SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

October 2018, Vol. 61 102302:1–102302:10
https://doi.org/10.1007/s11432-017-9312-9

A low-latency list decoder for polar codes

Qingyun XU, Zhiwen PAN*, Nan LIU & Xiaohu YOU

National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096*, China*

Received 1 September 2017/Revised 18 October 2017/Accepted 12 December 2017/Published online 21 May 2018

Abstract Successive cancellation (SC) is a low complexity serial decoding algorithm for polar codes, and successive cancellation list (SCL) can achieve excellent error-correcting performance. However, SCL decoder suffers from long decoding latency compared with belief propagation (BP) decoder. In this paper, a low-latency list decoder whose latency performance can approach that of BP deocder is proposed. A prunable subtree recognizing scheme based on $\boldsymbol{H}$-Matrix check is proposed by taking the reliability of frozen bits into account. Then, a latency-reduced list decoder based on the prunable constituent codes is proposed. Simulation results show that the decoding latency of proposed list scheme can be reduced significantly, especially for high signal noise ratio (SNR) region.

Keywords list decoder, low latency, polar codes, successive cancellation

Citation Xu Q Y, Pan Z W, Liu N, et al. A low-latency list decoder for polar codes. Sci China Inf Sci, 2018, 61(10): 102302, https://doi.org/10.1007/s11432-017-9312-9

1 Introduction

Polar coding is an excellent error-correcting coding scheme that can achieve symmetric channel capacity under binary-input discrete memoryless channel (B-DMC) [1], and is considered as candidate coding scheme in the next generation wireless communications [2, 3].

Successive cancellation (SC) and belief propagation (BP) are two common decoding schemes for polar codes [4]. BP decoding is an iterative algorithm that can be implemented with parallel structure based on factor graph representation of polar codes [5]. Scaled min-sum (SMS) BP decoder shows faster convergence rate than original BP decoder, but with similar error correction performance [6]. The iteration number is fixed in BP and SMS decoder. However, the decoding process may converge before the iteration reaches the max iteration number. $\boldsymbol{G}$-Matrix BP and adaptive minimum log-likelihood ratio (minLLR) BP proposed in [7] are two early stopping schemes where the decoding iteration number can be reduced significantly. Hence, the complexity and decoding latency decrease remarkably compared with decoder with fixed iteration. Authors in [8] proposed a simplified early stopping criterion that the decoder stops early if the worst information bits are detected successfully. However, the need for a large number of iterations to improve the error correction performance of BP decoder makes BP decoder suffer from high Latency [9].

SC is an effective decoding algorithm with low complexity [1]. Successive cancellation list (SCL) [10] decoding with cyclic redundancy check (CRC) [11] improves the block error rate (BLER) performance greatly, and even outperforms that of low-density parity-check (LDPC) codes at similar code length.

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

However, SC based decoders suffer from long decoding latency due to the serial decoding process. Simplified successive cancellation (SSC) [12] and Fast SSC (FSSC) [13] can reduce the decoding latency by pruning subtrees in the decoding tree. SSC list (SSCL) and FSSC list (FSSCL) proposed in [14, 15] are list algorithms of SSC and FSSC. List decoder with multi-bit decision introduced in [16] can reduce latency by determining multi bits simultaneously. Symbol-decision SCL decoder based on symbol recursive channel combination has lower latency and complexity compared with bit-wise channel polarization [17]. A scheme deleting certain paths whose metrics are lower than a threshold is proposed in [18].

There is still a great gap between the latency of existing SCL based decoder and that of BP decoder [19]. In this paper, we propose a low-latency list algorithm whose decoding latency approaches that of BP deocder. A prunable subtree recognizing scheme based on $\boldsymbol{H}$-Matrix check is proposed by taking the reliability of frozen bits into account. In the decoding process, the frozen bits can provide extra information (frozen information) that the frozen bits can be decoded as zero directly. It is proven that if $\boldsymbol{H}$-Matrix check is satisfied for a constituent code, its frozen bits can be decoded accurately without the help of frozen information. The worst channels are chosen to transmit frozen bits. Thus, the decoding reliability is sufficiently high if the frozen bits can be decoded correctly without the help of frozen information. The constituent code can be decoded directly without traversing the subtree when the decoding reliability is sufficiently high. Then, we extend previous study in [16] and propose a list decoding scheme based on subtree pruning. The SCL based decoder traverses decoding tree sequentially. Hence, the decoding latency can be reduced effectively by pruning subtrees of some constituent codes.

The rest of the paper is organized as follows. Section 2 introduces the background of polar codes, SC and SCL decoder. Section 3 describes the proposed low-latency decoding algorithm. Section 4 shows the simulation results and analysis of the proposed decoding scheme. Section 5 concludes the paper.

2 Preliminary

2.1 Polar codes

Polar coding is a kind of linear block coding scheme which can be encoded as

$$\boldsymbol{x}_1^N = \boldsymbol{u}_1^N \boldsymbol{G}_N, \tag{1}$$

where $\boldsymbol{x}_1^N = (x_1, x_2, \ldots, x_N)$ is the encoded bit sequence and $\boldsymbol{u}_1^N = (u_1, u_2, \ldots, u_N)$ is the source bit sequence. $\boldsymbol{G}_N$ denotes the generator matrix, and is discussed in [1] detailedly. For an (N, K) polar code, the K most reliable subchannels are selected to transmit the K information bits, and the frozen bits are transmited by the remaining subchannels [20].

