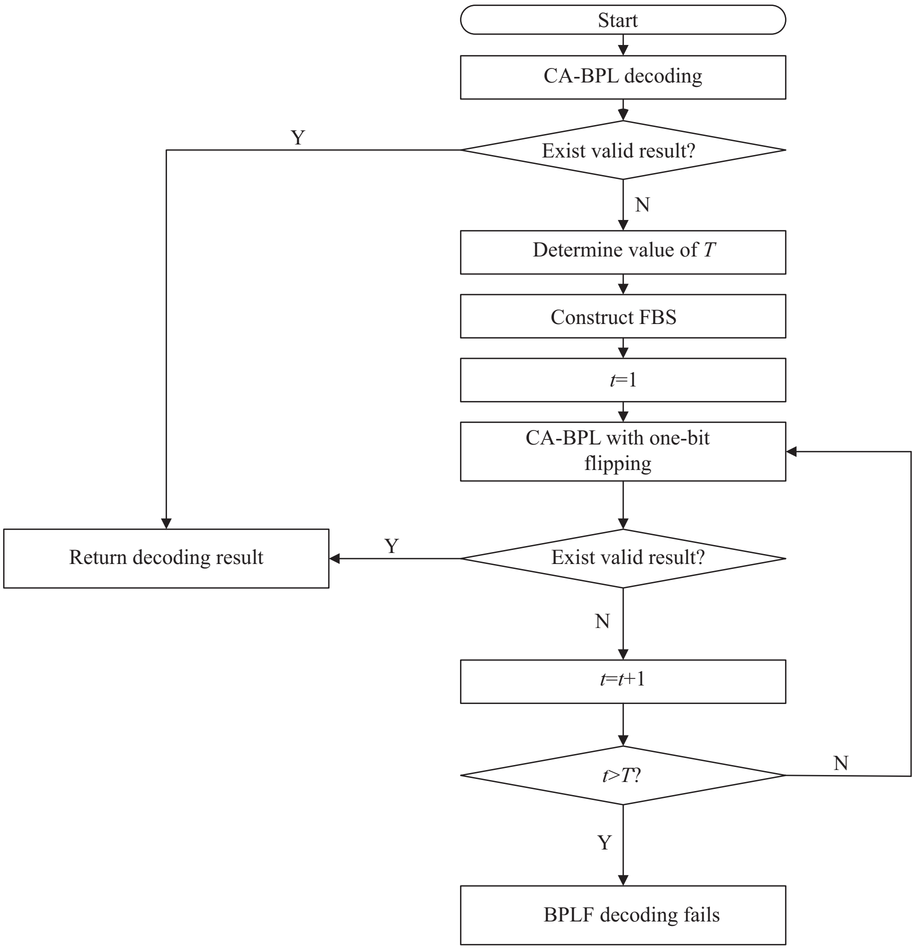

Figure 1 Flow chart of the proposed BPLF decoder.

most T times CA-BPL decoding with one-bit flipping, the maximum decoding latency and computational complexity of the proposed BPLF decoder can be considered as $(T+1)$ times of the CA-BPL decoder.

4 Simulation results and analysis

In this section, the BLER performance, average decoding complexity, and decoding latency of the proposed BPLF decoding are provided. $\mathcal{P}(N, K+r)$ denotes polar codes with length N and $K+r$ unfrozen bits (including r CRC bits). Further, 24-bit CRC with $g(x) = x^{24} + x^{23} + x^6 + x^5 + x + 1$ and 8-bit CRC with $g(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1$ are used in different code length. The simulation is under AWGN channel and BPSK modulation. The maximum number of iterations for every BP decoder in the BPLF algorithm is set to 100.

