International Journal of Innovative Computing, Information and Control
Volume **21**, Number **5**, October **2025**

ICIC International ©2025 ISSN 1349-4198
pp. **1317–1331**

RANDOMLY PENALIZED-AIDED BELIEF PROPAGATION FLIPPING DECODING OF POLAR CODES

Wenfan Wang[1], Jie Shen[2] and Guiping Li[3,*]

[1]School of Information Engineering
Henan Vocational College of Water Conservancy and Environment
No. 136, Huayuan Road, Zhengzhou 450008, P. R. China
wwf@hwec.edu.cn

[2]School of Mechanical Engineering
North China University of Water Resources and Electric Power
No. 36, Beihuan Road, Zhengzhou 450045, P. R. China
shenjie@ncwu.edu.cn

[3]School of Computer Science and Engineering
Xi'an Technological University
Jinhua Road, Xi'an 710021, P. R. China
*Corresponding author: liguiping@xatu.edu.cn

Received March 2025; revised June 2025

Abstract. *Belief propagation (BP) decoders of polar codes can run in parallel and have a remarkable throughput achievement at a cost of performance degradation, when compared with other decoding algorithms of polar codes. The existing improved algorithms still have a gap to be optimized in performance, complexity or latency. This paper proposes a modified BP flipping algorithm by introducing random penalty items into the reliability metrics of bits to correct un-converged errors under the type of noise with Gaussian distributions considered. Simulation results illustrate that the proposed penalizing method can exhibit significant performance gains over the original BP and other three existing BP flipping decoding algorithms with different blocks in the additive white Gaussian noise channel (AWGN) under a lower decoding latency. This shows the proposed scheme identifies the error-prone bits more efficiently.*

Keywords: Polar codes, Random penalty items, Belief propagation, Flipping decoding, Gaussian noise

1. **Introduction.** Polar codes [1], proposed by Arikan, have gained recognition for their rich algebraic structure, favorable analytical properties, regular coding framework, clear construction methodology, low complexity in encoding and decoding algorithms, and their flexible ability to construct codes of arbitrary length or rate. Consequently, polar codes are applicable in a variety of scenarios, including the Internet of Things (IoT), edge computing, distributed AI training, and robot communications [2], among others. For instance, polar codes can be employed to achieve a balance between performance and resource consumption for robot platforms with limited computational capabilities. To adapt polar codes to scenarios requiring ultra-reliability (e.g., a bit error rate of 10^{-6}) and ultra-low latency (in the sub-millisecond range), further optimization of their error-correction capabilities is necessary. In this paper, we explore methods to enhance the decoding performance of the flipping algorithm, which is based on BP decoding for polar codes. The BP algorithm is particularly adept at achieving low-latency and high-throughput, and it holds the potential to facilitate hardware support for polar codes.

DOI: 10.24507/ijicic.21.05.1317

While BP decoding serves as a significant alternative decoding method for polar codes, it is not without its shortcomings in terms of error-correction performance. We have observed that increasing the minimum distance of polar codewords is a key approach to enhancing the error-correcting capabilities of polar codes. Techniques such as BP list [3, 4] and BP flip [5-10] are examples of this. The BP list comprises multiple parallel independent BP decoders, each based on differently permuted factor graphs of polar codes. This results in a list of potential transmitted codewords, from which the one closest to the received vector in terms of Euclidean distance is selected. Although the BP list decoder currently offers the best performance for plain polar codes under iterative decoding, its high decoding complexity and latency make it unsuitable for certain real-time applications.

Given the success of the bit-flipping method in the BP decoder of Low Density Parity Check (LDPC) codes, its application to the polar BP decoder has also been explored [5]. Building on this reference, recent proposals have introduced additional methods to improve performance. This is because the BP flipping decoding algorithm shares the same space complexity as the original BP algorithm and the successive cancellation (SC) algorithm. In essence, the performance of BP flipping decoding hinges on the reliability metric of bits used to identify potential erroneous bits. Consequently, evaluating the reliability of polarized sub-channels and constructing the set of flipped bits are the central focuses of BP flipping decoders for polar codes. A BP flipping algorithm employing a stepping strategy was proposed in [6] to reduce the search space for flipping bits. The scheme presented in [7] leverages a convolutional neural network (CNN) to classify flipping bits and achieves performance gains with short block codes. All the aforementioned schemes are based on a single-bit flipping strategy. The multiple-bit flipping method under BP decoding proposed in [8] is also considered. However, this method necessitates the initiation of 2^m new branches in each additional decoding round, where m is the number of multiple-bit flips per round.

Although the aforementioned schemes improve the performance of the original BP decoding algorithm in terms of efficiency, complexity, or latency, there is still potential for optimization. This is because many bits in their flipped sets do not contribute to correcting error frames. Moreover, the process of constructing the flipped set is offline, relying solely on the likelihood ratio (LLR) magnitudes for the selection of erroneous bits. This necessitates more flipping attempts and hinders their efficiency.

Simulation results indicate that there are two primary types of errors in the decoding process of the BP flip algorithm. One error type occurs due to the oscillation of the leftmost likelihood values in the factor graph of polar codes, resulting in convergence failure. The other error type involves the values becoming trapped in a local maximum. Typically, to enhance the BP flipping decoding performance of polar codes, it is necessary to optimize the set of flip bits or the metric values of the least reliable bits.

Motivated by the random perturbation method proposed in [11], we explore the integration of randomly penalized elements into the flipping metric of BP decoding for polar codes, aiming to address the issue of spurious local maxima as discussed in this paper. Since the communication system is viewed as a non-linear system, adding random penalties, such as white noise, can help it reach a stable model [11]. Unlike [11], random penalty items are only utilized to refine the erroneous bit metric, not for every parallel decoding branch, thereby preventing a high level of computational complexity. In our simulations, Gaussian-distributed random noise is used as penalty items in AWGN channels. Theoretical analysis reveals that the random penalizing method can yield a higher probability of correct flipping when the penalty parameters are properly selected. Furthermore, simulation results also show that the frame error rate performance can be substantially improved.

The remainder of this paper is organized as follows. Section 2 reviews the polar codes, the original BP algorithm, and the BP flipping algorithm. Randomly penalized-aided BP flipping decoder is proposed in Section 3. Section 4 analyzes the decoding performance. Conclusions are drawn in Section 5.

