

# Improved Iterative Hard- and Soft-Reliability Based Majority-Logic Decoding Algorithms for Non-Binary Low-Density Parity-Check Codes

Chenrong Xiong and Zhiyuan Yan

**Abstract**—Non-binary low-density parity-check (LDPC) codes have some advantages over their binary counterparts, but unfortunately their decoding complexity is a significant challenge. The iterative hard- and soft-reliability based majority-logic decoding algorithms are attractive for non-binary LDPC codes, since they involve only finite field additions and multiplications as well as integer operations and hence have significantly lower complexity than other algorithms. In this paper, we propose two improvements to the majority-logic decoding algorithms. Instead of the accumulation of reliability information in the existing majority-logic decoding algorithms, our first improvement is a new reliability information update. The new update not only results in better error performance and fewer iterations on average, but also further reduces computational complexity. Since existing majority-logic decoding algorithms tend to have a high error floor for codes whose parity check matrices have low column weights, our second improvement is a re-selection scheme, which leads to much lower error floors, at the expense of more finite field operations and integer operations, by identifying periodic points, re-selecting intermediate hard decisions, and changing reliability information.

**Index Terms**—Error control codes, non-binary low-density parity-check codes, decoding, error floor, complexity

## I. INTRODUCTION

Low-density parity-check (LDPC) codes were first developed by Gallager [1] in 1963. They were forgotten until they were rediscovered in the late 1990s by MacKay and Neal [2]. Since then, the academic and industrial communities have focused on binary LDPC codes, because long binary LDPC codes can achieve performance approaching the Shannon limit (see, for example, [3]). Hence binary LDPC codes have been used in various applications, such as digital television [4], Ethernet [5], home networking [6], and Wi-Fi [7]. Efficient decoding algorithms, encoder implementations, and decoder implementations of binary LDPC codes (see, for example, [8]–[15]) have received significant attentions.

In 1998, the study of Davey and MacKay [16] showed that non-binary LDPC codes over  $\text{GF}(q)$  ( $q > 2$ ) perform better than their binary counterparts for moderate code lengths. Moreover, non-binary LDPC codes also outperform binary LDPC codes on channels with bursty errors and high-order modulation schemes [17]. These advantages have motivated a steady stream of work on code designs [18]–[20], decoding algorithms [16], [17], [21]–[27], and decoder implementations [28]–[30] for non-binary LDPC codes. Davey and MacKay

[16] first used belief propagation (BP) to decode non-binary LDPC codes. By applying the fast Fourier transform (FFT) of probabilities to the BP algorithm, they also proposed a fast Fourier transform (FFT) based  $q$ -ary sum-product algorithm (SPA), called FFT-QSPA [22]. The FFT-QSPA was further improved by Barnault and Declercq [23]. Song and Cruz proposed a logarithm domain FFT-BP algorithm [17]. The Min-Sum algorithm was applied to non-binary LDPC codes by Wymeersch *et al.* [24]. Then Declercq and Fossorier [25] proposed the Extended Min-Sum (EMS) algorithm by using only a limited number of probabilities in the messages at inputs of check nodes. Savin [26] proposed the Min-Max algorithm.

Advantages of non-binary LDPC codes come at the expense of significantly higher decoding complexity than their binary counterparts. Since complexity of decoding non-binary LDPC codes is a key challenge, the iterative hard- and soft-reliability based majority-logic decoding, referred to as IHRB-MLGD and ISRB-MLGD<sup>1</sup>, respectively, algorithms [27] are particularly attractive. Based on the one-step majority logic decoding, these majority-logic decoding algorithms represent reliability information with finite field elements and integers, and hence involve only finite field additions (FAs) and finite field multiplications (FMs) as well as integer additions (IAs), integer comparisons (ICs), integer multiplications (IMs) and integer divisions (IDs). As a result, they require much lower computational complexities at the expense of moderate error performance degradation. For instance, while the error performance of the ISRB algorithm is within 1 dB of that of FFT-QSPA [23], its complexity is only a small fraction of that of the latter [27]. With a performance loss of 1 dB, the IHRB algorithm has even lower complexity than the ISRB algorithm [27]. Based on the IHRB algorithm, Zhang *et al.* [30] proposed an enhanced IHRB-MLGD (EIHRB) algorithm by introducing the soft-reliability initialization and re-computing the extrinsic information. The EIHRB algorithm has a similar complexity to that of the IHRB algorithm, but its error performance approaches that of the ISRB algorithm. The majority-logic decoding algorithms are particularly effective for LDPC codes constructed based on finite geometries and finite fields [19], [20].

The main contributions of this paper are two improvements to the majority-logic decoding algorithms.

- The first improvement is a new reliability information

This work was supported in part by NSF under Grants ECCS-0925890 and ECCS-1055877.

<sup>1</sup>When there is no ambiguity, MLGD is omitted when referring to majority-logic decoding algorithms for brevity.

update, instead of the accumulation of reliability information used in existing majority-logic decoding algorithms.

- Since existing majority-logic decoding algorithms tend to have a high error floor for codes whose parity check matrices have small column weights, our second improvement is a re-selection scheme, which lowers error floors at the expense of more finite field operations and integer operations by identifying periodic points, re-selecting intermediate hard decisions, and changing reliability information.

In the ISRB and IHRB algorithm, the reliability information includes all check-to-variable (c-to-v) messages of previous iterations. The new reliability information update proposed in this paper excludes the c-to-v messages of previous iterations. It not only results in better error performance and fewer iterations on average, but also greatly reduces computational complexities of *all* existing majority-logic decoding algorithms. For instance, when applied to the ISRB majority-logic decoding algorithm, the new reliability information update results in a 0.15 dB coding gain and reduces required number of iterations by 10% at 4.7 dB for a (16, 16)-regular (255, 175) cyclic LDPC code over  $\text{GF}(2^8)$  constructed with the method as describe in [19, Example 4]. Also, at a block error rate (BLER) of  $10^{-4}$ , the coding gain over the EIHRB algorithm is about 0.07 dB. At the SNR of 4.7 dB, the average number of iterations is reduced by about 25%. Furthermore, with the new reliability information update, the improved algorithms require significantly fewer IAs and ICs than the ISRB and EIHRB algorithms. Finally, the existing majority-logic decoding algorithms are based on the accumulation of reliability information, and hence the numerical range of the reliability information increases with iterations. In contrast, the proposed reliability information update results in a fixed numerical range and thus simplifies hardware implementations. Our new reliability update has been presented in part in [31]. By applying both the layered scheduling and our first improvement to the IHRB algorithm, we proposed a layered improved IHRB decoder with a high throughput in [32]. Because the architecture design of non-binary LDPC decoders is beyond the scope of this paper, we will not discuss the layered improved IHRB decoder henceforth.

In the literature, to analyze the error floor of binary LDPC codes, some notions based on graphical structures have been introduced, such as stopping sets [33], trapping sets [34] and absorbing sets [35]. Unfortunately, trying to lower the error floor based on graphical structures usually incurs very high complexity. Also, some approaches for binary LDPC codes cannot be readily adapted to non-binary ones. For instance, a selective biasing postprocessing algorithm is proposed in [35] to lower the error floors of binary LDPC codes based on the relaxed graphical structure of absorbing sets. However, for non-binary LDPC codes, trapping sets are difficult to identify because they involve not only the graph topology but also values of non-zero entries of parity-check matrices [36]. Moreover, the biasing rule between two elements for binary LDPC codes cannot be applied to non-binary codes directly, because there are more than two elements in a non-binary

finite field.

In this paper, for the majority-logic decoding algorithms, we propose a re-selection scheme based on periodic points to lower the error floors. The re-selection scheme is not a postprocessing algorithm and can be integrated into the regular iteration procedure easily. For instance, for an (837, 726) non-binary quasi-cyclic LDPC code over  $\text{GF}(2^5)$  constructed with the method in [20] with a column weight of four, the EIHRB algorithm has a BLER floor around  $10^{-3}$ , while the hard-reliability based algorithm with the new reliability information update and the re-selection scheme achieves a BLER floor below  $10^{-5}$ . Although this re-selection scheme requires additional computation, it is used only when existing majority-logic decoding algorithms have a high error floor.

The rest of our paper is organized as follows. Section II reviews existing majority decoding algorithms. Section III proposes the two improvements. In Section IV-A, the two improvements are applied to existing majority decoding algorithms to illustrate their advantages in error performance and average numbers of iterations. Section IV-B discusses the reduction in the computational complexities due to the two improvements. Some conclusions are given in Section V.

