Generalized Binary Representation for the Nonbinary LDPC Code With Decoder Design

Yang Yu, Wen Chen, *Senior Member, IEEE*, Jun Li, *Member, IEEE*, Xiao Ma, and Baoming Bai

***Abstract*—In this paper, we consider the performance-optimized nonbinary low-density parity check code over general linear group, i.e., $\bar{\mathcal{C}}$. A new methodology for constructing the binary representation [generalized binary representation (GBR)] of $\bar{\mathcal{C}}$ is proposed, which can be optimized with regard to both degree distributions and girth. As to the decoding of the GBR, we develop a low-complexity hybrid parallel decoding process. It is shown that the decoding performance of the GBR under the proposed binary decoding process could closely approach the decoding performance of its mother code $\bar{\mathcal{C}}$ under nonbinary belief propagation decoding. A simple code optimization algorithm for the GBR is also provided. Simulations show the comparative results and justify the advantages of the proposed constructions.**

***Index Terms*—Non-binary LDPC code, binary image, binary Gaussian channel, binary symmetric channel.**

I. Introduction

LOW density parity check (LDPC) codes, as a class of forward error control codes, have gained considerable attention during the last decade due to their amazing decoding performance under different channels [1], [2]. The performance of a long LDPC code is usually evaluated in terms of the threshold for the average performance of its code ensemble based on the cycle-free condition [1], [3]–[7].

Performance-optimized LDPC codes are designed by optimizing the degree structure of the Tanner graphs so that their thresholds could be very close to the Shannon capacity. In the mean time, these codes will suffer from performance degradation if there exist non-negligible number of short length cycles, especially for the short block length codes. Moreover, codes with large girths will have respectable minimum/stopping distance bound, which also implies enhanced decoding performance. In this paper, we refer to the cycles in the binary parity check matrices as *bit-level* cycles and the cycles in the non-binary parity check matrices as *symbol-level* cycles. In [8]–[10], the authors show how to construct the parity check matrices with less bit-level cycles and large girths for binary LDPC codes. For the non-binary LDPC codes, investigations indicate that they could have sparser Tanner graphs as the field size increases. For short to moderate block lengths, the non-binary LDPC codes with sparser graphs are more likely to outperform the binary ones. In [11], [12], the authors investigate a particular type of non-binary LDPC codes, i.e., non-binary cycle LDPC codes, whose column weights are two. In [11], optimizations for this type of codes are performed over Cayley-graph. In [12], the authors propose bit-level coefficients selection methods to optimize the symbol-level performance for the non-binary cycle LDPC codes.

On the other hand, belief propagation (BP) decoding for the non-binary LDPC codes requires a potentially higher complexity. The complexity of the q-ary sum-product decoding algorithm (QSPA) is $O(q^2)$ for each check-sum operation. The Fourier transform QSPA reduces the complexity to $O(q \log q)$ [5]. The extended min-sum (EMS) algorithm in [13] further reduces the complexity to $O(n_m \log n_m)$ at the cost of a bit performance loss, where n_m is smaller than q. However, the computational complexity of the EMS decoder is still very high compared to the binary decoder. Hence, in [14], [15], the authors propose an extended binary representation for the non-binary LDPC code which can be decoded by binary decoders. The binary computational complexity is only $O(q)$ for BEC. Theoretically, based on the decoding error probability, the authors in [16], [17] prove that the minimal decoding complexities exist if the LDPC codes are constructed with properly chosen degree distributions.

A. Related Works

The codewords of a non-binary LDPC code are often transmitted over binary input channels in their bit-vector forms, i.e., binary images of the non-binary LDPC codes. At the receiver side, the non-binary decoder needs to transform the received bit sequences back to their non-binary forms to perform the symbol-level decoding [2], [6], [12], [18], [19] for retrieving the information bits. On the other hand, as an alternative of using the non-binary decoders for binary input channels, one can use a binary decoder to retrieve the information bits by utilizing the binary representations of the non-binary parity check matrices

Manuscript received August 27, 2013; revised February 16, 2014 and June 28, 2014; accepted July 19, 2014. Date of publication July 30, 2014; date of current version September 19, 2014. This work was supported in part by the National 973 Project under Grant 2012CB316106, by NSF China under Grants 61161130529 and 61328101, by the STCSM Science and Technology Innovation Program under Grant 13510711200, and by the SEU National Key Lab on Mobile Communications under Grant 2013D11. The associate editor coordinating the review of this paper and approving it for publication was K. Abdel-Ghaffar.

Y. Yu and W. Chen are with Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: yuyang83@sjtu.edu.cn; wenchen@sjtu.edu.cn).

J. Li is with the School of Electrical and Information Engineering, The University of Sydney, Sydney, N.S.W. 2006, Australia (e-mail: jun.li1@sydney.edu.au).

X. Ma is with the Department of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: maxiao@mail.sysu.edu.cn).

B. Bai is with the State Key Laboratory of Integrated Services Network, Xidian University, Xi'an 710071, China (e-mail: bmbai@mail.xidian.edu.cn).

Digital Object Identifier 10.1109/TCOMM.2014.2344912

0090-6778 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

for the purpose of reducing the computational complexity [14], [15], [20]. Especially in certain cases, when the receiver receives a non-binary codeword from the binary input channels and only limited computational resources are available, the consideration of using binary decoders is natural and practical for a fast and correct information recovery. However, the binary representation of a non-binary parity check matrix has numerous bit-level cycles, even if there is no symbol-level cycle [14], [20] in the non-binary parity check matrix. Thus, in [14], [15], the authors introduce the (punctured) extended binary representation for the non-binary LDPC code to solve this issue. When there is no symbol-level cycle, this representation will also be cycle-free. In [20], the authors propose a hybrid hard decision decoder particularly for the BEC which eliminates the local decoding cycles by introducing matrix inverse operations. In addition, the authors in [21] show how to optimize the binary representation of a non-binary parity check matrix with the perspective of stopping set.