2. Preliminaries.

2.1. **Polar codes.** The design inspiration for polar codes stems from a phenomenon of polarization, which reveals that the bit-channels of an infinite block length can be perfectly polarized into two states: either the channel capacity $I\left(W_N^{(i)}\right)$ approaches 0 or $I\left(W_N^{(i)}\right)$ approaches 1. As a result, the encoding process of polar codes entails selecting the good channels, whose capacities are near 1, to transmit the information bits, and the bad channels, whose capacities are close to 0, to transmit frozen bits for finite block lengths N. The encoding of the source bits sequence $\mathbf{u}_1^N$, which encompasses information bits u_A (A denotes the index set of the information bits) and frozen bits u_A^C (A^C is the complementary set of A), into a codeword sequence $\mathbf{x}_1^N$ can be expressed using the equation $\mathbf{x}_1^N = \mathbf{u}_1^N \mathbf{G}_N$. This is achieved with the aid of the generator matrix $\mathbf{G}_N$, which is derived from $\mathbf{B}_N \mathbf{F}^{\otimes n}$. Here, $\mathbf{B}_N$ represents the bit-reversal permutation, and $\mathbf{F}^{\otimes n} (n = \log N)$ signifies the n-th Kronecker power of $\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

2.2. **Belief propagation decoding.** BP decoding of polar codes is a message-passing algorithm that operates over a Forney-style factor graph (FFG), which corresponds to the $\mathbf{G}_N$ generator matrix. Moreover, it inherently enables soft-in/soft-out decoding, facilitating joint iterative detection and decoding tasks. For an (N, K) polar code, the FFG depicted in Figure 1 has $(\log N + 1)N$ nodes and $n = \log N$ stages. There are two kinds of LLR messages that propagate from the channel stage (the rightmost) to the information bit stage (the leftmost), and then pass from left to right, respectively. Each stage of the FFG has $N/2$ processing elements (PE). A PE, as shown in Figure 2, consists of four nodes. And each node is associated with two types of messages, namely, a right-to-left

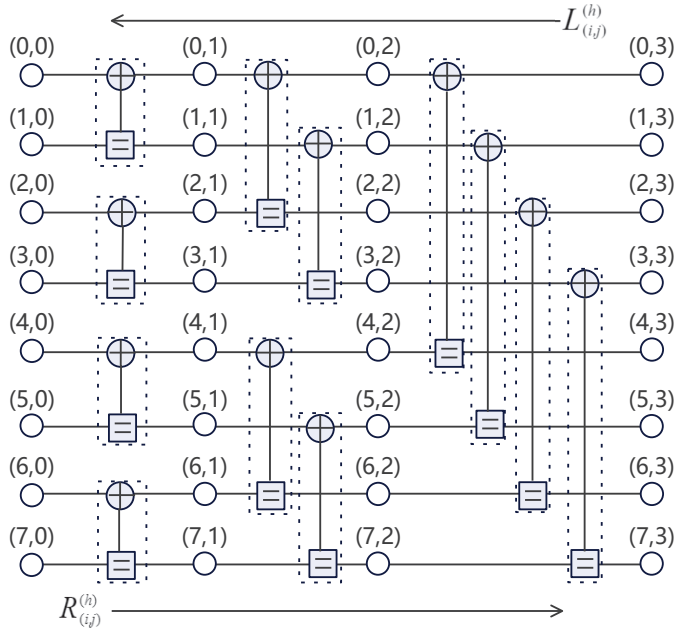

FIGURE 1. Factor graph of polar codes with the block length $N = 8$

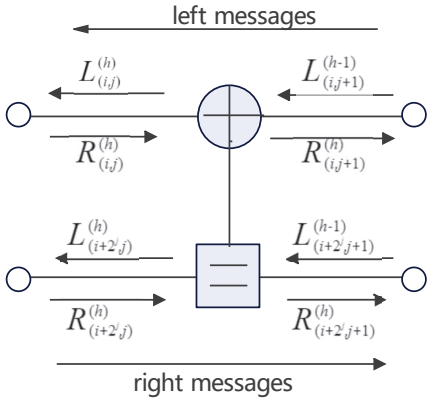

FIGURE 2. A processing element of factor graph for polar codes

message $L^{(h)}_{i,j}$ and a left-to-right message $R^{(h)}_{i,j}$. The updating rules for the two kinds of propagated messages with the number of iterations h are as follows:

$$\begin{cases} L^{(h)}_{(i,j)} = f\left(L^{(h-1)}_{(i,j+1)}, L^{(h-1)}_{(i+2^j,j+1)} + R^{(h)}_{(i+2^j,j)}\right), \\ L^{(h)}_{(i+2^j,j)} = f\left(L^{(h-1)}_{(i,j+1)}, R^{(h)}_{(i,j)}\right) + L^{(h-1)}_{(i+2^j,j+1)}, \\ R^{(h)}_{(i,j+1)} = f\left(L^{(h-1)}_{(i+2^j,j+1)} + R^{(h)}_{(i+2^j,j)}, R^{(h)}_{(i,j)}\right), \\ R^{(h)}_{(i+2^j,j+1)} = f\left(L^{(h-1)}_{(i,j+1)}, R^{(h)}_{(i,j)}\right) + R^{(h)}_{(i+2^j,j)}, \end{cases} \tag{1}$$

where the function $f(a,b) = \ln((1+xy)/(x+y))$ is also approximately equal to $f(a,b) \approx \alpha \text{sign}(a)\text{sign}(b)\min(|a|,|b|)$ in hardware implementation. And the initial values of $L^{(0)}_{(i,j)}$ and $R^{(0)}_{(i,j)}$ can be obtained by

$$L^{(0)}_{(i,n+1)} = \ln\frac{P(y_i|x_i=0)}{P(y_i|x_i=1)}, \tag{2}$$

$$R^{(0)}_{(i,0)} = \ln\frac{P(u_i=0)}{P(u_i=1)} = \begin{cases} 0 & \text{if } u_i \text{ is information bit} \\ \infty & \text{if } u_i \text{ is frozen bit} \end{cases}. \tag{3}$$

Finally, upon reaching the maximum number of iterations $I_{\max}$ or satisfying the early termination criterion, the BP decoder can determine the bit estimation sequence $\hat{u}_0^{N-1}$.

$$\hat{u}_i = \begin{cases} 0, & L^{(h)}_{(i,0)} + R^{(h)}_{(i,0)} \geq 0; \\ 1, & L^{(h)}_{(i,0)} + R^{(h)}_{(i,0)} < 0. \end{cases} \tag{4}$$