## II. EXISTING MAJORITY DECODING ALGORITHMS

A regular LDPC code  $\mathcal{C}$  of length  $N$  over a finite field  $\text{GF}(2^r)$  is the null space of an  $M \times N$  sparse parity check matrix  $\mathbf{H}$  over  $\text{GF}(2^r)$ .  $\mathbf{H}$  has constant column and row weights of  $\gamma$  and  $\rho$ , respectively. Let  $\mathbf{h}_0, \mathbf{h}_1, \dots, \mathbf{h}_{M-1}$  denote the rows of  $\mathbf{H}$ , where  $\mathbf{h}_i = (h_{i,0}, h_{i,1}, \dots, h_{i,N-1})$  for  $0 \leq i < M$ . Let  $(a_{l,0}, a_{l,1}, \dots, a_{l,r-1})$  be the binary representation of  $a_l \in \text{GF}(2^r)$ , for  $0 \leq l < 2^r$ . Suppose a codeword  $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})$  is transmitted. Since  $x_i \in \text{GF}(2^r)$  can be represented by an  $r$ -tuple  $(x_{i,0}, x_{i,1}, \dots, x_{i,r-1})$  over  $\text{GF}(2)$  for  $0 \leq i < N$ , an  $Nr$ -tuple over  $\text{GF}(2)$  is transmitted for each codeword. Assume the BPSK modulation is used: “0” is mapped to +1 and “1” to -1. Let  $\mathbf{y} = (y_0, y_1, \dots, y_{N-1})$  represent the received word, and  $\mathbf{z} = (z_0, z_1, \dots, z_{N-1})$  and  $\mathbf{q} = (q_0, q_1, \dots, q_{N-1})$  represent the hard decision and quantization, respectively, of the received word. Let  $\mathcal{N}(i) = \{j : h_{i,j} \neq 0, 0 \leq j < N\}$  for  $0 \leq i < M$  and  $\mathcal{M}(j) = \{i : h_{i,j} \neq 0, 0 \leq i < M\}$  for  $0 \leq j < N$ .  $I_{\max}$  represents the maximal iteration number.

### A. ISRB algorithm

The ISRB algorithm [27] is described in Alg. 1, where  $\lambda$  is a parameter to improve the error performance.  $\mathbf{s}^{(k)}$  is the syndrome vector corresponding to  $\mathbf{z}^{(k)}$ ,  $\varphi_{j,l}$  a channel reliability of the  $j$ -th received symbol being  $a_l$ .  $\phi_{i,j}$  and  $\sigma_{i,j}^{(k)}$  are the extrinsic weighting coefficient and the extrinsic check sum of the  $k$ -th iteration, respectively, from check node  $i$  to variable node  $j$ .  $\psi_{j,l}^{(k)}$  is the extrinsic reliability of the  $j$ -th received symbol being  $a_l$  in the  $k$ -th iteration.

In the ISRB algorithm, line 21 is an accumulation operation. Hence, the reliability  $R_{j,l}^{(k)}$  is a non-decreasing function of  $k$  as  $\psi_{j,l}^{(k)}$  is non-negative. To perform the ISRB algorithm correctly,  $R_{j,l}^{(k)}$  must be kept from numerical saturation based on two

**Algorithm 1:** ISRB algorithm [27]

```

/* -----Initialization----- */
1 for j = 0 : (N - 1) do
2   z_j^{(0)} = z_j;
3   for l = 0 : (2^r - 1) do
4     phi_{j,l} = sum_{t=0}^{r-1} (1 - 2a_{l,t}) q_{j,t};
5     R_{j,l}^{(0)} = lambda phi_{j,l};
6 for i = 0 : (M - 1) do
7   for j in N(i) do
8     phi_{i,j} = min_{t in N(i) \setminus {j}} max_l phi_{t,l};
/* -----Iteration----- */
9 for k = 0 : I_max do
10  s^{(k)} = H * (z^{(k)})^T;
11  if s^{(k)} == 0 then return z^{(k)} else if k == I_max
    then return Failure else
12    for j = 0 : (N - 1) do
13      for l = 0 : (2^r - 1) do
14        psi_{j,l}^{(k)} = 0;
15      for i in M(j) do
16        sigma_{i,j}^{(k)} = h_{i,j}^{-1} sum_{t in N(i) \setminus {j}} h_{i,t} z_t^{(k)};
17        for l = 0 : (2^r - 1) do
18          if sigma_{i,j}^{(k)} == a_l then
19            psi_{j,l}^{(k)} = psi_{j,l}^{(k)} + phi_{i,j}
19  for j = 0 : (N - 1) do
20    for l = 0 : (2^r - 1) do
21      R_{j,l}^{(k+1)} = R_{j,l}^{(k)} + psi_{j,l}^{(k)};
22  z_j^{(k+1)} = arg_{a_l} max R_{j,l}^{(k+1)};

```

methods. One is to use a very large numerical range for  $R_{j,l}^{(k)}$ , and the other is to carry out the following clipping operation [27]:

$$R_{j,l}^{(k)} = \begin{cases} -\eta & \text{if } R_{j,l}^{(k)} < R_{j,max}^{(k)} - 2\eta \\ R_{j,l}^{(k)} - R_{j,max}^{(k)} + \eta & \text{otherwise} \end{cases} \quad (1)$$

Here,  $R_{j,max}^{(k)} \triangleq \max_l (R_{j,l}^{(k)})$  and  $\eta$  is the predefined maximal value of  $R_{j,l}^{(k)}$  after the clipping operation.

**B. IHRB algorithm**

When the soft-reliability information of the received word is not available to the decoder, the IHRB algorithm [27] can be used. The iteration procedure of the IHRB algorithm is the same as that of the ISRB algorithm, but the IHRB algorithm has a different initialization step, described in Alg. 2, where  $\lambda_h$  is a parameter to improve the error performance.

**C. EIHRB algorithm**

The EIHRB algorithm [30], described by Alg. 3, was devised based on the IHRB algorithm by introducing a soft-

**Algorithm 2:** Initialization of the IHRB algorithm [27]

```

1 for j = 0 : (N - 1) do
2   z_j^{(0)} = z_j;
3   for l = 0 : (2^r - 1) do
4     if (a_l == z_j) then R_{j,l}^{(0)} = lambda_h else R_{j,l}^{(0)} = 0
5 for i = 0 : (M - 1) do
6   for j in N(i) do
7     phi_{i,j} = 1;

```

reliability initialization and recalculating the extrinsic information.  $c_1$  and  $c_2$  are two parameters to improve the error performance.

**Algorithm 3:** EIHRB algorithm [30]

```

/* -----Initialization----- */
1 for j = 0 : (N - 1) do
2   z_j^{(0)} = z_j;
3   z_{i,j}^{(0)} = z_j;
4   for l = 0 : (2^r - 1) do
5     phi_{j,l} = sum_{t=0}^{r-1} (1 - 2a_{l,t}) q_{j,t};
6     R_{j,l}^{(0)} = max(phi_{j,l}/c_1 + c_2 - max_l(phi_{j,l}/c_1), 0);
/* -----Iteration----- */
7 for k = 0 : I_max do
8   s^{(k)} = H * (z^{(k)})^T;
9   if s^{(k)} == 0 then return z^{(k)} else if k == I_max
    then return Failure else
10    for i = 0 : (M - 1) do
11      for j in N(i) do
12        sigma_{i,j}^{(k)} = h_{i,j}^{-1} sum_{t in N(i) \setminus {j}} h_{i,t} z_t^{(k)};
13        for l = 0 : (2^r - 1) do
14          if sigma_{i,j}^{(k)} == a_l then R_{j,l}^{(k)} = R_{j,l}^{(k)} + 1
15          R_{j,l}^{(k+1)} = R_{j,l}^{(k)};
15  for j = 0 : (N - 1) do
16    R_j^{(k+1)m} = max_l R_{j,l}^{(k+1)};
17    z_j^{(k+1)} = field element of R_j^{(k+1)m};
18    R_j^{(k+1)m2} = second largest among R_{j,l}^{(k+1)};
19    z_j'^{(k+1)} = field element of R_j^{(k+1)m2};
20    for i in M(j) do
21      if (sigma_{i,j}^{(k)} == z_j^{(k+1)}) & (R_j^{(k+1)m} <= R_j^{(k+1)m2}) + 1
        then z_{i,j}^{(k+1)} = z_j^{(k+1)} else z_{i,j}^{(k+1)} = z_j'^{(k+1)}

```

If the soft-reliability information of the received symbol is available, the EIHRB algorithm achieves a better error performance than the IHRB algorithm. Therefore, we focus on the EIHRB algorithm and do not consider the IHRB algorithm further.

### III. TWO IMPROVEMENTS

#### A. New Reliability Information Update

The reliability information update of line 21 of Alg. 1 can be written as:

$$\begin{aligned} R_{j,l}^{(k+1)} &= R_{j,l}^{(k)} + \psi_{j,l}^{(k)} \\ &= R_{j,l}^{(0)} + \sum_{t=0}^k \psi_{j,l}^{(t)} \end{aligned} \quad (2)$$

For both the ISRB and IHRB algorithms, the reliability information of the  $k$ -th iteration,  $R_{j,l}^{(k)}$ , includes all check-to-variable (c-to-v) messages of previous iterations. This conflicts with the extrinsic information principle. In the EIHRB algorithm, lines 15 to 21 of Alg. 3 are used to recalculate the extrinsic information.

We propose a new reliability information update to exclude the c-to-v messages of previous iterations. In our new reliability information update, only the channel information  $\varphi_{j,l}$  and  $\psi_{j,l}^{(k)}$  of the current iteration are used to compute the reliability information  $R_{j,l}^{(k+1)}$ . Our new reliability information update is

$$R_{j,l}^{(k+1)} = \xi_1 \varphi_{j,l} + \xi_2 \psi_{j,l}^{(k)}, \quad (3)$$

where  $\xi_1$  and  $\xi_2$  are two parameters to improve the error performance.

Eq. (3) is used to replace the reliability information update of line 21 of Alg. 1, and consequently the new algorithm is called the IISRB algorithm.

To reduce complexity of the IISRB algorithm, we change the initialization as follows. For the ISRB algorithm,  $\phi_{i,j}$  and  $\varphi_{j,l}$  are calculated in the initialization. Hence, for the IISRB algorithm, we calculate  $\xi_1 \varphi_{i,j}$  and  $\xi_2 \phi_{i,j}$  in the initialization as well. This helps to reduce the complexity of each iteration. The IISRB algorithm is presented in Alg. 4.