2.2 SC decoding

The SC decoding estimates the information bits sequentially by performing hard decision on the log-likelihood ratio (LLR). The SC decoding can be implemented by traversing the decoding tree, which is a binary tree with depth $n = \log N + 1$. The decoding tree of an (8, 4) polar code is shown in Figure 1, where solid leaves represent the information bits, and hollow ones denote the frozen bits.

The root node of the decoding tree receives the channel LLRs. For a general node, its LLR information $\boldsymbol{\alpha}_v$ is received from its parent node. Then, the node passes $\boldsymbol{\alpha}_l$ to its left child node, and passes $\boldsymbol{\alpha}_r$ to its right child node after the left subtree is returned. The node returns a bit vector $\boldsymbol{\beta}_v$ to its parent node in the end. The child nodes pass the LLR message in the same manner.

When a leaf node receives the LLR information, $\boldsymbol{\beta}_v$ will be zero if the leaf node represents a frozen bit. Otherwise, $\boldsymbol{\beta}_v$ is decoded by

$$\boldsymbol{\beta}_v h(\boldsymbol{\alpha}_v), \tag{2}$$

where

$$h(x) = \begin{cases} 0, & x > 0, \\ 1, & \text{else.} \end{cases} \tag{3}$$

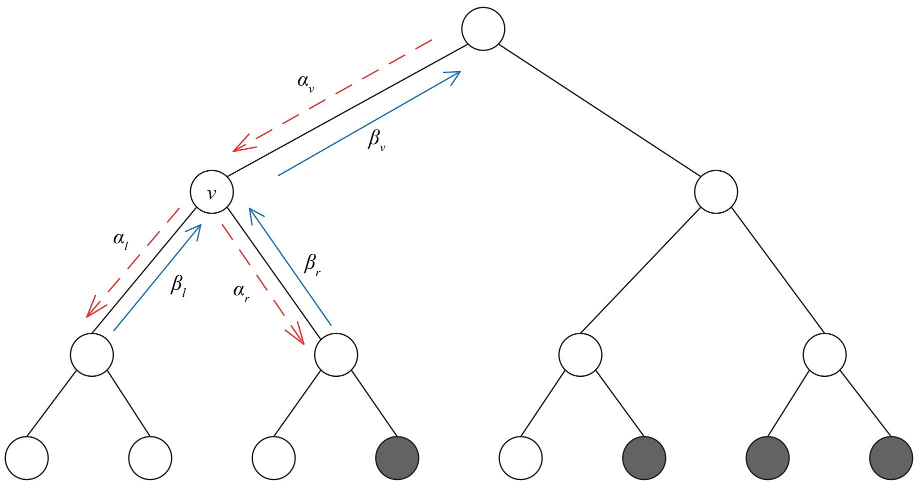

Figure 1 (Color online) Decoding tree of an (8, 4) polar code.

The SC decoding is completed when the bit sequence $\boldsymbol{\beta}_v$ is outputed from the root node, the transmitted bits are estimated as

$$\hat{\boldsymbol{u}}_1^N = \boldsymbol{\beta}_v \boldsymbol{G}_N. \tag{4}$$

In SSC decoder, the subtree whose leaf nodes are all frozen bits (Rate-0 nodes) or all information bits (Rate-1 nodes) can be pruned. The output of Rate-0 node will always be zero. Rate-1 node can output $\boldsymbol{\beta}_v$ directly without recursion by

$$\beta_v[i] = h(\alpha_v[i]), \quad \text{for } 1 \leqslant i \leqslant N_v. \tag{5}$$

Single-parity-check (SPC) node and repetition (REP) node in FSSC decoding [13] can further prune some subtrees to reduce the decoding latency.

2.3 List-CRC decoding

The SCL assumes that both 0 and 1 are estimated bits of an information bit. At each decoding step, one decoding path will generate 2 new decoding paths, one path takes 0 as the estimated bit, another takes 1. The list decoder keeps L paths with largest path metric (PM) and discards the remaining paths. For the i-th decoding step, the path metric PM_l^i of path l is updated by

$$\mathrm{PM}_l^i = \begin{cases} \mathrm{PM}_l^{i-1}, & \text{if } \hat{u}_i = h(\alpha_i), \\ \mathrm{PM}_l^{i-1} - |\alpha_i|, & \text{otherwise}, \end{cases} \tag{6}$$

where α_i denotes the LLR of the i-th bit. The CRC check is used to select the correct decoding path after the decoding is completed.

3 Low-latency list algorithm

In this section, we propose a low-latency list decoding scheme that can prune constituent codes which satisfy $\boldsymbol{H}$-Matrix check. Firstly, a scheme based on $\boldsymbol{H}$-Matrix check is proposed by taking the reliability of frozen bits into account to recognize prunable constituent codes. Then, a list decoding scheme is proposed based on subtree pruning. The SC based list decoder traverses decoding tree sequentially. Thus, the decoding latency can be reduced effectively by pruning subtrees of some constituent codes.