2.3. **BP flipping decoding.** BP flipping (BPF) decoding, inspired by successive cancellation flip (SCF) [12], can achieve greater performance gains than BP decoding while maintaining the parallel nature of the BP algorithm. BPF decoding involves identifying error-prone information bits using a metric method and constructing a flip set S that consists of T such bits. When the original BP decoding fails to pass the Cyclic Redundancy Check (CRC), BPF decoding is executed according to the following operation, which reassigns the correct message to each flipped information bit through a predefined number of additional decoding attempts.

$$R^{(0)}_{(i,0)} = \begin{cases} +\infty, & i \in (A \cap FS) \text{ and } \hat{u}_i = 1, \\ -\infty, & i \in (A \cap FS) \text{ and } \hat{u}_i = 0, \\ 0, & i \in A \backslash FS, \end{cases} \tag{5}$$

where the set FS represents the index set of flipped bits. Typically, the method for constructing the flip set FS involves identifying the indices with smaller absolute LLR metrics LM, as the magnitude of LLR often indicates the reliability of the bit estimation. If LM_i, defined as $LM_i = \left| L_{(i,0)}^{(I_{\max})} + R_{(i,0)}^{(I_{\max})} \right|$ for $i \in A$, is smaller, the estimation $\hat{u}_i$ is more prone to error.

Once the flipping set is established, candidate bits from this set are selected for flipping when BP fails to pass the CRC. Subsequently, BP decoding is performed once more. If the result meets the CRC criteria, the decoding process is concluded. If not, additional candidate bits in the flipping set are tried until either the CRC is passed or the predefined maximum number of flips is reached. Consequently, increased BP decoding attempts due to a large flipping set size can lead to higher latency in the polar decoder. Thus, the main research focus becomes how to select more accurate error-prone information bits for the flipping set, aiming to enhance decoding performance and reduce computational complexity. To address this, we propose a randomly penalized-aided BPF decoding method for polar codes, which balances the need for computational complexity with improved performance, without compromising the parallelism advantage of BP decoding.

3. Proposed Randomly Penalized-Aided BPF Decoding.

In this section, we first introduce the metrics used in the proposed scheme and then describe the method of appending random penalty items into the flipping metrics. Secondly, we illustrate the effects of the random penalty on the error rate performance and analyze the complexity increase caused by the random items. Finally, we provide a complete decoding process of randomly penalized BPF decoding.

3.1. Metrics for the BPF algorithm of polar codes.

The authors in [12] use the information bits with the smallest T values of LLR in a flipping set because the LLR is generally considered to reflect the reliability of information bits, as shown in Figure 3. However, in regions with a low signal-to-noise ratio (SNR) or for short block codes, some correct bit estimations also have small log-likelihood ratios (LLRs). In such cases, this metric might be unreliable.

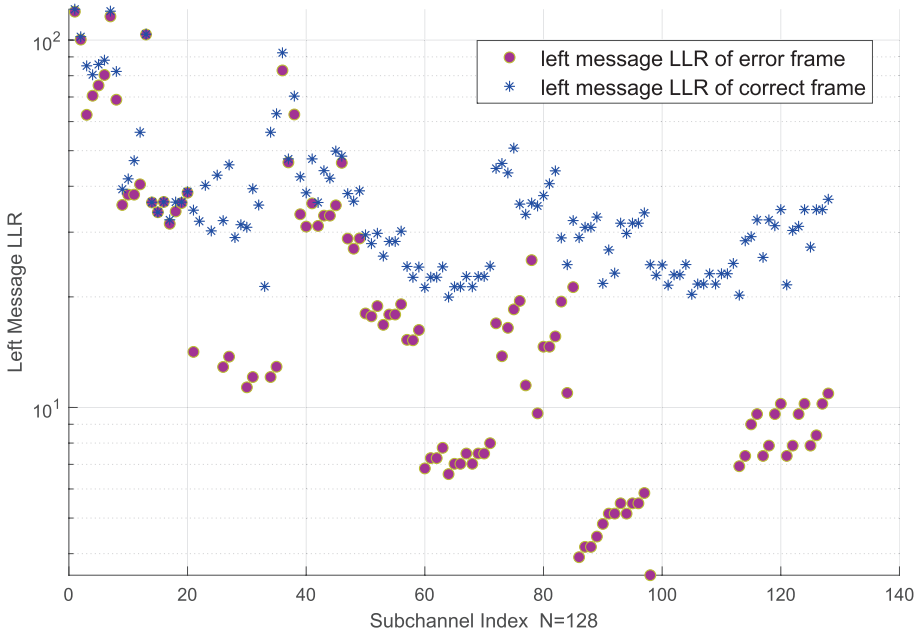

FIGURE 3. Left message LLRs of error bits and correct bits at 1.5dB

To more accurately identify the index of erroneous information estimations in the BP decisions $\hat{u}_0^{N-1}$, we propose a metric that utilizes the standard deviation of LLRs to optimize both frozen information processing elements (FIPEs)-based and decision-based metrics, as the standard deviation more accurately reflects the stability of data. Building upon the work in [10], the FIPE-based metric (FM) is defined as

$$FM_i = 1 - \frac{1}{I_{\max}} \sum_{h=1}^{I_{\max}} \gamma_i^{(h)}, \quad i \in A, \tag{6}$$

and

$$\gamma_i^{(h)} = g\left(L_{(i-1,1)}^{(h-1)}\right) \oplus g\left(L_{(i,1)}^{(h-1)}\right), \tag{7}$$

where $g(a) = \begin{cases} 0, & a \geq 0 \\ 1, & a < 0 \end{cases}$. Thus, the function $g(\cdot)$ acts as a sign function. The variable $\gamma_i^{(h)}$ serves as an indicator variable that signifies the detection result of the FIPE for the information bit u_i during the hth BP iteration. If $\gamma_i^{(h)} = 0$, we can infer that the estimation $\hat{u}_i$ is correct during the hth BP iteration. Conversely, if $\gamma_i^{(h)} = 1$, $\hat{u}_i$ is deemed incorrect and requires correction. Based on the aforementioned equations, FM_i reflects the likelihood of an erroneous estimation for the information bit u_i. Typically, the smaller the value of FM_i, the more prone the estimation $\hat{u}_i$ is to errors.