A new reliability information is also applied to the EIHRB algorithm. The reliability information update in line 14 of Alg. 3 is replaced with

$$R_{j,l}^{(k)} = R_{j,l}^{(k)} + c_3, \quad (4)$$

where,  $c_3$  is a parameter to improve the error performance. Meanwhile, at the beginning of each iteration,  $R_{j,l}^{(k)}$  is initialized as  $R_{j,l}^{(0)}$  which is already scaled in line 6 by a parameter  $c_1$ . Furthermore, to be consistent with the IISRB algorithm,  $z_{i,j}^{(k+1)} = z_j^{(k+1)}$ . Finally, lines 15 to 21 of Alg. 3 are not needed any more. The new algorithm derived from the EIHRB algorithm with the four modifications above is referred to as the IEIHRB algorithm.

#### B. Re-selection Scheme

Furthermore, we observe that the error floor of the ISRB algorithm becomes higher, as the column weight of the parity check matrix decreases. The IISRB algorithm suffers the same problem.

In Fig. 1, C1 is an (837, 726) LDPC code over  $\text{GF}(2^5)$  with a column weight of four, C2 an (806, 680) LDPC code over  $\text{GF}(2^5)$  with a column weight of five, C3 a (775, 634)

---

#### Algorithm 4: IISRB algorithm

---

```

1 /* -----Initialization----- */
2 for j = 0 : (N - 1) do
3   z_j^{(0)} = z_j;
4   for l = 0 : (2^r - 1) do
5     phi'_{j,l} = sum_{t=0}^{r-1} (1 - 2a_{l,t}) q_{j,t};
6     phi_{j,l} = xi_1 phi'_{j,l};
7 for i = 0 : (M - 1) do
8   for j in N(i) do
9     phi_{i,j} = xi_2 min_{t in N(i) \setminus {j}} max_l phi'_{t,l};
/* -----Iteration----- */
10 for k = 0 : I_max do
11   s^{(k)} = H * (z^{(k)})^T;
12   if s^{(k)} == 0 then return z^{(k)} else if k == I_max
13     then return Failure else
14     for j = 0 : (N - 1) do
15       for l = 0 : (2^r - 1) do
16         psi_{j,l}^{(k)} = 0;
17         for i in M(j) do
18           sigma_{i,j}^{(k)} = h_{i,j}^{-1} sum_{t in N(i) \setminus {j}} h_{i,t} z_t^{(k)};
19           for l = 0 : (2^r - 1) do
20             if sigma_{i,j}^{(k)} == a_l then
21               psi_{j,l}^{(k)} = psi_{j,l}^{(k)} + phi_{i,j}
22 for j = 0 : (N - 1) do
23   for l = 0 : (2^r - 1) do
24     R_{j,l}^{(k+1)} = phi_{j,l} + psi_{j,l}^{(k)};
25     z_j^{(k+1)} = arg_{a_l} max R_{j,l}^{(k+1)};

```

---

LDPC code over  $\text{GF}(2^5)$  with a column weight of six. All three codes are constructed based on Reed–Solomon codes with two information symbols [20]. The error floor of BLER performance becomes lower as the column weight of the parity check matrix increases. Hence, the column weight of the parity check matrix is one key factor for the error floor.

We propose a re-selection scheme to address this problem.

To simplify the discussion, here we focus on the IISRB algorithm. Our simulation results show that the re-selection scheme also applies to the ISRB, EIHRB and IEIHRB algorithms.

To analyze the error floor, the concept of periodic points is introduced. Given an endomorphism  $f : Z \rightarrow Z$ , a point  $\mathbf{z}$  in  $Z$  is called a *periodic point with a period of  $i$*  if there exists a smallest positive integer  $i$  so that  $f^{(i)}(\mathbf{z}) = \mathbf{z}$ , where  $f^{(i)} = f(f^{(i-1)}(\mathbf{z}))$ .

An iteration of the IISRB algorithm can be considered a function  $f$ . The  $k$ -th iteration of the IISRB algorithm is  $\mathbf{z}^{(k)} = f(\mathbf{z}^{(k-1)}) = f^{(2)}(\mathbf{z}^{(k-2)}) = \dots = f^{(k)}(\mathbf{z}^{(0)})$ , and if  $\mathbf{s}^{(k)} \neq 0$  and  $\mathbf{z}^{(k)} = \mathbf{z}^{(k-i)}$  for  $0 < i \leq k$ , the decoding algorithm results in a periodic point with a period of  $i$ . Our algorithm focuses on only the periodic points with a period of up to

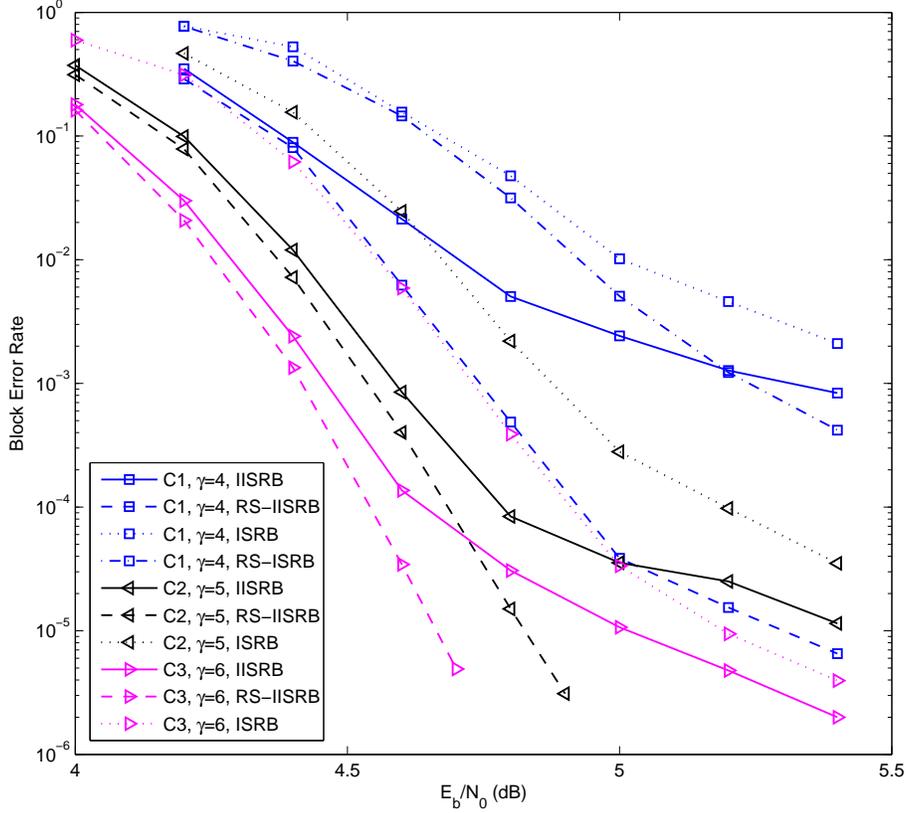


Fig. 1. Block error rates of the soft-reliability based algorithms for different codes with different column weights over the AWGN channel when  $I_{\max} = 50$  and the modulation scheme is BPSK

two for two reasons. First, our simulation results show that the BLERs are caused mainly by periodic points with periods one and two. Second, to identify the existence of a periodic point with a period of greater than two needs more memory to keep track of the hard decisions of the previous iterations.

If the Hamming distance between a periodic point and its corresponding transmitted codeword is less than  $\theta$ , the periodic point is called a small-distance periodic point. Otherwise, it is called a large-distance periodic point. Fig. 2 compares the BLERs of the IISRB algorithm with those caused by the large-distance and small-distance periodic points for the (837, 726) code when  $\theta = 8$ . For low SNRs, the overall BLER is dominated by those caused by large-distance periodic points. The sum of the BLERs due to the large-distance and the small-distance periodic points is less than the overall BLER, because periodic points with a period greater than two also cause some BLERs. For high SNRs, the BLER caused by the small-distance periodic points dominates the total BLER. A similar trend for the ISRB algorithm was observed as well. Hence, for the ISRB and IISRB algorithms, the error floor is mainly caused by the small-distance periodic points. In order to lower the error floor of IISRB algorithm, the BLER caused by the small-distance periodic points should be reduced.

Consider the hard decision process of line 22 of Alg. 1. If the most likely decision is wrong, the second most likely

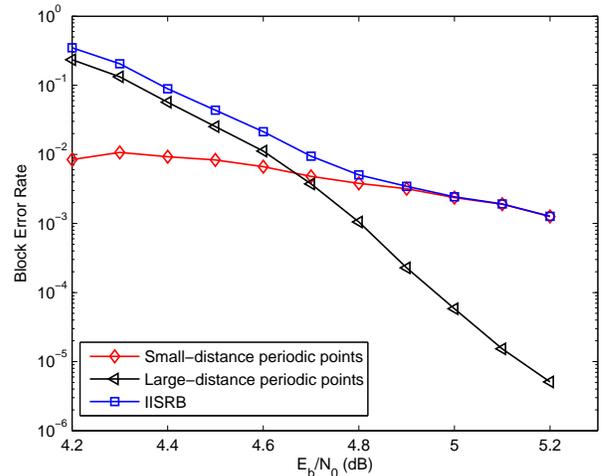


Fig. 2. BLERs of the small-distance and large-distance periodic points for the IISRB algorithm to decode the (837, 726) code over the AWGN channel when the modulation scheme is BPSK

decision is supposed to be the best choice to be decoded. The smaller the difference between the maximal reliability information and the second maximal reliability information, the greater the probability that the most likely decision is

wrong.