B. Contributions

In this paper, we focus on the performance-optimized $\bar{\mathcal{C}}$ (the non-binary LDPC code over general linear group). We aim at further improving the bit-level decoding performance and reducing the bit-level computational complexity. To this end, we develop a hybrid parallel decoding process over binary input Gaussian channel to achieve enhanced decoding performance and propose a new methodology to construct the binary representation for $\bar{\mathcal{C}}$ which can be optimized with regard to both girth and irregular code profile (degree distributions). Contributions of this paper are summarized as follows.

1) We first give an extended iterative hard decision decoder (EHDD) over binary symmetric channel (BSC). Then, by allowing the EHDD and binary BP decoder working iteratively, we develop a hybrid parallel decoder (HPD) for the GBR. The bit-level computational complexity is dominated by $O(m_s), m_s < q$. Systematic investigation of the proposed decoders is also carried out. It is shown that the low complexity bit-level decoding (HPD) could perform closely to the symbol-level decoding for $\bar{\mathcal{C}}$. A simple code optimization algorithm for these binary decoders is also provided.
2) We propose a generalized binary representation (GBR) for $\bar{\mathcal{C}}$ which can be optimized with regard to both girth and irregular code profile (primarily the irregular code profile). A general approach is given to study the constructions and optimizations of the GBR. Significant results and conditions regarding the constructions and optimizations are also derived.

C. Organization of the Paper

The contents of this paper are organized as follows. In Section II, we introduce the binary representations of the non-binary LDPC code and give a unified framework for the extended binary representation. In Section III, we give the details about the GBR. In Section IV, we give the decoder design, carry out the systematic investigation of the proposed decoders and provide a simple code optimization algorithm. Section V presents the simulation results.

II. Binary Representations for Non-Binary LDPC Codes

A. Binary Images for Non-Binary LDPC Codes

We denote the finite field of size $q = 2^p$ by $\mathbb{F}_q$ and the column vector space of dimension-N over $\mathbb{F}_q$ by $\mathbb{F}_q^N$. Let $\mathbb{F}_q^* = \mathbb{F}_q \backslash \{0\}$. We assume that $\mathbb{F}_q$ is endowed with a binary vector space structure. Every $u \in \mathbb{F}_q$ can be denoted by a binary vector

$$\bar{\mathbf{u}} = (\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_{p-1})^T \in \mathbb{F}_2^p,$$

i.e., the *binary image* of u. We denote the general linear group over $\mathbb{F}_2$ by $\mathrm{GL}(p, \mathbb{F}_2)$ whose elements are $p \times p$ invertible matrices with entries taken from $\mathbb{F}_2$.

A non-binary LDPC code $\mathcal{C}$ of length N is the dimension $N - M$ linear subspace of $\mathbb{F}_q^N$. Its parity check matrix is denoted by

$$\mathbf{H} = \{h_{i,j}\}_{M \times N}, h_{i,j} \in \mathbb{F}_q.$$

Then $\mathcal{C}$ is defined as the kernel of $\mathbf{H}$.

The non-binary LDPC code $\bar{\mathcal{C}}$ defined over $\mathrm{GL}(p, \mathbb{F}_2)$ is generalized from $\mathcal{C}$ [22]. The code symbols of $\bar{\mathcal{C}}$ are elements in $\mathbb{F}_2^p$. A codeword is constituted of N symbols. The parity check matrix of $\bar{\mathcal{C}}$ is an $M \times N$ matrix with each non-zero entry being an element in $\mathrm{GL}(p, \mathbb{F}_2)$. By using the binary vector notation, we denote the binary image of its codeword as

$$\bar{\mathbf{x}} = \left(\bar{\mathbf{x}}_1^T, \bar{\mathbf{x}}_2^T, \ldots, \bar{\mathbf{x}}_N^T\right)^T,\ \bar{\mathbf{x}}_j \in \mathbb{F}_2^p,\ j = 1, 2, \ldots, N.$$

The equivalent binary parity check matrix for $\bar{\mathcal{C}}$ is denoted by

$$\bar{\mathbf{H}} = (\mathbf{A}_{i,j})_{M \times N}, \mathbf{A}_{i,j} \in \mathrm{GL}(p, \mathbb{F}_2) \cup \{\mathbf{0}\}.$$

The non-binary LDPC code $\mathcal{C}$ is a particular case of $\bar{\mathcal{C}}$ in the sense that $\mathbb{F}_q \cong \mathbb{F}_2^p$ and the non-zero entries in $\mathbf{H}$ can be represented by the powers of the companion matrix over $\mathbb{F}_q$ [20], [22]–[24]. With a little abuse of the notation, in the following, we denote any binary parity check matrix over $\mathbb{F}_2$ by $\bar{\mathbf{H}}$ and any non-binary parity check matrix over $\mathbb{F}_q$ by $\mathbf{H}$. We also define $\mathrm{diag}(\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_N)$ as the matrix

$$\mathrm{diag}(\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_N) = \begin{pmatrix} \mathbf{B}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B}_N \end{pmatrix},$$

where $\mathbf{B}_j,\ j = 1, 2, \ldots, N$, are not necessarily to be square matrices.

B. Extended Binary Representation for Non-Binary LDPC Codes

In this subsection, we give a unified framework for the extended binary representation. We denote the set of natural

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

integers including 0 by $\mathbb{N}$, and define $\mathbb{N}^* = \mathbb{N}\backslash\{0\}$. Let $\mathbb{N}_q = \{0, 1, \ldots, q-1\}$ and $\mathbb{N}_q^* = \mathbb{N}_q\backslash\{0\}$. For an arbitrary matrix $\mathbf{B}$, we denote the entries of $\mathbf{B}$ by $\mathbf{B}(i,j), i,j \in \mathbb{N}$, where i and j are the row number and column number, respectively. In addition, $\mathbf{B}(i,0)$ represents the ith row vector, $\mathbf{B}(0,j)$ represents the jth column vector. We denote the $p \times p$ identity matrix by $\mathbf{I}_{p\times p}$. The extended representation begins with a linear transformation of a binary vector $\bar{\mathbf{x}}_j \in \mathbb{F}_2^p$ [14].