It is known that the BP decoder makes a decision based on the sign of $\left(L_{(i,0)}^{(h)} + R_{(i,0)}^{(h)}\right)$ for each information bit at the end of each iteration. To mitigate the effects of oscillating bits that periodically change their estimations, the variable DM can be used, defined by the following equation to represent the degree of divergence in estimating $\hat{u}_i$ [8]

$$DM_i = \frac{1}{I_{\max}} \left| \sum_{h=1}^{I_{\max}} \operatorname{sign}\left(L_{(i,0)}^{(h)} + R_{(i,0)}^{(h)}\right) \right|, \quad i \in A, \tag{8}$$

where DM_i is smaller, indicating a stronger disagreement among the BP iterations in estimating u_i, i.e., the estimation $\hat{u}_i$ is more error-prone. Since the DM of oscillating bits is close to 0, it is easy to distinguish the effect of oscillation or divergence.

Given that reliability and divergence both contribute to identifying the error-prone nature of bits from different perspectives, we can define the metric for whether the information bit needs to be flipped by considering both FM and DM.

$$WM_i = \eta FM_i + (1-\eta) DM_i, \quad i \in A, \tag{9}$$

where $WM_i \in [0,1]$ and $\eta \in [0,1]$. η represents a weight for trading off the importance of the metrics [10]. Specifically, a smaller WM_i signifies a lower reliability for the estimate $\hat{u}_i$. Therefore, the index with the smallest WM is prioritized for flipping. Additionally, a larger standard deviation suggests that the data set is more dispersed, which may indicate less reliability in the corresponding data. Consequently, in our work, we utilize the LLR standard deviation as a substitute for the LLR in the variables FM and DM. The LLR standard deviation for an information bit at the leftmost of the FFG is defined as

$$LLRstd_i = \sqrt{\frac{1}{I_{\max}} \sum_{h=1}^{I_{\max}} \left(L_{(i,0)}^{(h)} - AS_i\right)^2}, \tag{10}$$

where AS_i represents the average value of the ith information bit, calculated by considering all the left message $L_{(i,0)}$ of the information bit i over $I_{\max}$ iterations.

Utilizing the aforementioned WM and standard deviation, an enhanced flipped set with improved prediction accuracy for identifying erroneous positions can be achieved. The construction algorithm for the flipped set is outlined in Algorithm 1.

Algorithm 1: Construction algorithm of a flipped set (FS)

Input: A, $T_{\max}$, $LLRstd$
Output: FS

Initialize: $FS \leftarrow \phi$;
update $LLRstd$ and WM as in (10) and (9), respectively;
while $|FS| \leq T_{\max}$ **do**
  flipped bit $f \leftarrow \arg \min_{i \in A \setminus FS} \{WM_i\}$;
  if $WM_f \neq 1$ **then**
    $FS \leftarrow FS \cup \{f\}$;
  end
  else $FS \leftarrow FS \cup \left\{ \arg \min_{i \in A \setminus FS} \{LLRstd_i\} \right\}$;
end
Return FS;

where $T_{\max}$ denotes the number of flipping attempts.

3.2. **Randomly penalized flipping metrics.** In the metric denoted as WM, the estimation of an information bit is generally considered correct when $WM = 1$. However, this conclusion may not always hold true in the low-SNR regions or for short and moderate block lengths in polar codes. It is understood that the decoding result will be correct if the BP decoder converges within one iteration. Yet, the decoder can sometimes reach a local maximum and make an erroneous decision, as illustrated in Figure 4, because the estimation of an information bit might be deemed correct under the metric $WM = 1$.

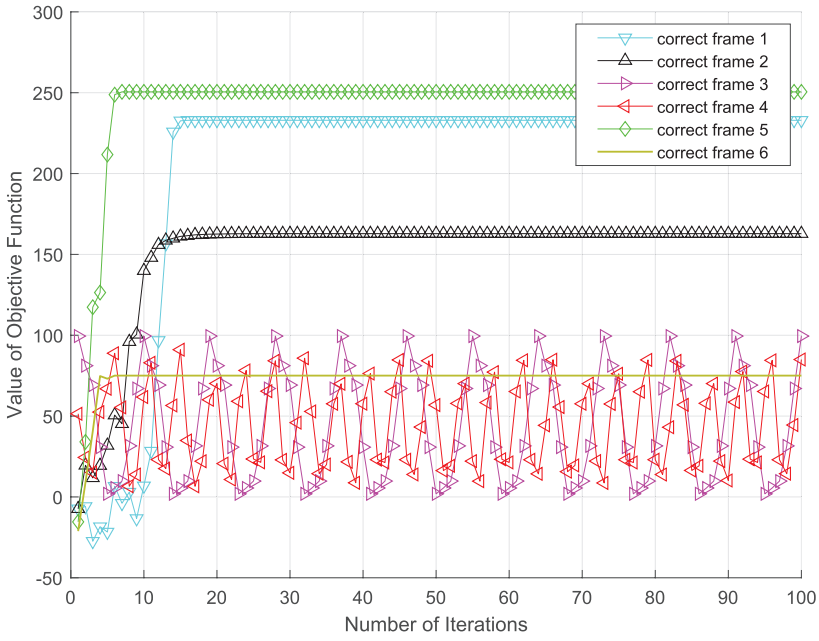

FIGURE 4. Value of objective function versus the number of iterations for a $(128, 64)$ polar code with error frames at $E_b/N_0 = 1.5$dB

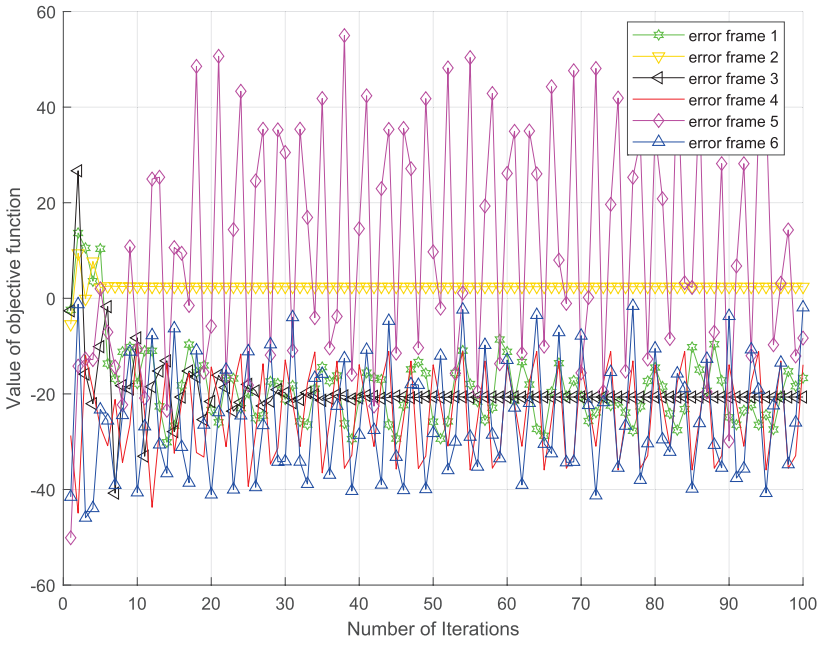

FIGURE 5. Value of objective function versus the number of iterations for a $(128, 64)$ polar code with correct frames at $E_b/N_0 = 1.5$dB

Conversely, even if the leftmost LLRs in the FFG of the BP decoder are still oscillating, the estimation sequence can pass the CRC as depicted in Figure 5, under the metric WM.