Based on this intuition, when a periodic point is detected, the re-selection scheme tries to help the decoder get away from the periodic point by using the second most likely decision. The re-selection scheme consists of two steps. The first step is to identify the existence of a periodic point when the syndrome vector is a non-zero vector. The second step is to identify positions of erroneous symbols. A set is defined to include variable nodes adjacent to unsatisfied check nodes. This set contains some erroneous symbols. Then, among the variable nodes belonging to the set, the position of a variable node which has the smallest difference between its maximal reliability information and second maximal reliability information can be identified. If there are multiple variable nodes having the smallest difference, the first one is selected. Assume the index of this position is  $rs\_n$ . Let  $us\_c_j$  represent the number of unsatisfied check nodes connected with the  $j$ -th variable node for  $0 \leq j < N$ . The most likely decision  $z_{rs\_n}^{(k)}$  is replaced by the second most likely decision  $\tilde{z}_{rs\_n}^{(k)}$ . Meanwhile,  $\varphi_{rs\_n, z_{rs\_n}^{(k)}}$  is reduced by a preset offset  $\zeta$  and  $\varphi_{rs\_n, \tilde{z}_{rs\_n}^{(k)}}$  is added by the same preset offset. The detailed re-selection scheme is described in Alg. 5. Here,  $s_i^{(k)}$  is the  $i$ -th value of the syndrome vector  $\mathbf{s}^{(k)}$ .

---

**Algorithm 5:** Re-selection scheme

---

```

1 for  $j = 0 : (N - 1)$  do
2    $\tilde{z}_j^{(k)} = \arg_{a_l \in GF(2^r) \setminus \{z_j^{(k)}\}} \max R_{j,l}^{(k)}$ ;
3 if  $(\mathbf{z}^{(k-1)} == \mathbf{z}^{(k)})$  or  $(\mathbf{z}^{(k-2)} == \mathbf{z}^{(k)})$  then
4   dif_R =  $R_{0, z_0^{(k)}}^{(k)}$ ;
5 for  $j = 0 : (N - 1)$  do
6   us_c_j = 0;
7   for  $i \in \mathcal{M}(j)$  do
8     if  $(s_i^{(k)} > 0)$  then us_c_j ++
9     if  $(us\_c_j > 0)$  and  $((R_{j, z_j^{(k)}}^{(k)} - R_{j, \tilde{z}_j^{(k)}}^{(k)}) < dif\_R)$ 
10      then
11       dif_R =  $(R_{j, z_j^{(k)}}^{(k)} - R_{j, \tilde{z}_j^{(k)}}^{(k)})$ ;
12       rs_n = j;
13  $\varphi_{rs\_n, z_{rs\_n}^{(k)}} = \varphi_{rs\_n, z_{rs\_n}^{(k)}} - \zeta$ ;
14  $\varphi_{rs\_n, \tilde{z}_{rs\_n}^{(k)}} = \varphi_{rs\_n, \tilde{z}_{rs\_n}^{(k)}} + \zeta$ ;
15  $z_{rs\_n}^{(k)} = \tilde{z}_{rs\_n}^{(k)}$ ;
16 for  $i \in \mathcal{M}(rs\_n)$  do
17    $s_i^{(k)} = \mathbf{h}_i \cdot (\mathbf{z}^{(k)})^T$ ;

```

---

This scheme can be applied to any majority decoding algorithms. For the ISRB algorithm, this scheme is added between lines 11 and 12. Similarly, the re-selection scheme can be inserted at the corresponding position of other algorithms. “RS-” is prefixed in front of the name of the algorithms to show that an algorithm adopts the re-selection scheme. For instance, the ISRB algorithm with the re-selection scheme is called as the RS-ISRB algorithm.

Fig. 3 shows the BLERs of the RS-IISRB algorithm and those caused by the low-distance and high-distance periodic points. Compared with the IISRB algorithm, the BLER caused by the high-distance periodic points descends to  $2 \times 10^{-4}$  from  $1.2 \times 10^{-3}$ , and the BLER caused by the low-distance periodic points is reduced to  $7 \times 10^{-5}$  from  $4 \times 10^{-3}$  when SNR is 4.8 dB. Hence, the rs-selection scheme reduces the occurrences of both the low-distance and high-distance periodic points and works better on the low-distance periodic points. Even for the RS-IISRB algorithm, the low-distance periodic points still are the primary reason for the error floor.

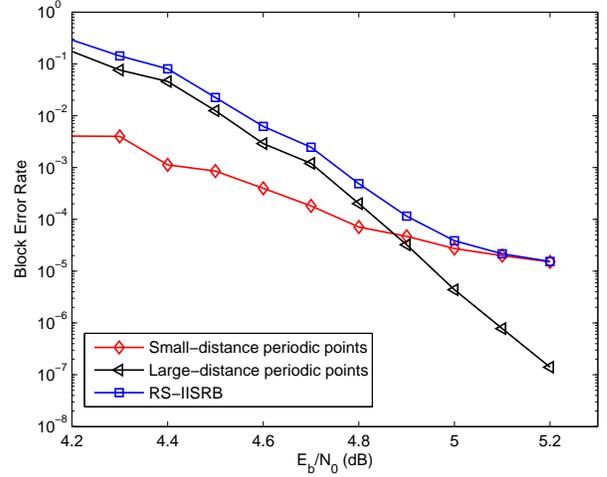


Fig. 3. BLERs of the small-distance and large-distance periodic points for the RS-IISRB algorithm to decode the (837, 726) code over the AWGN channel when the modulation scheme is BPSK

The re-selection scheme helps the decoding algorithm correct some periodic points. It is likely that the decoding algorithm goes out of a periodic point temporarily, and goes back to the same periodic point or results in another periodic point. Therefore, even with the re-selection scheme, the decoding algorithm still encounters the error floor problem. Moreover, the re-selection scheme works better on the small-distance periodic points because in general a small-distance periodic point involves fewer unsatisfied check nodes than a large-distance periodic point.

## IV. PERFORMANCE EVALUATION

### A. Error Performance and Average Numbers of Iterations

The BPSK modulation scheme, the additive white Gaussian noise (AWGN) channel with a single-sided power spectral density  $N_0$ , and a 6-bit uniform quantization with 64 levels which has an interval length  $\Delta = 0.0625$  are used in our numerical simulations. The maximum number of iterations is 50, i.e.,  $I_{\max} = 50$ . Our simulations focus on C1, C2, and C3, whose parity check matrices have small column weights, as well as a (255, 175) cyclic LDPC code over  $GF(2^8)$  constructed with the method as describe in [19, Example 4], because it has a large column weight of 16.

We first compare the performance of the soft-reliability based algorithms. The ISRB, IISRB and RS-IISRB algorithms

are used to decode the (255, 175) code. For the ISRB algorithm, different values of  $\lambda = 4l$  for  $l = 1, 2, \dots, 8$ , were tried, and  $\lambda = 16$  leads to the best performance. For the new reliability information update, different combinations of  $\xi_1$  and  $\xi_2$  were tested. Since they are weighting factors, we fix  $\xi_2 = 1$  and try different values for  $\xi_1$ . As shown in Fig. 4, for the (255, 175) code,  $(\xi_1 = 7, \xi_2 = 1)$  results in the best error performance. The real values from 6.2 to 7 with a step size of 0.2 for  $\xi_1$  and  $\xi_2 = 1$  were tested, shown in Fig. 5. Performance differences between different real value coefficients are very small. Henceforth, integer values are used for  $\xi_1$  and  $\xi_2$  to reduce complexity.

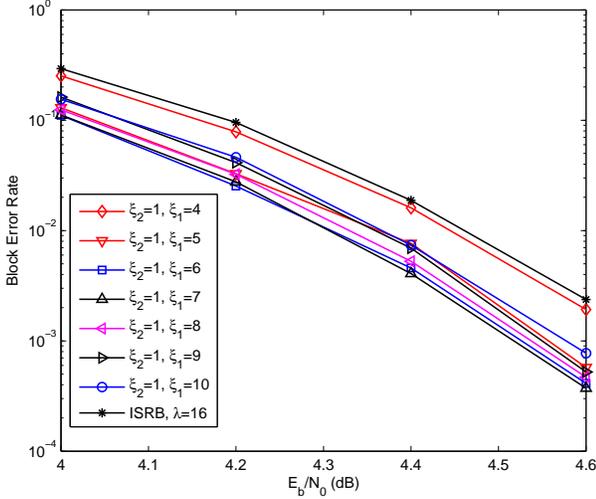


Fig. 4. The impact of different integer values for  $\xi_1$  on the BLER of the IISRB algorithm for the (255, 175) code when  $I_{\max} = 50$  and the modulation scheme is BPSK over the AWGN channel

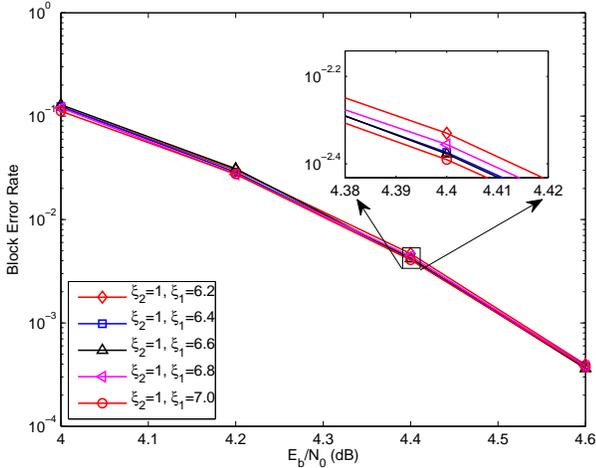


Fig. 5. The impact of different real values for  $\xi_1$  on the BLER of the IISRB algorithm to decode the (255, 175) code when  $I_{\max} = 50$  and the modulation scheme is BPSK over the AWGN channel

The BLER curves of the ISRB, IISRB, RS-IISRB and Min-Max algorithms for the (255,175) code are shown in Fig. 6. The IISRB algorithm has a 0.15 dB coding gain versus

the ISRB algorithm in this case. The RS-IISRB algorithm also achieves a slight improvement compared to the IISRB algorithm and has a performance loss of about 0.4 dB versus the Min-Max algorithm.