3.1 Latency analysis of SC based decoder

The SC decoding can be represented by a binary tree with depth for block length N. When a non-leaf tree node is activated, it takes 3 clock cycles (CC) to calculate $\boldsymbol{\alpha}_l$, $\boldsymbol{\alpha}_r$, $\boldsymbol{\beta}_r$, respectively. And for a leaf tree node, it takes 1 clock cycle to calculate $\boldsymbol{\beta}_r$. Hence the clock cycles SC decoder takes is

$$\mathrm{CC}_{\mathrm{SC}} = 3(2^n - 1) + 2^n = 4N - 3, \tag{7}$$

where N is code length and $n = \log N + 1$.

For SSC and FSSC, some subtrees can be pruned and it is unnecessary to traverse all tree nodes. Hence the clock cycles of FSSC decoder is

$$\mathrm{CC}_{\mathrm{FSSC}} = 3M_t + M_d, \tag{8}$$

where M_t is the number of traversed nodes and M_d is the number of nodes that can be decoded directly. However, there is no analytical expressions for M_t and M_d for different code length N since they are relevant to the information bits selection schemes. However, it is certain that $M_t < 2^n - 1$ and $M_d < 2^n$. Hence, $\mathrm{CC}_{\mathrm{FSSC}} < \mathrm{CC}_{\mathrm{SC}}$.

3.2 Proposed subtree pruning scheme

The frozen bits can provide extra information (frozen information) that the frozen bits can be decoded as zero directly. Inspired by the early stop algorithm for BP decoder proposed in [7], the decoding reliability is sufficiently high if all of the frozen bits can be decoded correctly without the help of frozen information. Then, the constituent code can be decoded directly by hard decision rather than SC decoding.

Theorem 1. If $h(\boldsymbol{\alpha}_v)\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$ ($\boldsymbol{H}$-Matrix check) is satisfied for an activated node, its frozen bits can be decoded accurately in SC decoding without the help of frozen information, where $\boldsymbol{H}$ is the corresponding check matrix of the constituent code.

Proof. The frozen bits can be regarded as information bits without the help of frozen information. Then, the constituent code can be regarded as Rate-1 node.

We have $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$ for a Rate-1 node [12]. Its corresponding transmited bit vector is $\boldsymbol{u}_v$=$\boldsymbol{\beta}_v\boldsymbol{G}_{N_v}$, where N_v is length of vector $\boldsymbol{\beta}_v$ and $\boldsymbol{G}_{N_v}$ is generator matrix of length N_v. The frozen bit vector is

$$\boldsymbol{u}_{\mathrm{frozen}}=\boldsymbol{\beta}_v\boldsymbol{G}_{N_v}(A), \tag{9}$$

where A denotes index set of frozen bits, $\boldsymbol{G}_{N_v}(A)$denotes the submatrix of $\boldsymbol{G}_{N_v}$ formed by the rows with indices in A.

The check matrix $\boldsymbol{H}$ of $\boldsymbol{G}_{N_v}$ is $\boldsymbol{G}_{N_v}(A)^{\mathrm{T}}$, we may write (9) as

$$\boldsymbol{u}_{\mathrm{frozen}}=\boldsymbol{\beta}_v\boldsymbol{H}^{\mathrm{T}}. \tag{10}$$

$\boldsymbol{\beta}_v\boldsymbol{H}^{\mathrm{T}}$ denotes the frozen bit vector obtained without the help of frozen information and frozen bit should be an all-zero vector. Hence, if $\boldsymbol{\beta}_v\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$, frozen bits are decoded correctly. Considering $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$, Theorem 1 can be proven immediately.

The frozen bits are transmitted over the most unreliable subchannels, thus, the decoding reliability is sufficiently high if all of the frozen bits can be decoded correctly. Then, the tree node that satisfies $\boldsymbol{H}$-Matrix check can be decoded by hard decision instead without recursion. A subtree pruning method can be proposed based on Theorem 1. When a tree node is activated, $\boldsymbol{H}$-Matrix check and calculation of LLR vector is performed simultaneously. If $\boldsymbol{H}$-Matrix check is satisfied, $\boldsymbol{\beta}_v$ is obtained immediately via $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$, and the child nodes will not be traversed. Otherwise, it just continues traversing its child tree nodes. The flow chart of proposed subtree pruning method is shown in Figure 2.

Theorem 2. If $h(\boldsymbol{\alpha}_v)\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$ is satisfied for an activated node, the output of traditional SC decoding is $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$.