Based on the aforementioned analysis, we incorporate random penalties into the flipping metric at each iteration to assist the BP flipping algorithm in escaping undesirable local maxima, thereby enhancing error rate performance. Specifically, we modify the flipping metric WM in (9) by introducing a random penalty term as follows:

$$WM_i = 2 \times (\eta FM_i + (1-\eta)DM_i + noise \times Gauss())/stdev^2, \quad i \in A, \tag{11}$$

where

$$noise = (-10) \times \log\left(2 \times stdev^2\right) \tag{12}$$

and

$$stdev = \sqrt{\frac{1}{2 \times rate \times 10^{SNR/10}}}, \tag{13}$$

where $rate = N/|A|$, and it denotes the code rate.

During the BP flipping decoding process with Gaussian noise, the objective function values undergo a random number of large fluctuations as the number of iterations increases, and eventually, they emerge from these fluctuations. The simulation results, as depicted in Figure 6, indicate that incorporating random penalties into the flipping metric can enable the BP flipping algorithm to escape local maxima. This helps to minimize some previously undetectable errors. It becomes evident that the penalty-aided BP flipping algorithm significantly reduces the proportion of error patterns across the entire spectrum of Hamming weights. Consequently, the penalized flipping metrics aid in correcting segments that are prone to errors.

3.3. **Decoding steps for BPF aided by penalized.** In the proposed scheme, the original BP decoding is performed first. If the bit estimations $\hat{u}_0^{N-1}$ fail to pass the CRC detector and all BP iterations have been exhausted, the flipped set is generated using Algorithm 1. Subsequently, the bit flipping scheme is applied to the positions in the set

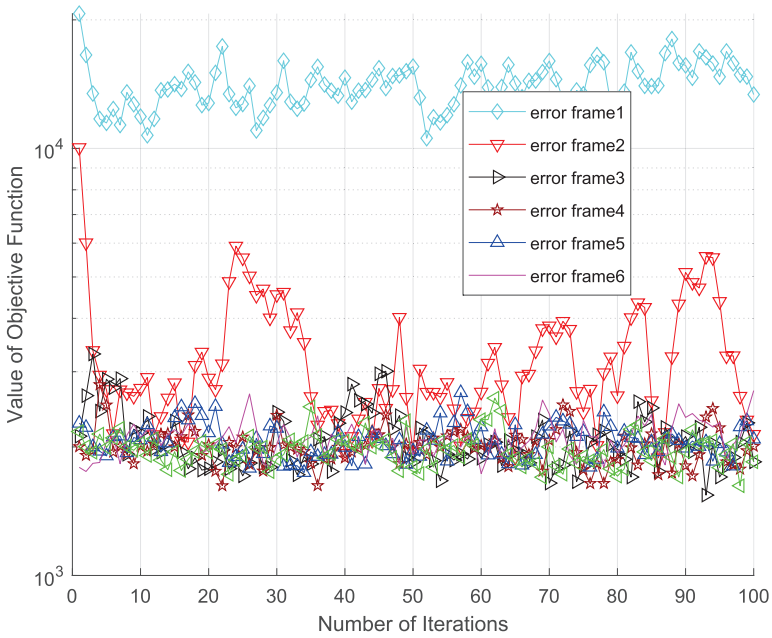

FIGURE 6. Value of objective function versus the number of iterations for a $(128, 64)$ polar code with error frames at $E_b/N_0 = 1.5$dB using the penalties-aided WM metric

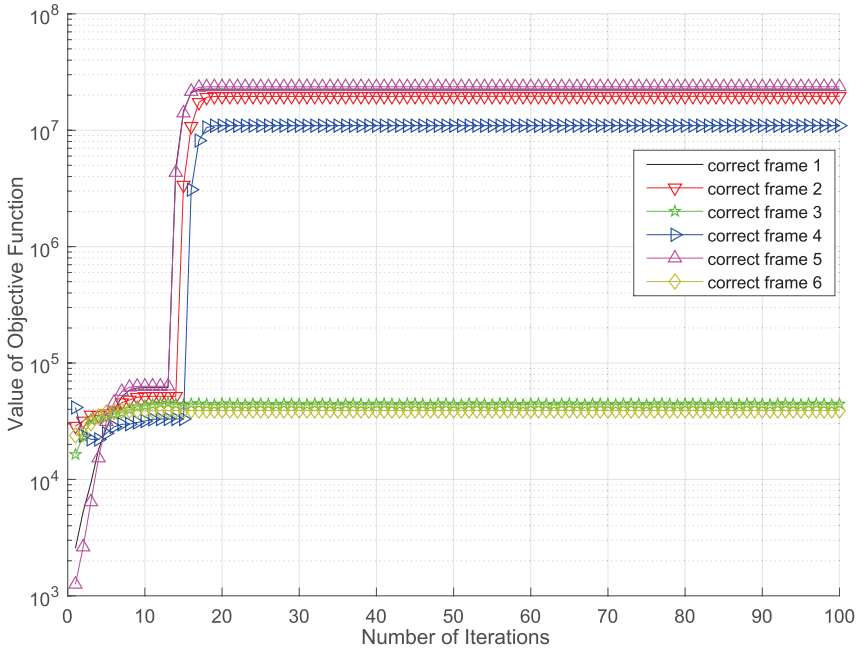

FIGURE 7. Value of objective function versus the number of iterations for a $(128, 64)$ polar code with correct frames at $E_b/N_0 = 1.5$dB using the penalties-aided WM metric

FS during each decoding attempt. The flipping operation is carried out by altering the prior right messages, the LLRs of the flipped information bits. The proposed decoder only terminates when the correct estimations $\hat{u}_0^{N-1}$ are obtained or all flipping attempts have been exhausted. The decoding process described above is summarized in Algorithm 2. Figure 8 further illustrates the entire decoding process.