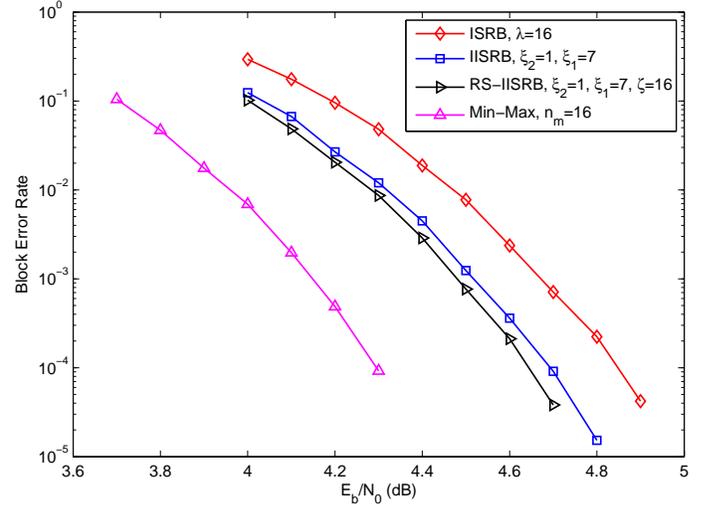


Fig. 6. Block error rates of the ISRB, IISRB and RS-IISRB algorithms for the (255, 175) code when  $I_{\max} = 50$  and the modulation scheme is BPSK over the AWGN channel

If a total of  $T$  iterations is used to decode  $K$  received words, the average number of iterations per received word is  $T/K$ . The average numbers of iterations per received word for the soft-reliability algorithms are compared in Table I, where  $K$  is chosen such that at least 100 erroneous decoded words are observed for each SNR. Table I shows that both the RS-IISRB and IISRB algorithms require fewer iterations than the ISRB algorithm. At 4.7 dB, the average number of iterations of the IISRB algorithm is fewer by 10% than that of the ISRB algorithm. The advantage of the IISRB and RS-IISRB algorithm is even more pronounced for low SNRs.

TABLE I  
AVERAGE NUMBER OF ITERATIONS OF THE MIN-MAX ( $n_m = 16$ ), ISRB, IISRB AND RS-IISRB ALGORITHMS FOR THE (255, 175) CODE WHEN  $I_{\max} = 50$  AND THE MODULATION SCHEME IS BPSK OVER THE AWGN CHANNEL

$E_b/N_0$ (dB)	Min-Max [26]	ISRB [27]	IISRB	RS-IISRB
4.0	2.35	18.76	11.58	11.25
4.1	1.91	13.25	8.17	7.85
4.2	1.60	9.10	5.78	5.71
4.3	1.36	6.46	4.46	4.41
4.4	N/A	4.59	3.59	3.59
4.5	N/A	3.68	3.06	3.05
4.6	N/A	3.10	2.71	2.70
4.7	N/A	2.74	2.46	2.46
4.8	N/A	2.50	2.27	2.28

The ISRB, IISRB and RS-IISRB algorithms are also used to decode the (837, 726) code. The best BLER performance of the IISRB algorithm is achieved when  $\xi_1 = 4$  and  $\xi_2 = 1$ . Fig. 7 compares the BLERs of the ISRB, IISRB, RS-IISRB and Min-Max algorithms for this code. The IISRB algorithm has a 0.2 dB coding gain versus the ISRB algorithm, but both algorithms show an error floor around  $10^{-3}$ . Compared with

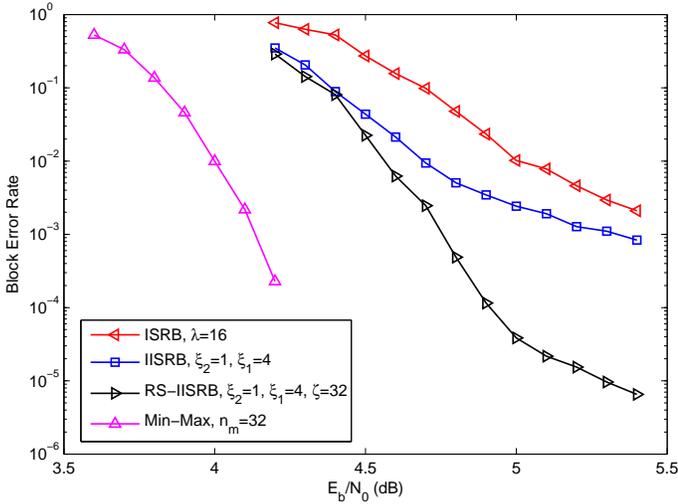


Fig. 7. Block error rates of the ISRB, IISRB and RS-IISRB algorithms for the (837, 726) code when  $I_{\max} = 50$  and the modulation scheme is BPSK over the AWGN channel

these two algorithms, for low SNRs the RS-IISRB algorithm shows a slight improvement, and for high SNRs the RS-IISRB algorithm lowers the error floor to below  $10^{-5}$  and has a performance loss of only 0.6 dB versus the Min-Max algorithm at the BLER of  $10^{-3}$ .

The average numbers of iterations for the (837,726) code with different SNRs are listed in Table II. The average numbers of iterations required by the IISRB and RS-IISRB algorithms are reduced by at least 20% when the  $I_{\max} = 50$ . Iterations required by the RS-IISRB algorithm is slightly fewer than that of the IISRB algorithm because of the re-selection scheme. In addition, we compare the running time of different decoding algorithms (implemented in C) on a DELL Optiplex 755. To decode 10,000 codewords of the (837,726) code over the AWGN channel at the SNR of 5.4 dB, the ISRB, IISRB and RS-IISRB algorithms run 22.22, 19.48 and 19.37 seconds, respectively. In terms of the running time, RS-IISRB < IISRB < ISRB, which is consistent with the comparison based on the average number of iterations.

TABLE II  
AVERAGE NUMBER OF ITERATIONS OF THE ISRB, IISRB AND RS-IISRB ALGORITHMS FOR THE (837, 726) CODE WHEN  $I_{\max} = 50$  AND THE MODULATION SCHEME IS BPSK OVER THE AWGN CHANNEL

$E_b/N_0$ (dB)	ISRB [27]	IISRB	RS-IISRB
4.5	22.18	10.58	9.97
4.6	16.52	8.22	7.62
4.7	12.69	6.54	6.30
4.8	9.40	5.49	5.33
4.9	7.44	4.79	4.64
5.0	5.99	4.20	4.12
5.1	5.21	3.80	3.71
5.2	4.53	3.44	3.38
5.3	4.06	3.15	3.10
5.4	3.66	2.90	2.87

For C1, C2 and C3, the BLERs of the RS-IISRB algorithm are shown with the dashed curves in Fig. 1. For C1 and C2,  $\xi_1 = 4$ ,  $\xi_2 = 1$ ,  $\lambda = 16$ ,  $\zeta = 32$ . For C3,  $\xi_1 = 5$ ,  $\xi_2 = 1$ ,  $\lambda = 16$ ,  $\zeta = 32$ . The RS-IISRB algorithm improves the BLER

performance and lowers the error floor for all three codes. In Fig. 1, for C1, the simulation result for the RS-ISRB algorithm is shown as well, which does not adopt the new reliability information update but the re-selection scheme. It appears that the re-selection scheme also provides some performance gain. If both improvements are applied, the RS-IISRB algorithm achieves a greater performance gain.

Next, we compare the performances of hard-reliability based MLGD algorithms. The EIHRB-INIT algorithm [30] is a simplified version of the EIHRB algorithm without the recalculation of the extrinsic information. The RS-IEIHRB algorithm is developed by integrating the re-selection scheme describe in Section III-B into the IEIHRB algorithm.