We define $\mathbf{\Phi}$ as the $p \times (q-1)$ binary matrix of the following form

$$\mathbf{\Phi} = \left(\mathbf{\Phi}(0,1), \mathbf{\Phi}(0,2), \ldots, \mathbf{\Phi}(0,q-1)\right),$$

where each column vector $\mathbf{\Phi}(0,j)$, $j = 1, 2, \ldots, q-1$, is the binary representation of $j \in \mathbb{N}_q^*$. For the binary image of the jth coded symbol, i.e., $\bar{\mathbf{x}}_j$, we have

$$\mathbf{v}_j = \mathbf{\Phi}^T \bar{\mathbf{x}}_j \in \mathbb{F}_2^{q-1}.$$

Note that $\mathbf{\Phi}$ is the parity check matrix of the $[q-1, q-1-p]$ hamming code. So, each $\mathbf{v}_j$ is also a codeword of the simplex code (dual code of the hamming code). The *extended binary representation* (EBR) of $\bar{\mathbf{x}}$ is then

$$\mathbf{v} = \left(\mathbf{v}_1^T, \ldots, \mathbf{v}_N^T\right)^T.$$

The EBR of $\bar{\mathcal{C}}$ is defined as the vector space constituted of those $\mathbf{v}$s transformed from the binary images of all the codewords of $\bar{\mathcal{C}}$. In addition, for each non-zero $\mathbf{A}_{i,j}$, we can get a $(q-1)\times(q-1)$ matrix $\mathbf{\Omega}_{i,j}$ while satisfying an endomorphism of $\mathbb{N}_q$ and an isomorphism between $\mathbb{N}_q$ and $\mathbb{F}_2^p$ [14]. If we replace the non-zero $\mathbf{A}_{i,j}$ in $\bar{\mathbf{H}}$ by $\mathbf{\Omega}_{i,j}$ and the zero $\mathbf{A}_{i,j}$ by $\mathbf{0}_{(q-1)\times(q-1)}$, we get the extended binary parity check matrix $\mathbf{\Omega} = (\mathbf{\Omega}_{i,j})_{M\times N}$. Then $\mathbf{\Omega}\mathbf{v} = \mathbf{0}$ and the simplex constraints on $\mathbf{v}$ together form the extended binary representation. The decoding applications of the extended binary representation over general channel models are given in [22].

III. Generalized Binary Representation for Non-Binary LDPC Codes

In this section, we introduce the generalized binary representation (GBR) for the non-binary LDPC codes over General linear group. We will also discuss the constructions and optimizations of the GBR.

A. Definition of the Generalized Binary Representation

We first give the definitions that will be used in the following sections. We define $\mathrm{wt}(\cdot)$ as the function that calculates the number of non-zero columns in a matrix or of the non-zero elements in a vector.

Definition 1: The *mother matrix* $\mathbf{\Lambda}_p$ of a binary matrix $\bar{\mathbf{H}}$ over $\mathbb{F}_2$ or of a non-binary matrix $\mathbf{H}$ over $\mathbb{F}_q$ is defined as a matrix with each entry being either 0 or 1. The binary matrix $\bar{\mathbf{H}}$ can be obtained by replacing the 0s by $\mathbf{0}$ matrices of size $p \times p$ and the 1s by non-zero matrices of size $p \times p$. These $p \times p$ matrices are also referred to as the *matrix labels* of $\bar{\mathbf{H}}$. The non-binary matrix $\mathbf{H}$ can be obtained by replacing the 0s in $\mathbf{\Lambda}_p$ by the zero element in $\mathbb{F}_{2^p}$ and the 1s by the non-zero elements in $\mathbb{F}_{2^p}$. Cycles in $\mathbf{\Lambda}_p$ or $\mathbf{H}$ are referred to as the *symbol-level* cycles. Cycles in $\bar{\mathbf{H}}$ are referred to as the *bit-level* cycles.

Recall that, in Section II-A, the equivalent binary parity check matrix $\bar{\mathbf{H}}$ for the non-binary LDPC code $\bar{\mathcal{C}}$ over general linear group $\mathrm{GL}(p, \mathbb{F}_2)$ can be expressed as $(\mathbf{A}_{i,j})_{M\times N}$. Each $\mathbf{A}_{i,j}$ is either a $p \times p$ zero matrix or a $p \times p$ full-rank matrix. Then, the $\mathbf{A}_{i,j}$s are referred to as the matrix labels of $\bar{\mathbf{H}}$.

Definition 2 ($\preccurlyeq$)*:* We denote the relationship between two vectors $\mathbf{a}, \mathbf{b}$ by $\mathbf{a} \preccurlyeq \mathbf{b}$ if $\mathbf{a}$ is obtained by replacing some elements in $\mathbf{b}$ by zeros. For two matrices $\mathbf{A}, \mathbf{B}$, we denote $\mathbf{A} \preccurlyeq \mathbf{B}$ if $\mathbf{A}$ is obtained by replacing some column vectors in $\mathbf{B}$ by zero vectors.

Definition 3 ($\prec$)*:* We denote the relationship between two vectors $\mathbf{a}, \mathbf{b}$ by $\mathbf{a} \prec \mathbf{b}$ if $\mathbf{a} \preccurlyeq \mathbf{b}$ and $\mathrm{wt}(\mathbf{a}) < \mathrm{wt}(\mathbf{b})$. For two matrices $\mathbf{A}, \mathbf{B}$, we denote the relationship between them by $\mathbf{A} \prec \mathbf{B}$ if $\mathbf{A} \preccurlyeq \mathbf{B}$ and $\mathrm{wt}(\mathbf{A}) < \mathrm{wt}(\mathbf{B})$.