Algorithm 2: Random penalties-aided BP flipping algorithm

Input: A, $I_{\max}$, $T_{\max}$, y_0^{N-1}, η
Output: $\hat{u}_0^{N-1}$

Initialize: $FS \leftarrow \phi$, $t \leftarrow 0$;
$\hat{u}_0^{N-1} \leftarrow BP\left(A, I_{\max}, y_0^{N-1}\right)$;
if $CRC\left(\hat{u}_0^{N-1}\right) = true$ **then**
  return $\hat{u}_0^{N-1}$;
end
else $FS \leftarrow$ perform Algorithm 1;
while $CRC\left(\hat{u}_0^{N-1}\right) = false$ and $t < T_{\max}$ **do**
  obtaining flipping index from FS;
  Assiging $R_{(i,0)}^{(0)}$ is $+\infty$ or $-\infty$;
  $\hat{u}_0^{N-1} \leftarrow BP\left(A, I_{\max}, y_0^{N-1}\right)$;
  if $CRC\left(\hat{u}_0^{N-1}\right) = true$ **then**
    return $\hat{u}_0^{N-1}$;
  end
  else $t \leftarrow t + 1$;
end
return $\hat{u}_0^{N-1}$;

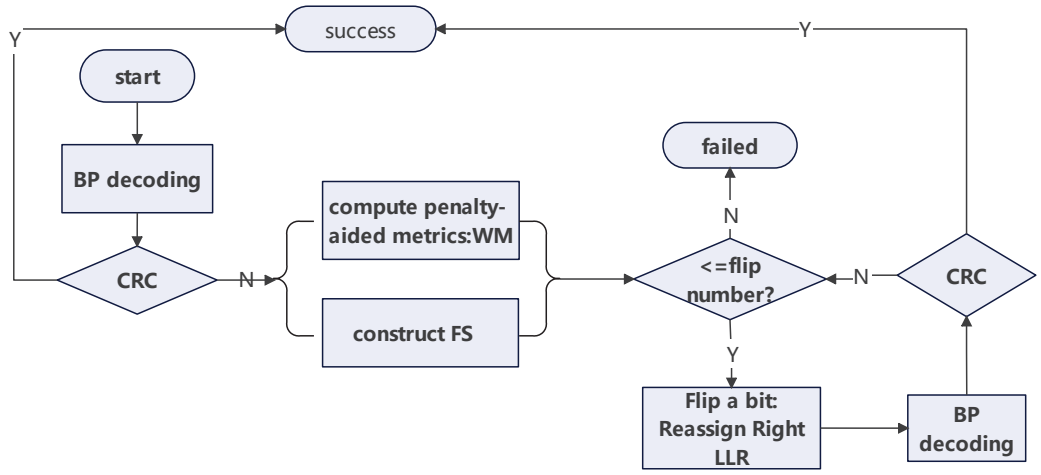

FIGURE 8. Whole decoding process of randomly penalized-aided BPF for polar codes

3.4. **Complexity analysis of the proposed scheme.** In this paper, we measure time complexity by the average number of clock cycles (CCs) required to decode one frame, as indicated by [10]. For the proposed penalty-aided BP flipping (PABF) decoding, it also requires $2n$ CCs to update the left message LLR and the right message LLR of a node at each iteration, as shown in (1). Additionally, one CC is required to compute the metrics for the FS construction, and two CCs to generate the set FS. Computing the LLR standard deviation requires another one CC. We use the variable I_{avg} to denote the average number of BP iterations in the entire decoding process. The average latency of the proposed scheme is

$$\tau_{PABF} = 2n \cdot I_{avg} + 4. \tag{14}$$

Specifically, despite the improvements in metrics for BP flipping in our work, the worst-case decoding failures for all flipping attempts persist. Consider that each frame incorrectly decoded in the original BP decoding requires at most $T_{\max}$ additional flipping attempts. If the FER of the original BP decoder is $P_e^{BP}(SNR)$ at a given SNR, the average number of iterations in the worst-case for PABF decoding for an (N, K) polar code can be calculated as

$$I_{worst} = I_{\max}\left(1 + T_{\max}P_e^{BP}(SNR)\right). \tag{15}$$

Based on the average latency and the average number of iterations, we can further deduce the worst-case decoding latency of the proposed scheme as follows:

$$\tau_{worst\text{-}PABF} = 2n \cdot I_{\max}\left(1 + T_{\max}P_e^{BP}(SNR)\right) + 4. \tag{16}$$

The computational complexity of the proposed scheme encompasses both the complexity of BP decoding and that of constructing the set FS. It is established that the complexity of BP decoding amounts to $2N\log N \cdot I_{avg}$. The computational complexity for calculating the LLR standard deviation is $2N$. Consequently, the average complexity of BP decoding remains defined as $2N\log N \cdot I_{avg}$. Meanwhile, the complexity involved in selecting the $T_{\max}$ smallest elements from the set FS of size M is $T_{\max}(M-1)P_e^{BP}(SNR)$. Drawing from the aforementioned analysis, the average complexity of PABF decoding is

$$C_{PABF} = 2N\log N \cdot I_{avg} + T_{\max}(M-1)P_e^{BP}(SNR), \tag{17}$$

and the worst-case complexity of the proposed scheme is

$$C_{worst\text{-}PABF} = 2N\log N I_{\max}\left(1 + T_{\max}P_e^{BP}(SNR)\right) + T_{\max}(M-1)P_e^{BP}(SNR). \tag{18}$$

The time and computational complexities of the proposed scheme both indicate that the additional complexity introduced by the modifications to the flipping metrics is not significant compared to other existing BP flipping algorithms. On the contrary, it can reduce the number of flips and cause the BP decoder to accelerate convergence to the correct value, as demonstrated by simulations.