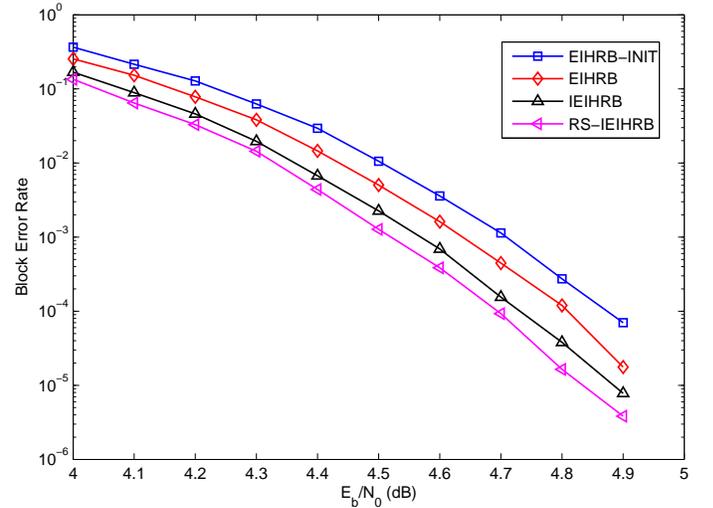


Fig. 8. Block error rates of hard-reliability based algorithms for the (255,175) code when  $I_{\max} = 50$  and the modulation scheme is BPSK over the AWGN channel

Fig. 8 shows the BLERs of different hard-reliability based algorithms for the (255,175) code, and Table III lists the average numbers of iterations when  $I_{\max} = 50$ . For the EIHRB-INIT and EIHRB algorithm,  $c_1 = 4$  and  $c_2 = 15$ . For the IEIHRB and RS-IEIHRB algorithm,  $c_1 = 1$ ,  $c_2 = 63$ ,  $c_3 = 12$  and  $\zeta = 16$ . For the (255,175) code the new reliability information update provides about 0.05 dB performance gain, and the re-selection scheme provides another 0.05 dB performance gain. Hence, compared with the EIHRB algorithm, the RS-IEIHRB algorithm has about 0.1 dB performance gain, and the average number of iterations required by the RS-IEIHRB algorithm is reduced by about 30%.

Fig. 9 compares the BLERs of hard-reliability based algorithms for different non-binary LDPC codes with different column weights. For C1 and C2,  $c_1 = 10$ ,  $c_2 = 63$ ,  $c_3 = 2$  and  $\zeta = 32$ . For C3,  $c_1 = 11$ ,  $c_2 = 63$ ,  $c_3 = 2$  and  $\zeta = 32$ . For the (837,726) code, the EIHRB algorithm also has an error floor of  $10^{-3}$ . For low SNRs, the IEIHRB algorithm outperforms the EIHRB algorithm and the RS-IEIHRB algorithm reduces the error floor to a level of  $10^{-5}$ . In the error floor region, the EIHRB algorithm is better than the IEIHRB algorithm because of the use of  $z_{i,t}^{(k)}$  and recalculating the extrinsic information in the latter. The two improvements in Section III also help to reduce the average number of iterations by about

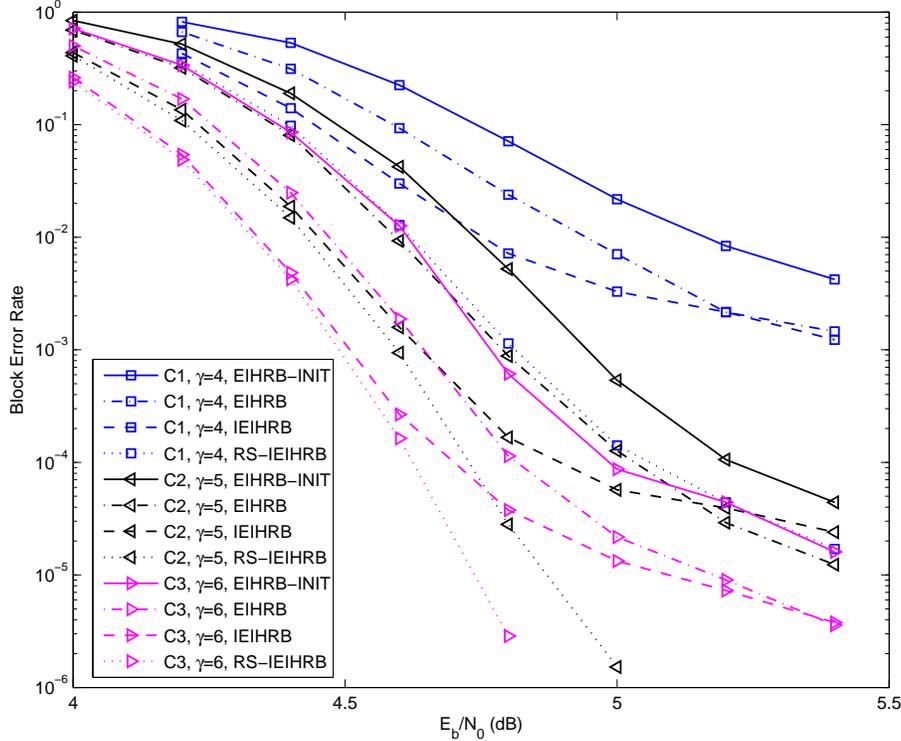


Fig. 9. Block error rates of hard-reliability based algorithms for different non-binary LDPC codes with different column weights when  $I_{\max} = 50$  and the modulation scheme is BPSK over the AWGN channel

TABLE III

AVERAGE NUMBER OF ITERATIONS OF THE HARD-RELIABILITY BASED ALGORITHMS FOR THE (255, 175) CODE WHEN  $I_{\max} = 50$  AND THE MODULATION SCHEME IS BPSK OVER THE AWGN CHANNEL

$E_b/N_0$ (dB)	EIHRB-INIT [30]	EIHRB [30]	IEIHRB	RS-IEIHRB
4.0	22.28	18.56	12.67	12.09
4.1	16.48	13.62	8.99	8.85
4.2	12.21	9.54	6.42	6.09
4.3	8.57	7.20	4.70	4.62
4.4	6.44	5.57	3.70	3.69
4.5	5.15	4.70	3.10	3.10
4.6	4.29	4.07	2.73	2.72
4.7	3.75	3.65	2.47	2.47
4.8	3.38	3.33	2.28	2.28
4.9	3.09	3.06	2.14	2.14

TABLE IV

AVERAGE NUMBER OF ITERATIONS OF THE HARD-RELIABILITY BASED ALGORITHM FOR THE (837, 726) CODE WHEN  $I_{\max} = 50$  AND THE MODULATION SCHEME IS BPSK OVER THE AWGN CHANNEL

$E_b/N_0$ (dB)	EIHRB-INIT [30]	EIHRB [30]	IEIHRB	RS-IEIHRB
4.2	43.01	37.25	28.83	27.49
4.4	32.19	23.09	15.52	14.55
4.6	18.85	12.41	8.29	7.83
4.8	10.57	7.60	5.45	5.22
5.0	6.79	5.49	4.13	4.01
5.2	4.98	4.37	3.36	3.27

20% for the (837,726) code as listed in Table IV. For C2 and C3, the new reliability information update provides some performance gains for low SNRs, and the error floors are lowered effectively.

We evaluate the proposed decoding algorithms over block fading channels, which are widely used in wireless communication systems involving slow time-frequency hopping or multi-carrier modulation using orthogonal frequency division multiplexing technique. We assume that each codeword experiences a block Rayleigh fading channel and that the receiver has perfect channel state information. Fig. 10 and Fig. 11 show the BLERs and the average numbers of iterations of

different MLGD algorithms for the (837,726) code over a block Rayleigh fading channel. In Fig. 10, the IISRB and RS-IISRB algorithms have a gain of about 0.2 dB over the ISRB algorithm, which is similar to that over the AWGN channel shown in Fig. 7. At a SNR of 23 dB, the proposed improvements reduce the average number of iterations by about 5%.

In a word, the two improvements introduced in Section III apply to both the soft-reliability and hard-reliability based MLGD algorithms. While both improvements improve the error performance and require fewer iterations on average, the re-selection scheme lowers the error floor of codes having low column weights effectively.

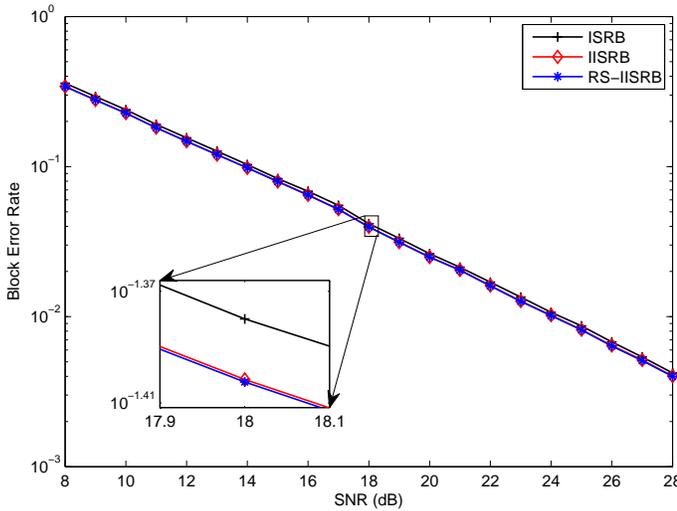


Fig. 10. Block error rates of algorithms for the (837, 726) code when  $I_{\max} = 50$  and the modulation scheme is BPSK over the block Rayleigh fading channel

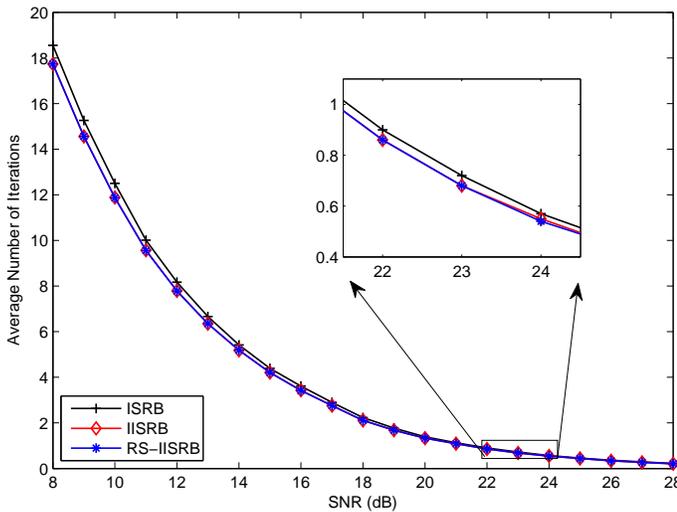


Fig. 11. Average numbers of iterations for different algorithms for the (837, 726) code when  $I_{\max} = 50$  and the modulation scheme is BPSK over the block Rayleigh fading channel