Figure 2 shows the BLER performance of the BPLF decoder. The simulation results show that the BPLF decoding is superior to CA-BPL decoder, e.g., there is 0.2 dB gain at BLER $= 10^{-4}$, and 0.3 dB gain at BLER $= 10^{-5}$ under $\mathcal{P}(1024, 512{+}8)$ with $T = \frac{1}{4}(K{+}r) = 130$. The BLER performance difference of BPLF decoder with different T is presented in Figure 2. The BLER of BPLF with $T = \frac{1}{2}(K{+}r) = 260$ is almost the same as the BLER of BPLF with $T = \frac{1}{4}(K+r) = 130$, and the BLER gap between BPLF with $T = (K+r) = 520$ and $T = \frac{1}{4}(K+r) = 130$ is less than 0.2 dB in all simulated signal-to-noise ratio (SNR) region. This simulation result can be considered as the evidence to support the validity of the FBS to find the error bits in the BPL decoding. And T can be set to $\frac{1}{4}(K+r)$, which is a tradeoff between BLER and computational complexity.

In Figure 3, the BLER performance of BPLF and other decoders under $\mathcal{P}(2048, 1024{+}24)$ are provided. To verify the universality of the BPLF decoder, the list size of BPLF is set to 32 and 64. The BPF decoder was proposed in [20] based on critical set [6], and the parameter w in BPF decoder denotes how many

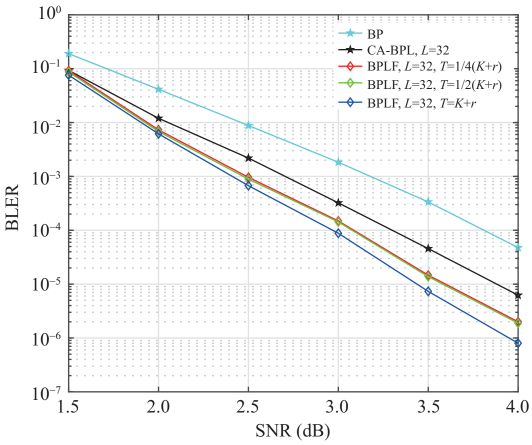

Figure 2 (Color online) BLER performance of BPLF for $\mathcal{P}(1024, 512+8)$.

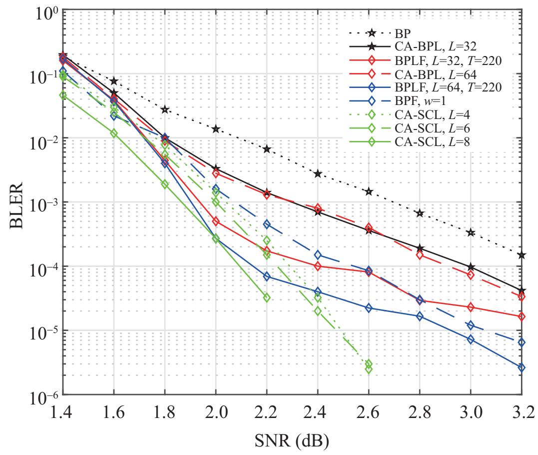

Figure 3 (Color online) BLER performance of BPLF for $\mathcal{P}(2048, 1024+24)$.

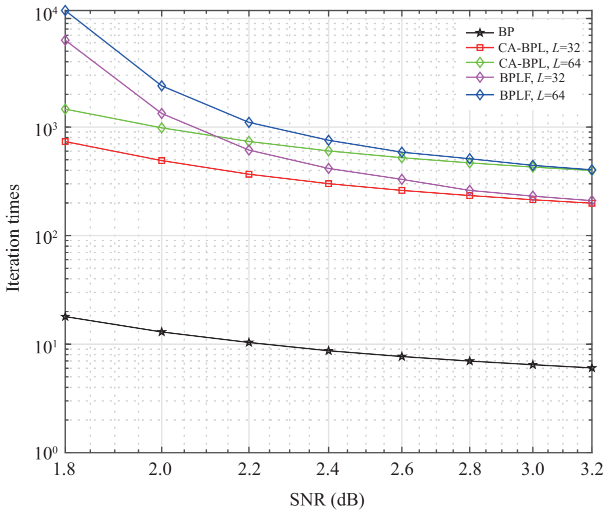

Figure 4 (Color online) Computation complexity $\overline{\text{Iter}}$ of BPLF for $\mathcal{P}(2048, 1024+24)$.