4. Simulation Results.

In this section, we compare the FER performance, decoding latency, and complexity among the proposed scheme, the original BP decoding, an enhanced BPF (EBPF) [10], the state-of-the-art BP correction (BPC) decoding [13], and the generalized BPF (GBPF) [14] for polar codes under $I_{\max} = 100$. The $T_{\max}$ values are fixed and are respectively equal to 20 and 30 for $(256, 128)$ and $(1024, 512)$ polar codes, and the 16-bit CRC is used. The difference between the aforementioned decoding schemes lies in the different metrics used to identify possible erroneous positions. Specifically, the flipping set of GBPF decoding is composed of the positions of bits with the smallest LLR magnitudes. In contrast, EBPF decoding introduces a weighted metric by exploiting the reliability and divergence of bit estimations during BP decoding to construct the flipping set. For BPC, it focuses on the prior knowledge of code bits in the factor graph rather than the information bits to identify the prone error bits.

Simulations are performed using binary phase shift keying (BPSK) modulation under an AWGN channel. In our work, we employ Gaussian approximation (GA) [15] with a design-SNR of 2.5dB. The parameter η is set to 0.55 for improved performance, achieved through a one-dimensional search.

4.1. Error rate performance of comparisons.

Figure 9 depicts the decoding FER performance of the proposed PABF scheme for the $(256, 128)$ polar code, with $T_{\max} = 20$. It involves sequentially flipping all the information bits within the set FS. As depicted in Figure 9, the decoding performance of the PABF algorithm, which employs both the WM and penalties, surpasses that of the EBPF algorithm, which relies solely on the WM

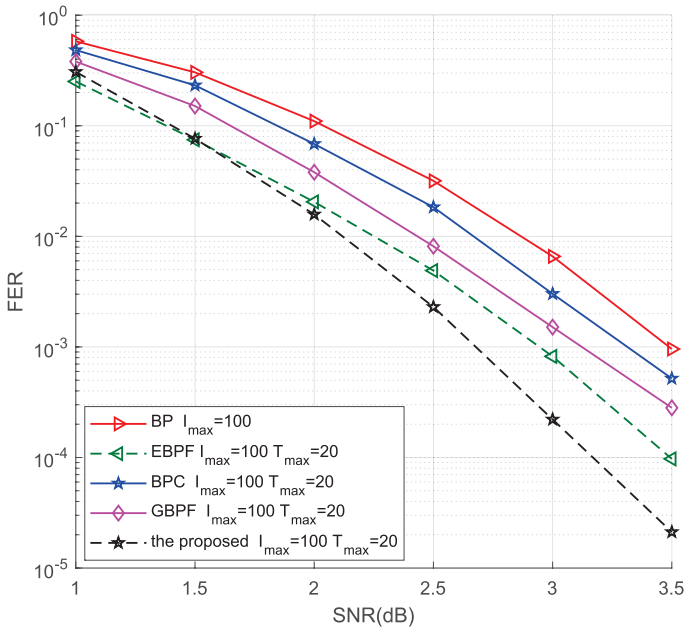

FIGURE 9. Decoding performance of various algorithms for the $(256, 128)$ polar code

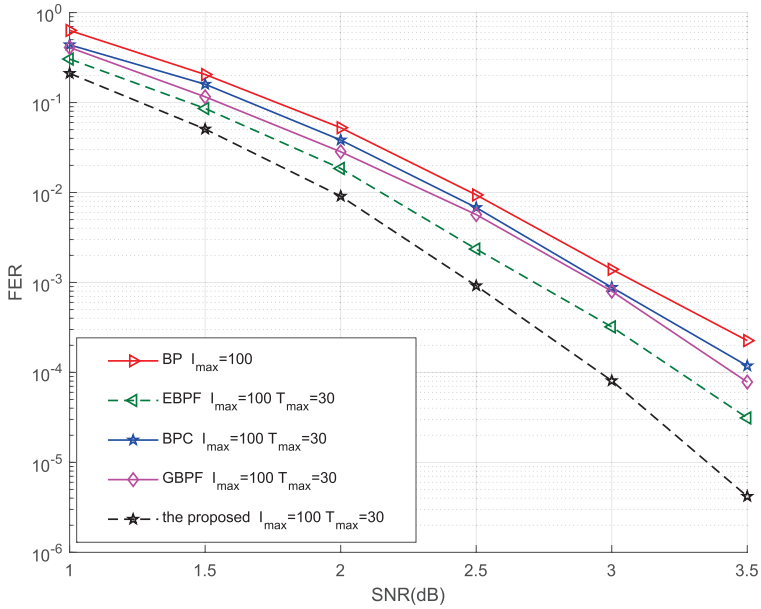

FIGURE 10. Decoding performance of various algorithms for the $(1024, 512)$ polar code

as a metric. It is also evident that at an FER of 10^{-3}, the proposed decoding method achieves performance gains of approximately 1.6dB, 0.45dB, 0.7dB, and 1.0dB over the BP decoding, EBPF decoding, GBPF decoding, and BPC decoding, respectively. Figure 10 illustrates the performance of the $(1024, 512)$ polar code under various decoding schemes, with $T_{\max} = 30$. It is clear that the proposed scheme also exhibits superior decoding performance compared to other algorithms. For instance, at an FER of 10^{-3}, there are performance gains of approximately 0.6dB, 0.3dB, 1.0dB, and 1.0dB, respectively. Based on the aforementioned simulation results, it is evident that the use of artificial noise is effective for the BP flipping decoding algorithm in the AWGN channel. However, when

compared to the SC list [16], which is assisted by CRC and offers the best decoding performance for polar codes but comes with increased decoding latency and complexity due to maintaining a list of up to L candidate bit sequences, the proposed scheme still exhibits a significant performance gap. Therefore, it is desirable to develop more robust flipping decoding algorithms to narrow this performance gap.

4.2. **Convergence rate and complexity comparisons.** Based on the above analysis, the convergence rate of the decoder is most affected by the number of frames that need to be further decoded in flipping trials and the number of flipping attempts. Figure 11 illustrates the average latency of the proposed scheme and different schemes for the $(1024, 512)$ polar code. It is evident that the proposed scheme requires fewer CCs ($T_{\max} = 70$) compared to both the EBPF decoding ($T_{\max} = 70$) and GBPF ($T_{\max} = 117$), while maintaining a superior FER performance. This indicates that the EBPF and GBPF algorithms necessitate a larger number of operations compared to the proposed algorithm. The gap appears more significant in low-SNR regions. The results demonstrate that the proposed scheme can construct a set FS with higher accuracy, thereby reducing the number of flipping attempts. Consequently, the proposed scheme converges at a faster rate. From the above Equations (17) and (18), it is apparent that with the same $T_{\max}$, the proposed decoding has almost the same complexity as that of EBPF decoding, and yields a lower complexity than that of BPC decoding.