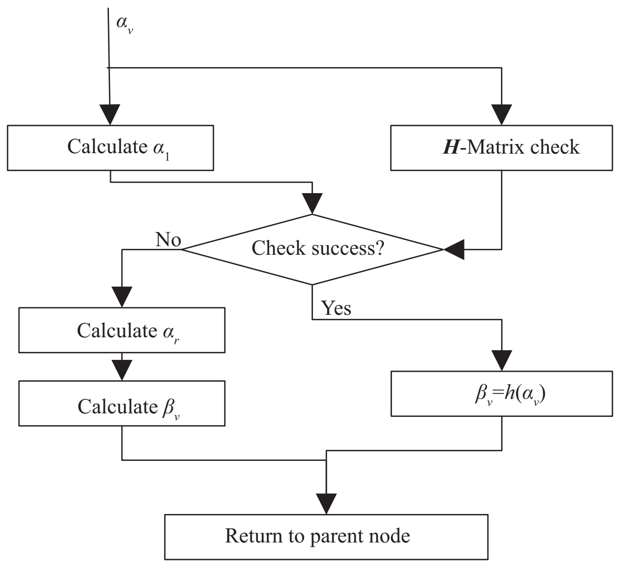

Figure 2 Flow chart of subtree pruning method.

Proof. It can be proven by mathematical induction. Denote N'_v as the length of $\boldsymbol{\alpha}_v$ and N_v as bit length of activated node.

(1) Base condition. For N'_v=1, the proof of Theorem 1 suggests that the frozen bits can be decoded correctly if $h(\boldsymbol{\alpha}_v)\boldsymbol{H}^{\rm T} = \mathbf{0}$. Thus, we have $\boldsymbol{\alpha}_v > 0$ and $\boldsymbol{\beta}_v = 0$ for frozen bits. The information bits are decoded via $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$. Therefore, $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$ holds for leaf nodes.

(2) Inductive step. Suppose $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$ holds for $N'_v = N_v/2$. For $N'_v = N_v$, we have

$$h(\alpha_l[i]) = h(\alpha_v[2i]) \oplus h(\alpha_v[2i+1]),$$

and

$$\begin{aligned} h(\alpha_r[i]) &= h\left(\alpha_v[2i+1] + (1-2h(\alpha_r[i]))\alpha_v[2i]\right) \\ &= h\left(\alpha_v[2i+1] + \{1-2h(\alpha_v[2i]) \oplus h(\alpha_v[2i+1])\}\alpha_v[2i]\right) \\ &= h(\alpha_v[2i+1]). \end{aligned}$$

According to assumption, $\boldsymbol{\beta}_l = h(\boldsymbol{\alpha}_l)$ and $\boldsymbol{\beta}_r = h(\boldsymbol{\alpha}_r)$ hold for $N'_v = N_v$. So we can arrive at

$$\begin{aligned} \beta_v[2i] &= \beta_l[i] \oplus \beta_r[i] = h(\boldsymbol{\alpha}_r) \\ &= h(\alpha_v[2i]) \oplus \mathrm{h}(\alpha_v[2i+1]) \oplus \mathrm{h}(\alpha_v[2i+1]) \\ &= h(\alpha_v[2i]), \\ \beta_v[2i+1] &= \beta_r[i] = h(\boldsymbol{\alpha}_r) = h(\alpha_v[2i+1]). \end{aligned}$$

Hence, the output of traditional SC decoding is

$$\beta_v[i] = h(\alpha_v[i]), \quad \text{for } 0 \leqslant i \leqslant N_v.$$

Namely,

$$\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v).$$

Theorem 2 can be proven immediately.

The output of hard decision is $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$ as well. Theorem 2 suggests that the proposed subtree pruning method will not introduce extra error rate for single decoding path.

3.3 Proposed low-latency list decoder

Having defined the prunable subtree recognizing scheme, a low-latency list decoding scheme is proposed. In the list decoding, L paths are retained in the list, where L is the list size. $\boldsymbol{H}$-Matrix check can be performed on each survival path, a path is prunable path if $\boldsymbol{H}$-Matrix check is satisfied. It is difficult to satisfy $\boldsymbol{H}$-Matrix check simultaneously for all paths. A path is more likely to output correct decoding result if the PM of a path is larger than other paths. If the paths that satisfy $\boldsymbol{H}$-Matrix check have larger PM, it is more likely to get accurate decoding result from these paths. Hence, the stop condition for the decoding tree recursion is by considering PM of survival path. The stop condition is

$$\frac{\sum_{i\in Z_H}\mathrm{PM}_i}{\sum_{j=1}^{L}\mathrm{PM}_j} > t, \tag{11}$$

where PM_i is the path metric of i-th path, Z_H denotes indexes set of paths that satisfy $\boldsymbol{H}$-Matrix check, L is list size, and t is a threshold parameter less than 1.

As shown in Figure 2, $\boldsymbol{H}$-Matrix check and calculation of LLR vector using (2) is performed simultaneously. Therefore, the proposed scheme will not take additional clock cycles to perform $\boldsymbol{H}$-Matrix. If (11) is satisfied, the decoding process will be same as a Rate-1 node. The output bit vector of this node is calculated via $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$. At most 3 new paths will be generated by flipping combination of the two least confident bits. $\alpha_v[\min_1]$, $\alpha_v[\min_2]$ denote the two least elements in vector $|\boldsymbol{\alpha}_v|$. For each source path s, three new paths will be generated

$$\begin{aligned}
&\mathrm{P}_1\colon \mathrm{PM}_1^p = \mathrm{PM}_s^{p-1} - |\alpha_v[\min_1]|\,,\\
&\mathrm{P}_2\colon \mathrm{PM}_2^p = \mathrm{PM}_s^{p-1} - |\alpha_v[\min_2]|\,,\\
&\mathrm{P}_3\colon \mathrm{PM}_3^p = \mathrm{PM}_s^{p-1} - |\alpha_v[\min_1]| - |\alpha_v[\min_2]|\,,
\end{aligned}$$

where path P_1 is generated by flipping $\beta_v[\min_1]$, P_2 by flipping $\beta_v[\min_2]$, P_3 by flipping both $\beta_v[\min_1]$ and $\beta_v[\min_2]$. If $\alpha_v[\min_1]$ or $\alpha_v[\min_2]$ represents the LLR of a frozen bit, its corresponding $\beta_v[\min_1]$ or $\beta_v[\min_2]$ in the new path needs to be set to zero.