bits can be flipped at most at one time. To ensure that the comparison is intuitive, the parameter T in the BPLF decoder is set to 220, which is the same as the size of the critical set in BPF decoder. Then, the simulation results demonstrate that the BLER of BPLF is significantly superior to that of CA-BPL with $L = 64$; e.g., there is 0.75 dB gain at BLER=10^{-4}. Results show that increasing the list size of BPLF can further improve the BLER of the BPLF decoder. The BLER of CA-BPL with $L = 32$ and $L = 64$ are very close while the BLER of BPLF with $L = 64$ is superior to that of BPLF with $L = 32$; e.g., there is 0.25 dB gain at BLER $= 10^{-4}$. Because FBS is constructed by the result of CA-BPL, the probability of FBS to find error bits is improved by increasing list size with an unchanged parameter of T. The comparison between BPLF and BPF indicates that the BLER of BPLF with $L = 64$ is superior to BPF ($w = 1$); e.g., there is 0.4 dB gain at BLER $= 10^{-4}$. Furthermore, the BLER of BPLF with $L = 64$ is comparable to CA-SCL with $L = 6$ when SNR < 2.3 dB, but in higher SNR region, the BLER of BPLF with $L = 64$ is inferior to that of the CA-SCL with $L = 6$, which may result from the error floor of BP decoding. The simulation shows that the proposed BPLF decoder further narrows the BLER performance gap between the BP-based decoder and CA-SCL decoder.

The CA-BPL decoder and BPLF decoder are based on the BP decoder, hence the average computational complexity can be computed using $2N \cdot \log_2 N \cdot \overline{\text{Iter}}$ where $\overline{\text{Iter}}$ denotes the average iteration times. In Figure 4, the $\overline{\text{Iter}}$ of the proposed BPLF decoder is presented. As SNR increases, the $\overline{\text{Iter}}$ of BPLF approaches the $\overline{\text{Iter}}$ of CA-BPL with the same list size. The reason is with the increase of SNR, the BLER of CA-BPL decoding decreases, hence the probability to employ CA-BPL decoding with one-bit

Table 1 Average decoding clock cycle of the proposed BPLF decoder for $\mathcal{P}(2048, 1024 + 24)$

Decoder	SNR (dB)						
	2	2.2	2.4	2.6	2.8	3.0	3.2
BP	285	228	191	169	154	142	133
CA-BPL, $L = 64$	357	265	213	182	162	148	133
BPLF, $L = 64, T = 220$	869	408	274	208	180	154	133
BPF, $w = 1$	1805	847	484	317	228	180	152
SCL, $L = 6$	5142	5142	5142	5142	5142	5142	5142

flipping decreases, and the wrongly estimated information bits are easier to be corrected by the bit-flip method.

Next, we present the decoding latency comparison between the proposed BPLF and some other decoders. In Table 1, the target code is $\mathcal{P}(2048, 1024 + 24)$; the BPF($w = 1$) represents the proposed BPF decoder with parameter $w = 1$ in [20] and SCL ($L = 6$) denotes the decoder proposed by [4] with list size equals to 6. The simulation results demonstrate that the decoding clock cycle of the proposed BPLF decoder approaches that of the traditional BP decoder as SNR increases. The decoding latency is shorter than BPF($w = 1$) decoder, which confirms that with the help of permuted factor graphs and the ordered set FBS, the BPLF decoder can find the error estimated bit quicker than BPF decoder. Although decoding latency of the BPLF decoder is slightly higher than the BP decoder, it is still much faster than the SC-based decoder, which means the proposed BPLF decoder is suitable for the scenarios that need low decoding latency.