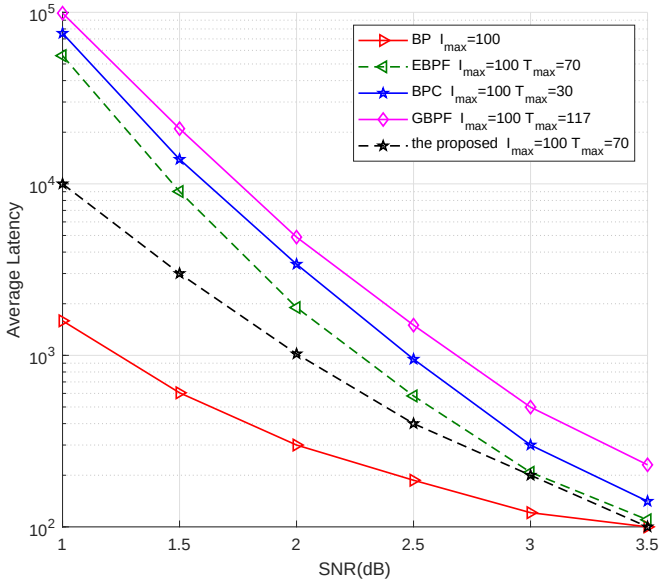

FIGURE 11. Average latency in decoding the $(1024, 512)$ polar code

5. **Conclusions.** In this paper, we propose a random penalized-aided BP flipping algorithm for polar codes. By introducing Gaussian noise into the flipping metrics, the proposed algorithm efficiently corrects errors that the EBPF, BPC, and GBPF algorithms cannot detect. Simulation results indicate that incorporating random penalty items significantly enhances error rate performance at high SNRs. Additionally, the proposed scheme requires fewer operations because the added penalties enable the decoder to converge slightly faster than other versions at low SNRs. Although our scheme achieves performance gains compared to the EBPF, BPC, and GBPF algorithms, it still lags behind the SCL-CRC decoding in performance. Consequently, we plan to investigate a multi-bits flipping strategy based on random penalty items to further enhance decoding performance in future work.

Acknowledgment. This work has received partial support from the Key Research Projects of Higher Education Institutions in Henan Province, under grant number 25A413012; in part from the Science and Technology Plan Project of Weiyang District in Shaanxi Province, under Grant 202424; in part from the Key Research and Development General Project of Shaanxi Provincial Science and Technology Program, under Grant 2025CY-YBXM-006; and in part from the Scientific and Technological Project of Henan Province, under grant number 252102211010. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] E. Arikan, Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels, *IEEE Trans. Information Theory*, vol.55, no.7, pp.3051-3073, 2009.

[2] R. Joshi and J. Sattar, One-shot gesture recognition for underwater diver-to-robot communication, *arXiv Preprint*, arXiv: 2503.00676, 2025.

[3] A. Elkelesh, M. Ebada, S. Cammerer and S. T. Brink, Belief propagation list decoding of polar codes, *IEEE Communications Letters*, vol.22, no.8, pp.1536-1539, 2018.

[4] H. Liu, E. Gunawan, Y. Hu and Y. L. Guan, BP-based sparse graph list decoding of polar codes, *IEEE Communications Letters*, vol.27, no.5, pp.1257-1261, 2023.

[5] Y. Yu, Z. Pan, N. Liu and X. You, Belief propagation bit-flip decoder for polar codes, *IEEE Access*, vol.7, pp.10937-10946, 2019.

[6] X. Zhang, Y. Liu, C. Chen et al., An enhanced belief propagation flipping decoder for polar codes with stepping strategy, *Entropy*, vol.24, 1073, 2022.

[7] X. Zhang, Y. Qiu, W. Kong et al., BP flip decoding algorithm of polar code based on convolutional neural network, *2022 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Sanshui, Foshan, China, pp.444-449, 2022.

[8] X. Wu, W. Wei, S. Zhang et al., An efficient belief propagation list flip decoder for polar codes, *IEEE Communications Letters*, vol.27, no.9, pp.2264-2268, 2023.

[9] Y. Mao, D. Yang et al., Improved segmented belief propagation list decoding for polar codes with bit-flipping, *China Communications*, vol.21, no.3, pp.19-36, 2024.

[10] Z. Yang and L. Chen, An enhanced belief propagation decoding algorithm with bit-flipping for polar codes, *IEEE Communications Letters*, vol.29, no.2, pp.348-352, 2025.

[11] A. Ç. Arlı and O. Gazi, Noise-aided belief propagation list decoding of polar codes, *IEEE Communications Letters*, vol.23, no.8, pp.1285-1288, 2019.

[12] O. Afisiadis, A. Balatsoukas-Stimming et al., A low complexity improved successive cancellation decoder for polar codes, *The 48th Asilomar Conference on Signals, Systems and Computers*, pp.2116-2120, 2014.

[13] M. Zhang, Z. Li and L. Xing, An enhanced belief propagation decoder for polar codes, *IEEE Communications Letters*, vol.25, no.10, 2021.

[14] Y. Shen, W. Song, H. Ji et al., Improved belief propagation polar decoders with bit-flipping algorithms, *IEEE Trans. Communications*, vol.68, no.11, pp.6699-6713, 2020.

[15] P. Trifonov, Efficient design and decoding of polar codes, *IEEE Trans. Communications*, vol.60, no.11, pp.3221-3227, 2012.

[16] I. Tal and A. Vardy, List decoding of polar codes, *IEEE Trans. Information Theory*, vol.61, no.5, pp.2213-2226, 2015.

Author Biography

Wenfan Wang received the B.S. degree from Henan Normal University, China in 2004, the M.S. degree from Zhengzhou University, China in 2007. She is currently an associate professor in School of Information Engineering, Henan Vocational College of Water Conservancy and Environment, China. Her current research interests include artificial intelligence, big data and computer applications.

Jie Shen received the B.S. degree in 2004 and the M.S. degree in 2007 from Zhengzhou University, China. He is currently a teacher in the School of Mechanical Engineering, North China University of Water Resources and Electric Power, China. His current research interests include signal acquisition and processing, embedded system and its application.

Guiping Li received the B.S. degree from Zhengzhou University, China in 2000, the M.S. degree from Xi'an University of Science and Technology, China in 2006, and the Ph.D. degree from Xidian University, China in 2016. She is currently an associate professor in School of Computer Science and Engineering, Xi'an Technological University, China. Her current research interests include communication information theory, channel coding theory and their applications.