### B. Computational Complexity Reduction

We evaluate impacts on the complexity by the two proposed improvements and focus on the soft-reliability based MLGD algorithms first. Assume the quantized input information  $q_{j,t}$  has a bit width of  $\omega$ . Without the clipping operation of Eq. (1), for the ISRB algorithm,  $R_{j,l}^{(k)}$  needs  $\omega + \lceil \log_2((\lambda + I_{\max}\gamma)r) \rceil$  bits and its bit width increases as  $I_{\max}$  grows. However, for the IISRB algorithm,  $R_{j,l}^{(k)}$  needs only  $\omega + \lceil \log_2((\xi_1 + \xi_2\gamma)r) \rceil$  bits and  $I_{\max}$  has no impact on  $R_{j,l}^{(k)}$ 's bit width. With the clipping operation,  $R_{j,l}^{(k)}$  needs a smaller bit width in the ISRB algorithm. However,  $N2^r$  IAs and  $N(2^r - 1)$  ICs are needed per iteration to carry out the clipping operation. In contrast, there is no accumulation operation in the IISRB and RS-IISRB algorithms. Thus, saturation is not an issue for the IISRB and RS-IISRB algorithms, and the clipping operation is not needed.

For the IISRB algorithm, let us consider the initialization step first. There are  $N2^r$   $\varphi_{j,l}$ 's. To compute  $\varphi_{j,l}$ 's needs  $Nr2^r$  IAs. Because  $\max_l \varphi_{t,l} = \varphi_{t,z_t}$  and there are  $N\gamma$   $\phi_{i,j}$ 's,  $N\gamma(\rho - 2)$  ICs are needed to calculate  $\phi_{i,j}$ 's. The calculations of  $\xi_1\varphi_{j,l}$ 's and  $\xi_2\phi_{i,j}$ 's need  $N2^r$  and  $N\gamma\rho$  IMs, respectively. Therefore, the initialization step needs  $Nr2^r$  IAs,  $N2^r + N\gamma\rho$  IMs and  $N\gamma(\rho - 2)$  ICs.

We now analyze the complexity per iteration of the IISRB algorithm. Each iteration needs  $M\rho$  FMs and  $M(\rho - 1)$  FAs to calculate the syndrome  $\mathbf{s}^{(k)}$ . Line 16 in Alg. 1 can be reformulated as:

$$\sigma_{i,j}^{(k)} = h_{i,j}^{-1} s_i^{(k)} + z_j^{(k)} \quad (5)$$

Hence,  $N\gamma$  FAs and  $N\gamma$  FMs are needed to calculate  $\sigma_{i,j}^{(k)}$ 's. Assume there are  $u_j^{(k)}$  ( $0 < u_j^{(k)} \leq \gamma$ ) different values among  $\sigma_{i,j}^{(k)}$ 's for each  $j$ , then  $u_j^{(k)}$   $R_{j,l}^{(k+1)}$ 's need to be updated. To compute  $\psi_{j,l}^{(k)}$ 's and  $R_{j,l}^{(k+1)}$ 's,  $\gamma - u_j^{(k)}$  and  $u_j^{(k)}$  IAs are needed, respectively, for each  $j$ . For  $z_j^{(k+1)}$ ,  $R_{j,z_j}^{(k+1)}$  must be one of  $R_{j,z_j}^{(k+1)}$  and those  $R_{j,l}^{(k+1)}$  updated in the  $k$ -th iteration. To make the hard decisions,  $N\gamma$  ICs are needed at most. Hence, in the worst case, each iteration of the IISRB algorithm requires  $2N\gamma$  FMs,  $2N\gamma - M$  FAs,  $N\gamma$  IAs and  $N\gamma$  ICs ( $M\rho = N\gamma$ ). Compared with the ISRB algorithm, the IISRB algorithm saves  $N2^r$  IAs and  $N(2^{r+1} - 2 - \gamma)$  ICs for each iteration, while requiring the same numbers of FAs and FMs. This saving is significant if  $2^r$  is large.

Let us calculate computational complexity overhead due to the re-selection scheme.  $\tilde{\mathbf{z}} = (\tilde{z}_0, \tilde{z}_1, \dots, \tilde{z}_{N-1})$  represents the second most likely decision of the received word  $\mathbf{y}$ . To acquire  $\tilde{z}_j$  in the initialization step,  $r - 1$  ICs are needed for each  $j$ , because  $r$ -bit representations of  $\tilde{z}_j$  and  $z_j$  differ by one bit and there are  $r$  elements over  $\text{GF}(2^r)$  satisfying this constraint. Hence, the initialization step of the RS-IISRB algorithm needs  $N(r - 1)$  ICs more than that of the IISRB algorithm. For each iteration, the second maximum among  $R_{j,l}^{(k)}$ 's must be one of  $R_{j,\tilde{z}_j}^{(k)}$ ,  $R_{j,z_j}^{(k)}$  and those  $R_{j,l}^{(k)}$ 's updated. It needs at most  $N(\gamma + 1)$  ICs per iteration. Line 3 of Alg. 5 needs  $2N$  ICs to identify the existence of a periodic point.  $N\gamma$  IAs and  $N\gamma$  ICs are needed to calculate  $us\_c_j$ . The calculation of  $\text{dif\_R}$  needs  $2N$  ICs and  $N$  IAs.  $\varphi_{rs\_n, z_{rs\_n}^{(k)}}$  and  $\varphi_{rs\_n, \tilde{z}_{rs\_n}^{(k)}}$  need two IAs. After the re-selection scheme, there are  $\gamma$  syndromes to be recalculated so that Eq. (5) can be applied, requiring  $2\gamma$  FAs and  $\gamma$  FMs. Therefore, the RS-IISRB algorithm needs  $2\gamma$  FAs,  $\gamma$  FMs,  $N\gamma + N + 2$  IAs and  $5N + 2N\gamma$  ICs per iteration more than the IISRB algorithm.

Complexities of the hard-reliability based algorithms can be analyzed similarly. The IEIHRB algorithm has the same computational complexity per iteration as the IISRB algorithm, because they have the same iteration procedure. For the same reason, the RS-IEIHRB algorithm has the same computational complexity per iteration as the RS-IISRB algorithm.

Tables V and VI compare computational complexities of various decoding algorithms. For the initialization step, the numbers of IMs of the IISRB and RS-IISRB algorithms are greater than that of the ISRB algorithm because the calculation of  $\xi_2\phi_{i,j}$  is done in initialization to reduce computational

TABLE V

COMPUTATIONAL COMPLEXITIES OF THE INITIALIZATION STEP FOR VARIOUS DECODING ALGORITHM TO DECODE AN LDPC CODE OVER  $\text{GF}(2^r)$  WITH AN  $M \times N$  PARITY CHECK MATRIX WHOSE COLUMN AND ROW WEIGHTS ARE  $\gamma$  AND  $\rho$

Algorithms	IA	IM	IC	ID	Floor
ISRB [27]	$Nr2^r$	$N2^r$	$MN(2^r - 1)(3\rho - 6)$	0	0
IISRB	$Nr2^r$	$N2^r + N\gamma\rho$	$N\gamma(\rho - 2)$	0	0
RS-IISRB	$Nr2^r$	$N2^r + N\gamma\rho$	$N\gamma(\rho - 2) + N(r - 1)$	0	0
EIHRB [30]	$N2^r(r + 2)$	0	$N2^r$	$N2^r$	$N2^r$
IEIHRB	$N2^r(r + 2)$	0	$N2^r$	$N2^r$	$N2^r$
RS-IEIHRB	$N2^r(r + 2)$	0	$N2^r + N(r - 1)$	$N2^r$	$N2^r$

TABLE VI

COMPUTATIONAL COMPLEXITIES PER ITERATION FOR VARIOUS DECODING ALGORITHM TO DECODE AN LDPC CODE OVER  $\text{GF}(2^r)$  WITH AN  $M \times N$  PARITY CHECK MATRIX WHOSE COLUMN AND ROW WEIGHTS ARE  $\gamma$  AND  $\rho$

Algorithms	FA	FM	IA	IC
ISRB [27]	$2N\gamma - M$	$2N\gamma$	$N\gamma + N2^r$	$2N2^r - 2N$
IISRB	$2N\gamma - M$	$2N\gamma$	$N\gamma$	$N\gamma$
RS-IISRB	$2N\gamma - M + 2\gamma$	$2N\gamma + \gamma$	$2N\gamma + N + 2$	$5N + 3N\gamma$
EIHRB [30]	$3N\gamma - 2M$	$3N\gamma$	$2N\gamma + N2^r$	$2N2^r - 2N + N\gamma$
IEIHRB	$2N\gamma - M$	$2N\gamma$	$N\gamma$	$N\gamma$
RS-IEIHRB	$2N\gamma - M + 2\gamma$	$2N\gamma + \gamma$	$2N\gamma + N + 2$	$5N + 3N\gamma$

complexities of iterations. This is a good trade-off for computational complexity. The number of ICs needed by the initialization step of the ISRB algorithm provided in [27, Section III-A] is significantly greater than those of the other algorithms. This is because in [27],  $\phi_{i,j}$ 's are calculated for every  $i$  and  $j$ , and  $\max_l \varphi_{t,l}$ 's are re-calculated for each  $\phi_{i,j}$ . For each iteration, the numbers of integer operations required by the ISRB and EIHRB algorithms scale with  $2^r$ , the order of the finite field. With the new reliability information update, the numbers of integer operations are reduced greatly and are now independent of  $2^r$ . The re-selection scheme incurs some additional complexity, but complexities of the RS-IISRB and RS-IEIHRB algorithms are still lower than those of the ISRB and EIHRB algorithms, respectively.