According to [10], each source path s updates its metric by

$$\mathrm{PM}_s^p = \mathrm{PM}_s^{p-1} - \sum_i |\beta_v[i] - h(\alpha_v[i])|\,|\alpha_v[i]|, \tag{12}$$

where $\boldsymbol{\beta}_v$ is the new vector after path generation. PM_s^{p-1} is the path metric of path s in step p-1, PM_s^p is the path metric of path s in step p.

Less tree nodes will be traversed in the proposed list decoding scheme. According to (8), the decoding latency will be reduced further. It should be noted that, the proposed decoding algorithm has worse BLER performance since in fact we made a tradeoff between BLER performance and decoding latency. The decoding latency is reduced by pruning constituent codes that satisfy $\boldsymbol{H}$-Matrix check. A constituent code outputs N_v bits, there are at most 2^{N_v} candidates paths being generated. This approach is impractical as it scales exponentially in N_v. In this paper, only the two least reliable bits are flipped and 4 paths are generated. Therefore, proposed decoding algorithm has little degradation on BLER performance, but the BLER performance can be improved by increasing list size L. The threshold parameter t affects both the error performance and decoding latency. Smaller t leads to lower decoding latency but larger the BLER.

4 Simulation results and analysis

The performance and analysis of the proposed decoding scheme is provided in this section, BLER performance and decoding latency are obtained under additive white Gaussian noise (AWGN) channel. Code rate $R = 1/2$ and 16-bit CRC check are used in this study. The computational complexity is measured according to analysis in Subsection 3.1. The simulation results are compared with FSSCL decoding algorithm in [15].

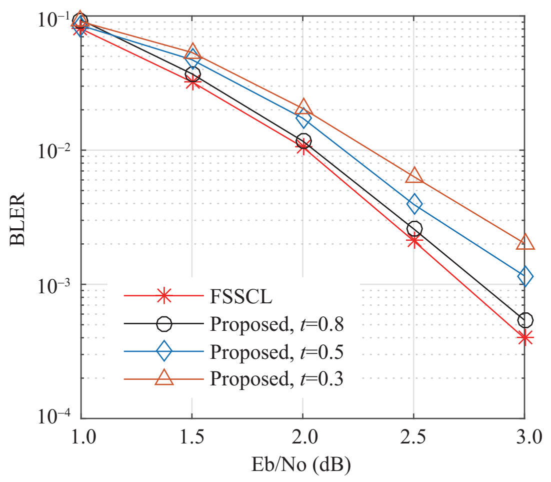

Figure 3 (Color online) BLER performance of different t for $N = 256$, $L = 8$.

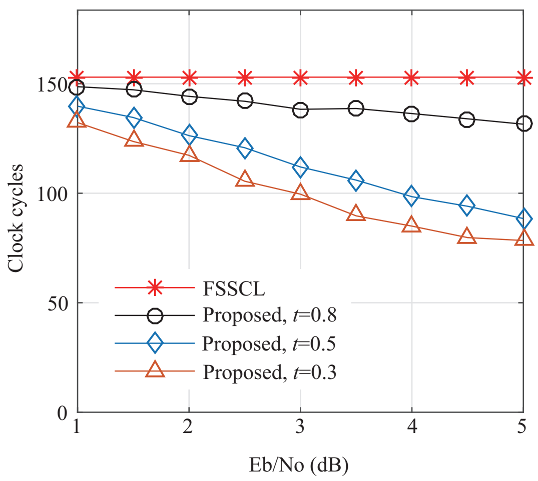

Figure 4 (Color online) Decoding latency of different t for $N = 256$, $L = 8$.

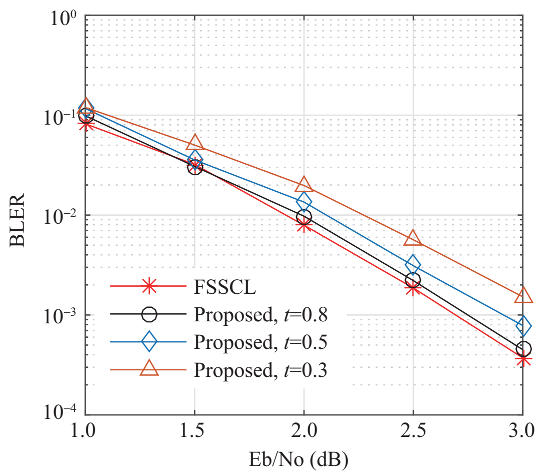

Figure 5 (Color online) BLER performance of different t for $N = 256$, $L = 16$.