5 Conclusion

In this paper, first, we introduced the bit-flip method to the BPL decoding scheme for polar codes and proposed a BPLF decoder based on FBS. Furthermore, we presented the BLER performance, decoding complexity, and decoding latency of the proposed BPLF decoder. The simulation results demonstrate that the BLER of the proposed BPLF decoder can approach that of the CA-SCL with $L = 6$. Thus, the decoding complexity and latency of the BPLF decoder can approach that of the CA-BPL decoder with the same list size in medium and high SNR regions.

Acknowledgements This work was partially supported by National Key Research and Development Project (Grant No. 2018-YFB1802402) and Huawei Tech. Co., Ltd.

References

1 Arikan E. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans Inform Theor, 2009, 55: 3051–3073

2 Niu K, Chen K, Lin J R, et al. Polar codes: primary concepts and practical decoding algorithms. IEEE Commun Mag, 2014, 52: 192–203

3 Alamdar-Yazdi A, Kschischang F R. A simplified successive-cancellation decoder for polar codes. IEEE Commun Lett, 2011, 15: 1378–1380

4 Tal I, Vardy A. List decoding of polar codes. IEEE Trans Inform Theor, 2015, 61: 2213–2226

5 Asiadis O, Balatsoukas-Stimming A, Burg A. A low-complexity improved successive cancellation decoder for polar codes. In: Proceedings of the 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, 2014. 2116–2120

6 Zhang Z, Qin K, Zhang L, et al. Progressive bit-fipping decoding of polar codes over layered critical sets. In: Proceedings of IEEE Global Communications Conference, Singapore, 2017. 1–6

7 Yu Y R, Pan Z W, Liu N, et al. Successive cancellation list bit-flip decoder for polar codes. In: Proceedings of the 10th International Conference on Wireless Communications and Signal Processing, Hangzhou, 2018. 1–6

8 Arkan E. A performance comparison of polar codes and reed-muller codes. IEEE Commun Lett, 2008, 12: 447–449

9 Arikan E. Polar codes: a pipelined implementation. In: Proceedings of the 4th International Symposium Broadband Communication, Melaka, 2010. 11–14

10 Yuan B, Parhi K K. Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders. IEEE Trans Signal Process, 2014, 62: 6496–6506

11 Simsek C, Turk K. Simplified early stopping criterion for belief-propagation polar code decoders. IEEE Commun Lett, 2016, 20: 1515–1518

12 Zhang Q, Liu A J, Tong X H. Early stopping criterion for belief propagation polar decoder based on frozen bits. Electron Lett, 2017, 53: 1576–1578

13 Yuan B, Parhi K K. Algorithm and architecture for hybrid decoding of polar codes. In: Proceedings of the 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, 2014. 2050–2053

14 Sun S H, Cho S G, Zhang Z Y. Post-processing methods for improving coding gain in belief propagation decoding of polar codes. In: Proceedings of IEEE Global Communications Conference, Singapore, 2017. 1–6

15 Elkelesh A, Ebada M, Cammerer S, et al. Belief propagation decoding of polar codes on permuted factor graphs. In: Proceedings of IEEE Wireless Communications and Networking Conference, Barcelona, 2018. 1–6

16 Doan N, Hashemi S A, Mondelli M, et al. On the decoding of polar codes on permuted factor graphs. In: Proceedings of IEEE IEEE Global Communications Conference, Abu Dhabi, 2018. 1–6

17 Elkelesh A, Ebada M, Cammerer S, et al. Belief propagation list decoding of polar codes. IEEE Commun Lett, 2018, 22: 1536–1539

18 Trifonov P. Efficient design and decoding of polar codes. IEEE Trans Commun, 2012, 60: 3221–3227

19 Zhang H Z, Li R, Wang J, et al. Parity-check polar coding for 5G and beyond. In: Proceedings of International Conference on Communications, Kansas City, 2018, 1–6

20 Yu Y R, Pan Z W, Liu N, et al. Belief propagation bit-flip decoder for polar codes. IEEE Access, 2019, 7: 10937–10946