Tables VII and VIII list the numbers of various operations for initialization and each iteration, respectively, needed by various decoding algorithms for the (255,175) code. For initialization, the ISRB algorithm needs significantly more ICs than the other algorithms. When the order of the finite field is higher, our improved algorithms reduce the numbers of IAs and ICs for each iteration significantly.

From a perspective of the computational complexity, the IISRB and IEIHRB algorithms are the best. The re-selection scheme needs more finite field operations and integer operations. All the improved algorithms are simpler than the ISRB and EIHRB algorithms.

Let us consider the memory overhead required by the two improvements. Our first improvement—the new reliability information update—does not need any extra memory units. The second improvement—the re-selection scheme—needs to store  $\mathbf{z}^{(k-1)}$  and  $\mathbf{z}^{(k-2)}$  and hence requires  $2Nr$  extra memory bits. Hence, the re-selection scheme increases the memory requirement slightly, but it does lower the error floor.

## V. CONCLUSION

In this paper, we propose two improvements to the soft-reliability and hard-reliability based MLGD algorithms for

TABLE VII

COMPUTATIONAL COMPLEXITIES OF THE INITIALIZATION STEP FOR VARIOUS DECODING ALGORITHMS TO DECODE THE (255,175) CODE

Algorithm	IA	IM	IC	ID	Floor
ISRB [27]	522240	65280	696417750	0	0
IISRB	522240	130560	57120	0	0
RS-IISRB	522240	130560	58905	0	0
EIHRB [30]	522750	0	65280	65280	65280
IEIHRB	522750	0	65280	65280	65280
RS-IEIHRB	522750	0	67065	65280	65280

TABLE VIII

COMPUTATIONAL COMPLEXITIES REQUIRED PER ITERATION FOR VARIOUS DECODING ALGORITHM TO DECODE THE (255,175) CODE

Algorithm	FA	FM	IA	IC
ISRB [27]	7905	8160	69360	130050
IISRB	7905	8160	4080	4080
RS-IISRB	7937	8176	8417	13515
EIHRB [30]	11730	12240	73440	134130
IEIHRB	7905	8160	4080	4080
RS-IEIHRB	7937	8176	8417	13515

non-binary LDPC codes. The first improvement—the new reliability information update—helps the reliability-based MLGD algorithms achieve better BLERs, require fewer iterations, and have lower complexities. The second improvement—the re-selection scheme—results in a better error performance, fewer iterations on average, and a lower error floor. Although the re-selection scheme needs additional complexity, the MLGD algorithms with the re-selection scheme still require lower computational complexities than the existing MLGD algorithms.

## REFERENCES

- [1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
- [2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

- [3] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
- [4] *Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)*, ETSI Std., Rev. 1.2.1, Aug. 2009.
- [5] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Std. 802.3an-2006, 2006. [Online].
- [6] *Unified high-speed wireline-based home networking transceivers system architecture and physical layer specification*, ITU-T Std. G.9960, Dec. 2011.
- [7] *IEEE Standard for Information technology-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, IEEE Std. IEEE Std 802.11n-2009, 2009. [Online].
- [8] A. Cohen and K. Parhi, "A low-complexity hybrid LDPC code encoder for IEEE 802.3an (10GBase-T) ethernet," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 4085–4094, Oct. 2009.
- [9] T. Zhang and K. Parhi, "Joint (3,k)-regular LDPC code and decoder/encoder design," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 1065–1079, Apr. 2004.
- [10] A. Naderi, S. Mannor, M. Sawan, and W. Gross, "Delayed stochastic decoding of LDPC codes," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5617–5626, Nov. 2011.
- [11] S. Sharifi Tehrani, A. Naderi, G.-A. Kamendje, S. Hemati, S. Mannor, and W. Gross, "Majority-based tracking forecast memories for stochastic LDPC decoding," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4883–4896, Sep. 2010.
- [12] S. Sharifi Tehrani, S. Mannor, and W. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5692–5703, Nov. 2008.
- [13] C.-Z. Zhan, X.-Y. Shih, and A.-Y. Wu, "High-performance scheduling algorithm for partially parallel LDPC decoder," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3177–3180.
- [14] Q. Huang, J. Kang, L. Zhang, S. Lin, and K. Abdel-Ghaffar, "Two reliability-based iterative majority-logic decoding algorithms for LDPC codes," *IEEE Transactions on Communications*, vol. 57, no. 12, pp. 3597–3606, Dec. 2009.
- [15] J. Kang, Q. Huang, S. Lin, and K. Abdel-Ghaffar, "An iterative decoding algorithm with backtracking to lower the error-floors of LDPC codes," *IEEE Transactions on Communications*, vol. 59, no. 1, pp. 64–73, Jan. 2011.
- [16] M. C. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *IEEE Communications Letters*, vol. 2, no. 6, pp. 165–167, Jun. 1998.
- [17] H. Song and J. R. Cruz, "Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording," *IEEE Transactions on Magnetics*, vol. 39, no. 2, pp. 1081–1087, Mar. 2003.
- [18] X. Jiang and M. H. Lee, "Large girth non-binary LDPC codes based on finite fields and euclidean geometries," *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 521–524, Jun. 2009.
- [19] L. Zeng, L. Lan, Y. Y. Tai, B. Zhou, S. Lin, and K. A. S. Abdel-Ghaffar, "Construction of nonbinary cyclic, quasi-cyclic and regular LDPC codes: A finite geometry approach," *IEEE Transactions on Communications*, vol. 56, no. 3, pp. 378–387, Mar. 2008.
- [20] B. Zhou, J. Kang, S. Song, S. Lin, K. Abdel-Ghaffar, and M. Xu, "Construction of non-binary quasi-cyclic LDPC codes by arrays and array dispersions," *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1652–1662, Jun. 2009.
- [21] W. Tang, J. Huang, L. Wang, and S. Zhou, "Nonbinary LDPC decoding by min-sum with adaptive message control," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2011, pp. 3164–3167.
- [22] D. J. MacKay and M. C. Davey, "Evaluation of Gallager codes of short block length and high rate application," in *Proceedings of IMA workshop on Codes, Systems and Graphical Models*, 2000, pp. 113–130.
- [23] L. Barnault and D. Declercq, "Fast decoding algorithm for LDPC over GF(2<sup>q</sup>)," in *Proceedings of IEEE Information Theory Workshop*, 2003, pp. 70–73.
- [24] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-domain decoding of LDPC codes over GF(q)," in *Proceedings of IEEE International Conference on Communications*, vol. 2, Jun. 2004, pp. 772–776.
- [25] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, Apr. 2007.
- [26] V. Savin, "Min-max decoding for non binary LDPC codes," in *Proceedings of IEEE International Symposium on Information Theory*, Jul. 2008, pp. 960–964.
- [27] C.-Y. Chen, Q. Huang, C.-C. Chao, and S. Lin, "Two low-complexity reliability-based message-passing algorithms for decoding non-binary LDPC codes," *IEEE Transactions on Communications*, vol. 58, no. 11, pp. 3140–3147, Nov. 2010.
- [28] K. Kasai and K. Sakaniwa, "Fourier domain decoding algorithm of non-binary LDPC codes for parallel implementation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2011, pp. 3128–3131.
- [29] F. Cai and X. Zhang, "Efficient check node processing architectures for non-binary LDPC decoding using power representation," in *Proceedings of IEEE Workshop on Signal Processing Systems*, Oct. 2012, pp. 137–142.
- [30] X. Zhang, F. Cai, and S. Lin, "Low-complexity reliability-based message-passing decoder architectures for non-binary LDPC," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 11, pp. 1938–1950, Nov. 2012.
- [31] C. Xiong and Z. Yan, "Improved iterative soft-reliability-based majority-logic decoding algorithm for non-binary low-density parity-check codes," in *Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers*, Nov. 2011, pp. 894–898.
- [32] —, "Low-complexity layered iterative hard-reliability-based majority-logic decoder for non-binary quasi-cyclic LDPC codes," in *IEEE International Symposium on Circuits and Systems*, May 2013, pp. 1348–1351.
- [33] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.
- [34] T. Richardson, "Error floors of LDPC codes," in *Proceedings of Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2003, pp. 1426–1435.
- [35] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. J. Wainwright, "Lowering LDPC error floors by postprocessing," in *Proceedings of IEEE Global Telecommunications Conference*, 2008, pp. 1–6.
- [36] A. Poloni, S. Valle, and S. Vincenti, "NB-LDPC: Absorbing set and importance sampling," in *Proceedings of 7th International Symposium on Turbo Codes and Iterative Information Processing*, Aug. 2012, pp. 101–105.