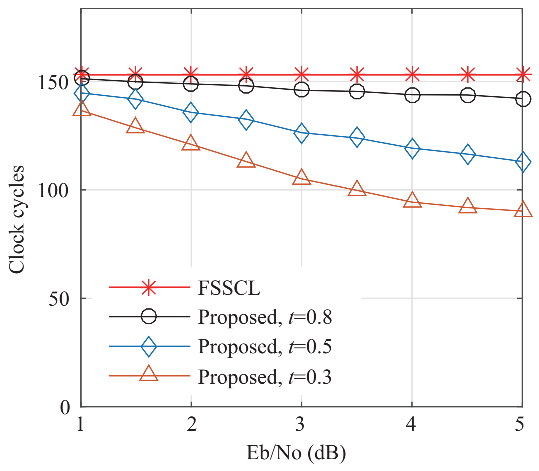

Figure 6 (Color online) Decoding latency of different t for $N = 256$, $L = 16$.

BLER performance is shown in Figure 3 for different t when list size $L = 8$, code length $N = 256$. The BLER of the proposed algorithm for $t = 0.8$ is very close to that of FSSCL decoder for $L = 8$. Hence, there is little degradation in error performance for appropriate t.

Figure 4 shows the decoding latency of different t with $L = 8$, $N = 256$. Reduction of decoding latency decreases prominently as the SNR increases since more tree nodes would be pruned as the decoding reliability increases. Decoding latency decreases with the decrease of t since less paths is needed to satisfy $\boldsymbol{H}$-Matrix check for a small t and more subtree can be pruned. However, less paths satisfy $\boldsymbol{H}$-Matrix check means the possibility of decoding correctly is low. Thus, smaller t may lead to larger the BLER. Hence, it is important to set a proper t for latency-performance trade-off.

Unfortunately, optimal t is difficult to obtain. The BLER performance and decoding latency of different t are shown in Figures 5 and 6 respectively for $L = 16$, $N = 256$. The BLER performance and decoding latency of different t are provided in Figures 7 and 8 respectively for $L = 8$, $N = 512$. The code rate of Figures 5–8 is $R = 0.5$. Figures 9 and 10 show the BLER performance and decoding latency of different t respectively for list size $L = 8$, code length $N = 256$ and $R = 0.4, 0.6$. From results of Figures 3–10, $t = 0.5$ corresponds to reduced decoding latency with acceptable deterioration in BLER performance. Thus, $t = 0.5$ may be a recommended value for a latency-performance trade-off.

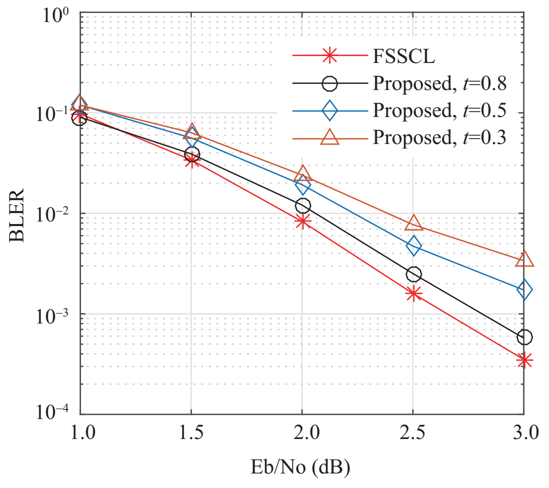

Figure 7 (Color online) BLER performance of different t for $N = 512$, $L = 8$.

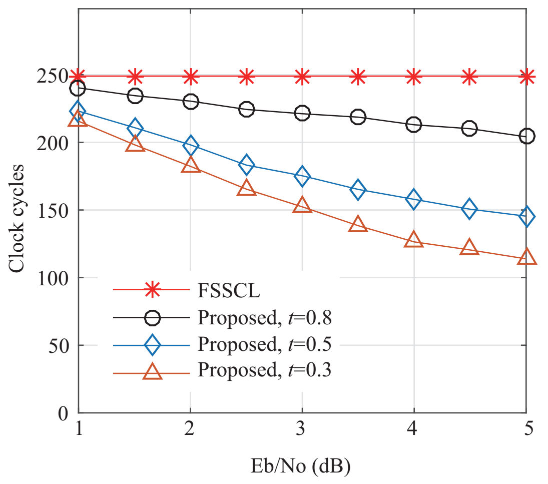

Figure 8 (Color online) Decoding latency of different t for $N = 512$, $L = 8$.

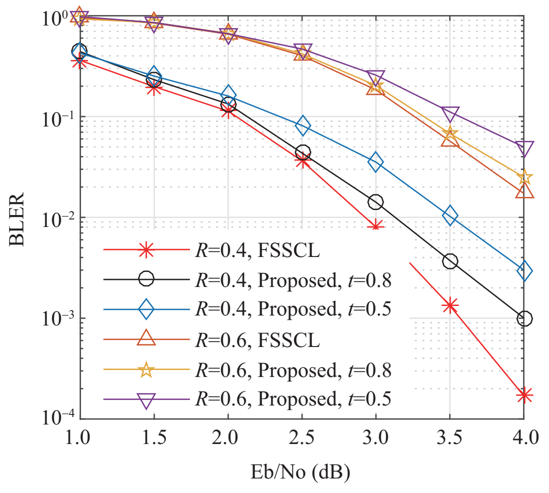

Figure 9 (Color online) BLER performance of different t for $N = 256$, $L = 8$, $R = 0.4, 0.6$.