Below, we first define $\mathbf{\Psi} = \{\mathbf{\Psi}_j, j = 1, 2, \ldots, N\}$ as the *extended generator matrices set*. Each $\mathbf{\Psi}_j$ is a full-rank binary matrix with p rows and p'_j columns, where $p \leqslant p'_j \leqslant q-1$. The non-zero columns in each $\mathbf{\Psi}_j$ are different from each other. Then, for the binary image of the codeword of $\bar{\mathcal{C}}$, i.e., $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_1^T, \bar{\mathbf{x}}_2^T, \ldots, \bar{\mathbf{x}}_N^T)^T, \bar{\mathbf{x}}_j \in \mathbb{F}_2^p, j = 1, 2, \ldots, N$, we have

$$\mathbf{v}^e = \mathrm{diag}\left(\mathbf{\Psi}_1^T, \mathbf{\Psi}_2^T, \ldots, \mathbf{\Psi}_N^T\right) \cdot \bar{\mathbf{x}}, \tag{1}$$

where $\mathbf{v}^e = (\mathbf{v}_1^{eT}, \mathbf{v}_2^{eT}, \ldots, \mathbf{v}_N^{eT})^T$.

Definition 4: Given the extended generator matrices set $\mathbf{\Psi}$, the *generalized binary representation* (GBR) of the non-binary LDPC code $\bar{\mathcal{C}}$ over general linear group is defined as the vector space constituted of all the $\mathbf{v}^e$s (which are transformed from the binary images of all the codewords of $\bar{\mathcal{C}}$ according to (1)). Moreover, we refer to $\mathbf{\Psi}_j(0, 2^{i-1}) \neq \mathbf{0}, \forall i \in \{1, 2, \ldots, p\}$ as the *trivial* case for the GBR of $\bar{\mathcal{C}}$.

Recall that $\mathbf{\Phi}$ is the generator matrix of the extended binary representation (EBR), and the codeword of the EBR is denoted by $\mathbf{v}$. Since $\mathbf{\Psi}_j$ has different non-zero vectors as its columns and $\mathbf{\Phi}$ has all the non-zero vectors in $\mathbb{F}_2^p$ as its columns, the non-zero column vectors in each $\mathbf{\Psi}_j$ form a subset of the column vectors in $\mathbf{\Phi}$. In the following, without loss of generality, we assume that $\mathbf{\Psi}_j \preccurlyeq \mathbf{\Phi}$ for all $j \in \{1, 2, \ldots, N\}$. Then, $\mathbf{v}_j^e \preccurlyeq \mathbf{v}_j$. Since the zero columns in $\mathbf{\Psi}_j$ will result in zero bits in $\mathbf{v}_j^e$ which can be ignored or readily removed, this assumption does not violate Definition 4 and will facilitate the discussion of the GBR too.

B. Exhaustive Search for the Desired Parity Check Matrix

The bits in $\mathbf{v}_j^e$, $j \in \{1, 2, \ldots, N\}$ represent different additions of the bits in $\bar{\mathbf{x}}_j$. Then, by finding the parity check relationships for different combinations of these additions, we could establish the parity check relationships for $\mathbf{v}^e$. We denote the parity check matrix for $\mathbf{v}^e$ by $\mathbf{\Omega}^e = (\mathbf{\Omega}_{i,j}^e)_{M\times N}$ where each $\mathbf{\Omega}_{i,j}^e$ is a $(q-1)\times(q-1)$ binary matrix. Then, the desired $\mathbf{\Omega}^e$ can be in general constructed by searching among different combinations of the parity check relationships for $\mathbf{v}^e$.

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

Definition 4 may imply that we should search for $\mathbf{\Omega}^e$ based on a given $\mathbf{\Psi}$. However, in order to guarantee enhanced decoding performance for $\mathbf{\Omega}^e$, we first determine the desired $\mathbf{\Omega}^e$ then we update $\mathbf{\Psi}$. That is,

1) We construct a set $\mathbf{S}$ whose elements are the rows of $\mathbf{\Omega}$, the rows established according to different combinations of the simplex parity check relations and the zero row.
2) By using the elements in $\mathbf{S}$, we construct different $\mathbf{\Omega}^e$s row by row such that the new row does not introduce cycles smaller than certain integer.
3) Among the constructed parity check matrices, we find the $\mathbf{\Omega}^e$ with desired performance threshold. Then, we update $\mathbf{\Psi}$ and $\mathbf{v}^e$.

For the non-binary LDPC code $\bar{C}$, there is only one associated EBR with the parity check matrix $\mathbf{\Omega}$ [14], [22]. However, based on the above searching process, we could establish many GBRs for $\bar{C}$ whose parity check matrices may be obtained by changing the matrix labels or the structure of $\mathbf{\Omega}$. This approach is different from the work in [14], [15], [21] because it generally results in non-trivial binary presentations of the code $\bar{C}$. Like the work in [22], we can decode these GBRs with a low-complexity binary decoder without changing the transmitted codewords $\bar{\mathbf{x}}$, i.e., the underlying code is not changed.