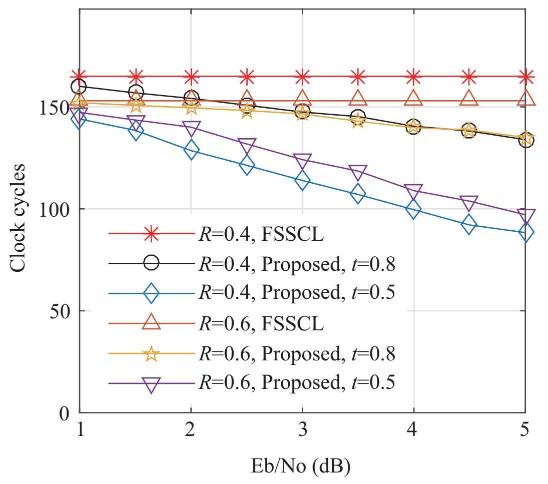

Figure 10 (Color online) Decoding latency of different t for $N = 256$, $L = 8$, $R = 0.4, 0.6$.

Figure 11 shows the BLER performance of different list size with $N = 256$, $t = 0.5$. The decoding latency under the same condition is shown in Figure 12. The results show that for $L = 4$, the proposed algorithm can reduce decoding latency remarkably with some degradation on BLER performance. However, the increasing L can make up for the degradation on BLER. The proposed algorithm with $L = 8$ shows a slight gap of BLER compared to FSSCL with $L = 4$, but outperforms FSSCL when $L = 16$. The latency of proposed algorithm with different L is lower than that of FSSCL. And the decoding latency of FSSCL is constant for different L. Larger list size means higher complexity. In other words, under same error performance, the proposed list decoder can reduce the latency at cost of increasing complexity.

The proposed list algorithm is compared with BP decoding which is a classic low-latency algorithm. CRC check will not be applied here because the implementation of CRC check by shift registers suffers from long calculation latency, the decoding scheme with CRC will takes more decoding latency than BP decoder. Figure 13 shows the error performance comparison of proposed algorithm with $N = 256$, $t = 0.5$ and $\boldsymbol{G}$-Matrix based BP decoding in [7] with Iteration $= 15, 30$. Figure 14 shows the corresponding decoding latency under the same condition. As shown in simulation results, the BLER performance of proposed algorithm without CRC can outperforms that of BP algorithm. The proposed algorithm

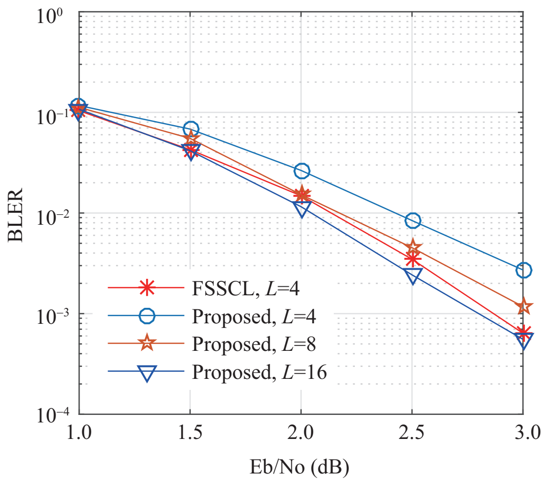

Figure 11 (Color online) BLER performance for $N = 256$, $L = 4, 8, 16$.

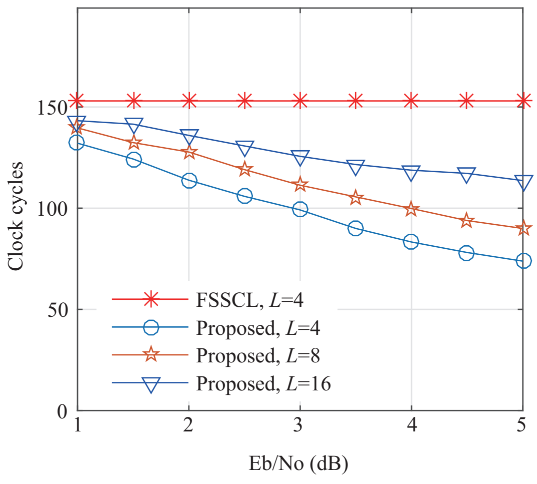

Figure 12 (Color online) Decoding latency for $N = 256$, $L = 4, 8, 16$.