C. Mapping Definition and Examples

In this subsection, we introduce a matrix map f_ω to provide more details about establishing the parity check relations for $\mathbf{v}^e$ and more insights into formulating the constructions of $\mathbf{\Omega}^e$. Consider the parity check matrix $\bar{\mathbf{H}} = (\mathbf{A}_{i,j})_{M\times N}$. Let $\mathbf{B}$ be a binary matrix of size $p \times (q-1)$. With a little abuse of notation, we use $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$ to denote the resulting binary matrix and $f_\omega(i', j'), i', j' = 1, 2, \ldots, q-1$ to denote the entries in $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$. Then

$$f_\omega(i', j') = \begin{cases} 1, & \text{if } \mathbf{B}(0, j') + \mathbf{A}_{i,j}^T \mathbf{\Phi}(0, i') = \mathbf{0}, \\ 0, & \text{if } \mathbf{B}(0, j') + \mathbf{A}_{i,j}^T \mathbf{\Phi}(0, i') \neq \mathbf{0}. \end{cases}$$

The matrix map f_ω defined above can be used to represent different parity check relations for the bits in $\mathbf{v}^e$. More specifically, different columns of $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$ associate with different bits in $\mathbf{v}_j^e$. Different rows of $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$ denote different additions between the bits in $\mathbf{v}_j^e$. To have a better understanding, we give simple examples for f_ω below.

Example 1: The additions between different binary parity check equations within $\bar{\mathbf{H}}_i^T \bar{\mathbf{x}} = \mathbf{0}, i \in \{1, 2, \ldots, M\}$ can be formulated as $\mathbf{\Phi}^T \bar{\mathbf{H}}_i^T \bar{\mathbf{x}} = \mathbf{0}$ which will result in $q-1$ different binary parity check equations [14], [22]. We divide the $q-1$ binary parity check equations into N partitions with the jth partition consisting of $q-1$ different additions of the bits in $\bar{\mathbf{x}}_j$, i.e., $\mathbf{\Phi}^T \mathbf{A}_{i,j} \bar{\mathbf{x}}_j$. As a result, these equations denote $q-1$ parity check relations for $\mathbf{v}$. If we set some of the $q-1$ equations to be zero equations, then there exist only one binary matrix $\mathbf{B}$ for the jth partition such that the $q-1$ rows of $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$ respectively represent the $q-1$ rows within the jth partition, e.g., if $p = 3$ and $\mathbf{A}_{i,j} = (\mathbf{\Phi}(0,3), \mathbf{\Phi}(0,6), \mathbf{\Phi}(0,7))$, then

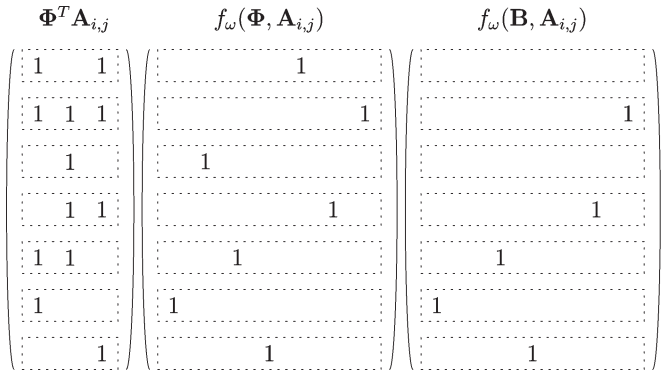

Fig. 1. Different matrices generated by f_ω in Example 1.

$f_\omega(\mathbf{\Phi}, \mathbf{A}_{i,j}) = \mathbf{\Omega}_{i,j}$, as displayed in Fig. 1. If we set the first and third rows in $\mathbf{\Omega}_{i,j}$ to be zero vectors, then we have

$$\begin{aligned} \mathbf{B} = {} & (\mathbf{\Phi}(0,1), \mathbf{0}, \mathbf{\Phi}(0,3), \mathbf{\Phi}(0,4), \mathbf{0}, \mathbf{\Phi}(0,6), \\ & \mathbf{\Phi}(0,7)) \prec \mathbf{\Phi}, \\ f_\omega(\mathbf{B}, \mathbf{A}_{i,j}) = {} & \big(\mathbf{0}, \mathbf{\Omega}_{i,j}(2,0)^T, \mathbf{0}, \mathbf{\Omega}_{i,j}(4,0)^T, \mathbf{\Omega}_{i,j}(5,0)^T, \\ & \mathbf{\Omega}_{i,j}(6,0)^T, \mathbf{\Omega}_{i,j}(7,0)^T\big)^T \prec \mathbf{\Omega}_{i,j}. \end{aligned}$$

Note that each $\mathbf{v}_j^e$ is a codeword generated by $\mathbf{\Psi}_j$. Since different columns of $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$ associate with different bits in $\mathbf{v}_j^e$, $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$ can be also used to represent some simplex parity check relations for $\mathbf{v}_j^e$. The construction of such matrices is trivial, so we leave it for briefness.

With the introduced f_ω, we can model the exhaustive searching processes (Step 2 and Step 3 in Section III-B) for the desired $\mathbf{\Omega}^e$ as follows. 1) For each $\mathbf{A}_{i,j}$ in $\bar{\mathbf{H}}$, we search for proper binary matrices $\mathbf{C}_h, \forall h \in \{1, 2, \ldots, q-1\}$ with size $p \times (q-1)$. Moreover, $f_\omega(\mathbf{C}_h, \mathbf{A}_{i,j}) \cdot \mathbf{\Psi}_j^T = \mathbf{\Phi}^T(0, h), h \in \{1, 2, \ldots, q-1\}$ or $f_\omega(\mathbf{C}_h, \mathbf{A}_{i,j}) \cdot \mathbf{\Psi}_j^T = \mathbf{0}$.2) Then $\mathbf{\Omega}^e$ is obtained by replacing each $\mathbf{A}_{i,j}$ with $\sum_{h=1}^{q-1} f_\omega(\mathbf{C}_j, \mathbf{A}_{i,j})$, where $\sum$ is the modulo-2 sum and each $f_\omega(\mathbf{C}_h, \mathbf{A}_{i,j})$ corresponds to a row in $\mathbf{\Omega}_{i,j}^e$ (some of $\mathbf{C}_h$s could be zero matrices). If each $\mathbf{A}_{i,j}$ is replaced by $\sum_{h=1}^{q-1} f_\omega(\mathbf{C}_j, \mathbf{A}_{i,j}) = f_\omega(\mathbf{\Phi}, \mathbf{A}_{i,j})$, the resulting matrix is the parity check matrix $\mathbf{\Omega}$ for the EBR. Another example of f_ω is that, by assuming $\mathbf{B} \preccurlyeq \mathbf{\Phi}$, we replace each $\mathbf{A}_{i,j}$ with $f_\omega(\mathbf{B}, \mathbf{A}_{i,j})$. Then, the construction of the resulting $\mathbf{\Omega}^e$ is equivalent to removing some rows (and some columns) of $\mathbf{\Omega}$.

D. Properties of the Matrix Mapping

Lemma 1: Let $\mathbf{B} \preccurlyeq \mathbf{\Phi}$ and $\mathbf{B}' \preccurlyeq \mathbf{\Phi}$ be two $p \times (q-1)$ binary matrices. Let $\mathbf{C}$ be a $p \times p$ full-rank binary matrix. $f_\omega(\mathbf{\Phi}, \mathbf{C})$ is a $(q-1) \times (q-1)$ permutation matrix. In addition, $\mathbf{B}' \preccurlyeq \mathbf{B}$ and $f_\omega(\mathbf{B}', \mathbf{C}) \preccurlyeq f_\omega(\mathbf{B}, \mathbf{C})$ are necessary and sufficient conditions for each other.

Proof: Since $\mathbf{C}$ is a $p \times p$ full rank matrix, all the $\mathbf{C}^T \mathbf{\Phi}(0, i'), i' = 1, 2, \ldots, q-1$ are different column vectors. Then $f_\omega(\mathbf{\Phi}, \mathbf{C})$ will have only one non-zero entry in each row or column. So, $f_\omega(\mathbf{\Phi}, \mathbf{C})$ is a $(q-1) \times (q-1)$ permutation matrix. If $\mathbf{B} \preccurlyeq \mathbf{\Phi}$, the zero columns in $\mathbf{B}$ will result in zero rows in $f_\omega(\mathbf{B}, \mathbf{C})$. Then $f_\omega(\mathbf{B}, \mathbf{C})$ can be obtained by setting some

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

rows of $f_\omega(\mathbf{\Phi}, \mathbf{C})$ to be zero vectors. Since $f_\omega(\mathbf{\Phi}, \mathbf{C})$ have only one non-zero entry in each column, then some columns become zero vectors in $f_\omega(\mathbf{B}, \mathbf{C})$. As a result $f_\omega(\mathbf{B}, \mathbf{C}) \preccurlyeq f_\omega(\mathbf{\Phi}, \mathbf{C})$. Similarly, we have $f_\omega(\mathbf{B}', \mathbf{C}) \preccurlyeq f_\omega(\mathbf{B}, \mathbf{C})$ if $\mathbf{B}' \preccurlyeq \mathbf{B}$. Conversely, if $f_\omega(\mathbf{B}, \mathbf{C}) \preccurlyeq f_\omega(\mathbf{\Phi}, \mathbf{C})$, it means that the columns in $\mathbf{B}$ generating the zero rows in $f_\omega(\mathbf{B}, \mathbf{C})$ are set to be zero vectors. Since there is a one-to-one correspondence between $\mathbf{B}$ and $f_\omega(\mathbf{B}, \mathbf{C})$, we have $\mathbf{B} \preccurlyeq \mathbf{\Phi}$. Similarly, we have $\mathbf{B}' \preccurlyeq \mathbf{B}$ if $f_\omega(\mathbf{B}', \mathbf{C}) \preccurlyeq f_\omega(\mathbf{B}, \mathbf{C})$. This completes the proof. ■

When each $\mathbf{A}_{i,j}$ in $\bar{\mathbf{H}}$ is replaced by $f_\omega(\mathbf{\Phi}, \mathbf{A}_{i,j})$, we denote the resulting matrix by

$$\begin{aligned}\mathbf{\Omega} &= (\mathbf{\Omega}_{i,j})_{M\times N} \\ &= (\mathbf{\Omega}_1, \mathbf{\Omega}_2, \ldots, \mathbf{\Omega}_M)^T = (\mathbf{\Omega}_1^c, \mathbf{\Omega}_2^c, \ldots, \mathbf{\Omega}_N^c),\end{aligned}$$

where $\mathbf{\Omega}_i$ is the $(q-1)N \times (q-1)$ sub-matrix and $\mathbf{\Omega}_j^c$ is the $(q-1)M \times (q-1)$ sub-matrix of $\mathbf{\Omega}$. According to Lemma 1, we also have the following properties of $\mathbf{\Omega}$.

Lemma 2:

1) For all the non-zero $\mathbf{A}_{i,j}, i \in \{1, 2, \ldots, M\}$, $j \in \{1, 2, \ldots, N\}$, the corresponding $\mathbf{\Omega}_{i,j}$ is a $(q-1) \times (q-1)$ permutation matrix.
2) $\mathbf{\Omega}$ inherits the node degrees of $\mathbf{\Lambda}_p$. That is, row weights of $\mathbf{\Omega}_i^T$ are the same and equal to the weight of $\mathbf{\Lambda}_p(i, 0)$. The column weights of $\mathbf{\Omega}_j^c$ are equal to the weight of $\mathbf{\Lambda}_p(0, j)$. Degree distributions of $\mathbf{\Omega}$ are the same as those of $\mathbf{\Lambda}_p$.

E. Bit-Level Cycles in $\mathbf{\Omega}$

In this subsection, we investigate the relations between the symbol-level cycles in $\mathbf{\Lambda}_p$ and the bit-level cycles in $\mathbf{\Omega}$ based on the properties of f_ω. In general, we assume that $\mathbf{\Lambda}_p$ is of girth g_h. $\mathbf{\Lambda}_p$ is cycle-free if $g_h = 0$. Next, we first give the definition for the matrix cycle.