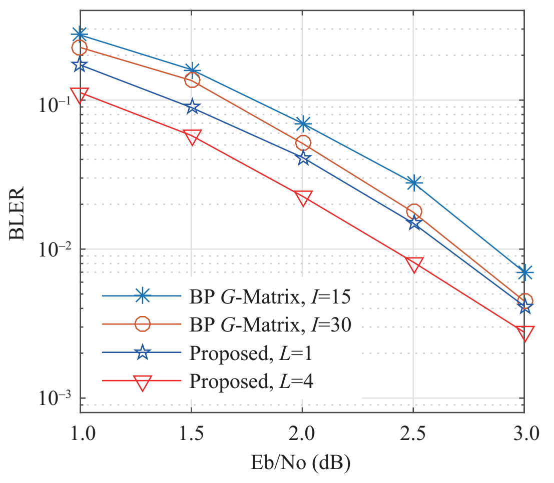

Figure 13 (Color online) BLER performance for $N = 256$, $L = 1, 4$ and BP.

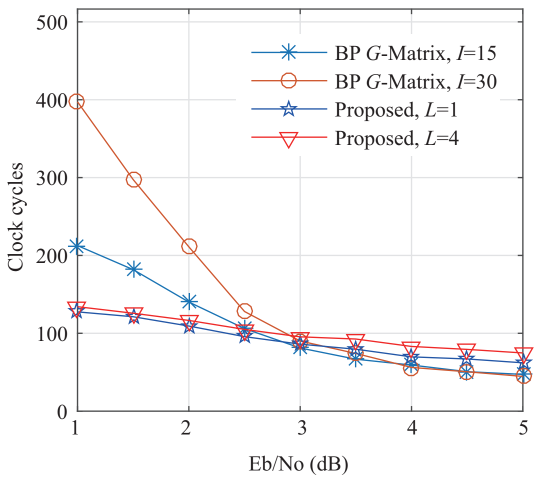

Figure 14 (Color online) Decoding latency for $N = 256$, $L = 1, 4$ and BP.

shows almost the same decoding latency than BP decoder and even lower when SNR is less than 2.5 dB. Furthermore, the SSC based list decoder has lower computation complexity than BP based decoder.

5 Conclusion

In this paper, we propose an improved list decoding scheme for polar codes that can significantly reduce the decoding latency. If $\boldsymbol{H}$-Matrix check is satisfied for an activated node, its frozen bits can be decoded accurately without the help of frozen information. Then, the subtree of this node can be prunable. A prunable subtree recognizing scheme based on $\boldsymbol{H}$-Matrix check is proposed and a list decoding scheme is proposed based this. Results show that the decoding latency is reduced greatly compared with the exiting list decoder, and decoding latency decreases with the increasing of SNR. Furthermore, the latency of proposed algorithm is close to that of BP decoder.

Acknowledgements This work was supported in part by National Major Project (Grant No. 2016ZX03001011-005), National Natural Science Foundation Project (Grant No. 61521061), 333 Program of Jiangsu (Grant No. BRA2017366), and Intel.

References

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

2 Arikan E, Costello D J, Kliewer J, et al. Guest editorial recent advances in capacity approaching codes. IEEE J Sel Areas Commun, 2016, 34: 205–208

3 Zhang B J, Shen H, Yin B, et al. A 5G trial of polar code. In: Proceedings of IEEE Globecom Workshops (GC Wkshps), Washington, 2016

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

5 Arkan E. A performance comparison of polar codes and Reed-Muller codes. IEEE Commun Lett, 2008, 12: 447–449

6 Yuan B, Parhi K K. Architecture optimizations for BP polar decoders. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, 2013. 2654–2658

7 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

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

9 Abbas S M, Fan Y Z, Chen J, et al. Low complexity belief propagation polar code decoder. In: Proceedings of IEEE Workshop on Signal Processing Systems (SiPS), Hangzhou, 2015

10 Tal I, Vardy A. List decoding of polar codes. IEEE Trans Inf Theory, 2015, 61: 2213–2226

11 Niu K, Chen K. CRC-aided decoding of polar codes. IEEE Commun Lett, 2012, 16: 1668–1671

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

13 Sarkis G, Giard P, Vardy A, et al. Fast polar decoders: algorithm and implementation. IEEE J Sel Areas Commun, 2014, 32: 946–957

14 Sarkis G, Giard P, Vardy A, et al. Increasing the speed of polar list decoders. In: Proceedings of IEEE Workshop on Signal Processing Systems, Belfast, 2014

15 Sarkis G, Giard P, Vardy A, et al. Fast list decoders for polar codes. IEEE J Sel Areas Commun, 2016, 34: 318–328

16 Yuan B, Parhi K K. LLR-based successive-cancellation list decoder for polar codes with multibit decision. IEEE Trans Circuits Syst II, 2017, 64: 21–25

17 Xiong C R, Lin J, Yan Z Y. Symbol-decision successive cancellation list decoder for polar codes. IEEE Trans Signal Process, 2016, 64: 675–687

18 Xu Q Y, Pan Z, Liu N, et al. A complexity-reduced fast successive cancellation list decoder for polar codes. Sci China Inf Sci, 2018, 61: 022309

19 Li G P, Mu J J, Jiao X P, et al. Enhanced belief propagation decoding of polar codes by adapting the parity-check matrix. J Wirel Com Network, 2017, 2017: 40

20 Tal I, Vardy A. How to construct polar codes. IEEE Trans Inf Theory, 2013, 59: 6562–6582