Definition 5 (Matrix Cycle): Given the binary parity check matrix $\bar{\mathbf{H}}$. Let $\mathbf{\Lambda}_p$ be its mother matrix. A *matrix cycle* of length-g in $\bar{\mathbf{H}}$ exists iff its corresponding positions in $\mathbf{\Lambda}_p$ form a symbol-level cycle of length-g.

Lemma 3: If the girth of the mother matrix $\mathbf{\Lambda}_p$ is $g_h > 0$, then the girth of its associated parity check matrix $\mathbf{\Omega}$ is $g_s \geqslant g_h$. If $g_h = 0$, $g_s = 0$.

Proof: Since $\mathbf{\Omega}_{i,j}$ is a $(q-1) \times (q-1)$ permutation matrix and cycle-free (due to the first item in Lemma 2), if $\mathbf{\Lambda}_p$ satisfies the cycle-free condition, $\mathbf{\Omega}$ will also be cycle-free. Moreover, a cycle in $\mathbf{\Lambda}_p$ will only cause a matrix cycle in $\mathbf{\Omega}$ with the same length. When $\mathbf{\Omega}_{i,j}$s are equal to $\mathbf{I}_{(q-1)\times(q-1)}$, a matrix cycle of length g_h will always and only cause bit-level cycles with the same length. Otherwise, the matrix cycle will not cause bit-level cycles with length g_h at certainty. Thus, the girth of the binary parity check matrix $\mathbf{\Omega}$ is not smaller than the girth of its mother matrix $\mathbf{\Lambda}_p$. ■

The above lemma implies that, for $\mathbf{H}$ over $\mathbb{F}_q$, the girth of its associated $\mathbf{\Omega}$ is also not smaller than its girth. Moreover, investigations indicate that the length-4 cycles contribute the most to the performance degradation. Next, we show that a length-4 symbol-level cycle in $\mathbf{H}$ will not always result in length-4 bit-level cycles in $\mathbf{\Omega}$.

Theorem 4: Let the non-zero matrix labels be uniformly taken from $\mathbb{F}_q^*$. The probability that a length-4 symbol-level cycle in the non-binary parity check matrix $\mathbf{H}$ will result in length-4 bit-level cycles in $\mathbf{\Omega}$ is denoted by p_4. Then

$$p_4 = \frac{1}{q-1}$$

for $q = 2^p \geqslant 4$.

Proof: Since the length-4 bit-level cycles are only caused by the length-4 symbol level cycle, we only consider the bit-level cycles within a symbol-level cycle. Let $(i_1, j_1), (i_1, j_2), (i_2, j_1), (i_2, j_2)$ be the four coordinates of four entries that represent a length-4 symbol level cycle in $\mathbf{H}$. We denote

$$\begin{pmatrix} \mathbf{\Omega}_{i_1,j_1} & \mathbf{\Omega}_{i_1,j_2} \\ \mathbf{\Omega}_{i_2,j_1} & \mathbf{\Omega}_{i_2,j_2} \end{pmatrix}$$

as the matrix cycle corresponding to a length-4 symbol-level cycle. We use $\alpha_1, \beta_1, \alpha_2, \beta_2 \in \{1, 2, \ldots, q-1\}$ to respectively represent the column numbers of non-zero entries in $\mathbf{\Omega}_{i_1,j_1}$, $\mathbf{\Omega}_{i_1,j_2}$, $\mathbf{\Omega}_{i_2,j_1}$, and $\mathbf{\Omega}_{i_2,j_2}$ with α_1, β_1 in the same row and α_2, β_2 in the same row. We denote

$$\mathscr{S}_1 = \{(\alpha_1, \beta_1), \alpha_1, \beta_1 \in \{1, 2, \ldots, q-1\}\}$$

and

$$\mathscr{S}_2 = \{(\alpha_2, \beta_2), \alpha_2, \beta_2 \in \{1, 2, \ldots, q-1\}\}$$

as the two-tuple sets containing all the different rows in $(\mathbf{\Omega}_{i_1,j_1}, \mathbf{\Omega}_{i_1,j_2})$ and $(\mathbf{\Omega}_{i_2,j_1}, \mathbf{\Omega}_{i_2,j_2})$, respectively. Then,

$$|\mathscr{S}_1| = |\mathscr{S}_2| = q - 1.$$

We denote $\mathscr{S}$ as the set containing all the rows that could be involved in the length-4 matrix cycles. Then

$$\mathscr{S} = \{(\alpha, \beta), \alpha, \beta = 1, 2, \ldots, q-1\}$$

and $|\mathscr{S}| = (q-1)^2$ with $\mathscr{S}_1, \mathscr{S}_2 \subset \mathscr{S}$. The length-4 bit-level cycle exist iff

$$\mathbf{Pr}(\mathscr{S}_\mathbf{1} \cap \mathscr{S}_\mathbf{2} \neq \emptyset) = 1 - \mathbf{Pr}(\mathscr{S}_\mathbf{1} \cap \mathscr{S}_\mathbf{2} = \emptyset).$$

We can calculate the probability of $\mathscr{S}_1 \cap \mathscr{S}_2 = \emptyset$ by counting the number of choices of $\mathscr{S}_1$ and $\mathscr{S}_2$ over $\mathscr{S}$. Since there are $q-1$ different non-zero $\mathbf{\Omega}_{i,j}$s, different $\mathbf{\Omega}_{i,j}$s have different row numbers of the same row-vectors and no two different $\mathscr{S}_i$s have common elements, different $\mathscr{S}_i$s divide $\mathscr{S}$ into $q-1$ disjoint subsets. And because each $\mathscr{S}_i$ is uniformly chosen, then for a $\mathscr{S}_1$, there exist $(q-2)$ $\mathscr{S}_2$s that do not form cycles. As a result,

$$\mathbf{Pr}(\mathscr{S}_\mathbf{1} \cap \mathscr{S}_\mathbf{2} = \emptyset) = \frac{(q-1)(q-2)}{(q-1)^2}.$$

■

Corollary 5: For the matrix $\bar{\mathbf{H}}$, let its matrix labels be chosen uniformly over a set $\{\mathbf{B}_g, g = 1, 2, \ldots, Q\}$. If there exist a largest integer $P \leqslant Q$ such that $\text{rank}(f_\omega(\mathbf{\Phi}, \mathbf{B}_{g_i}) + f_\omega(\mathbf{\Phi}, \mathbf{B}_{g_j})) = q - 1$ for all $i \neq j, i, j \in \{1, 2, \ldots, P\}$, then

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

the probability that a length-4 symbol-level cycle in $\mathbf{\Lambda}_p$ will result in length-4 bit-level cycles in $\mathbf{\Omega}$, i.e., p'_4, satisfies

$$\frac{1}{q-1} \leqslant p'_4 \leqslant \frac{1+(Q-P)^2}{P+(Q-P)^2} \tag{2}$$

and $P \leqslant q-1$ for $q=2^p \geqslant 4$. When $P=1$, $p'_4=1$.

Proof: The P matrix labels result in at most $q-1$ disjoint subsets of $\mathscr{S}$ then $P \leqslant q-1$. The proof for the above inequality which results from the different values of $Q-P$ is similar to the proof of Theorem 4. ■

According to Corollary 5, p'_4 can be minimized by enlarging q and minimizing $Q-P$. Consider a short length matrix cycle of length-$g_c, g_c \geqslant 4$. Based on the proof of Theorem 4, we suppose that the probability of the existence of corresponding bit-level cycles of length-g_c relates to both q and g_c. We also have the following observation for the short length symbol-level cycles with lengths $g_c \geqslant 4$.

Observation 1:

1) For a code in Corollary 5, the probability that a symbol-level cycle of length-g_c in $\mathbf{\Lambda}_p$ will cause corresponding bit-level cycles of length-g_c in $\mathbf{\Omega}$ is greater than or equal to $1/(q-1)$.
2) This probability increases as the length of the symbol-level cycle increases and decreases as $q=2^p$ increases.

F. Construction of $\mathbf{\Omega}^e$ Based on $\mathbf{\Omega}$

In this subsection, we show how to efficiently find the parity check matrix $\mathbf{\Omega}^e$ with certain girth. First, the exhaustive search for $\mathbf{\Omega}^e$ is based on the rows of $\mathbf{\Omega}$. In the mean time, according to Observation 1, more short length bit-level cycles in $\mathbf{\Omega}$ could be avoided by enlarging q in many cases. Therefore, we could obtain $\mathbf{\Omega}^e$ with desired girth property more efficiently by changing the structure of $\mathbf{\Omega}$ instead of searching among numerous parity check combinations. That is, we first remove some rows in $\mathbf{\Omega}$ which contain bit-level cycles, then replace them with some new rows that will not introduce cycles with lengths smaller than certain number. The resulting $\mathbf{\Omega}^e$ could eliminate the bit-level cycles more efficiently and have a larger girth than $\mathbf{\Omega}$. The details are provided as follows.

Step 1) Let $q=2^p$, $p>1$. Given a parity check matrix $\bar{\mathbf{H}}$ with mother matrix $\mathbf{\Lambda}_p$. We construct its associated $\mathbf{\Omega}$. Let g_s be an even number.

Step 2) We construct a binary matrices set $\{\mathbf{B}'_1, \mathbf{B}'_2, \mathbf{B}'_3, \ldots\}$ with each $\mathbf{B}'_k$ being a cycle-free $2 \times (q-1)$ or $2 \times 2(q-1)$ matrix. In addition, $\mathbf{B}'_k \cdot \mathbf{v}_j = \mathbf{0}, \forall k, j$ or $(\mathbf{B}'_k(0,1), \ldots, \mathbf{B}'_k(0,q-1)) \cdot \mathbf{v}_j = \mathbf{0}$ and $(\mathbf{B}'_k(0,q), \ldots, \mathbf{B}'_k(0,2q-1)) \cdot \mathbf{v}_j = \mathbf{0}, \forall k, j$.

Step 3) In $\mathbf{\Omega}$, we find the matrix cycles with lengths smaller than g_s (that will result in bit-level cycles with lengths smaller than g_s) and set the rows across the associated matrix labels to be zero vectors. Then, we rearrange these zero rows to the lower part of the resulting matrix.

Step 4) For every two zero rows, we place a $\mathbf{B}'_k$ that will not cause bit-level cycles with lengths smaller than g_s within them (also at the non-overlapped column-positions, a detailed example is given in Fig. 2). The resulting matrix is denoted by $\mathbf{\Omega}^e$.

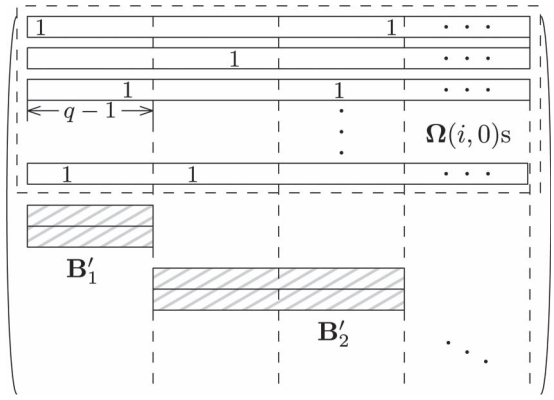

Fig. 2. The structure of a $\mathbf{\Omega}^e$. The upper part comprises some rows from $\mathbf{\Omega}$. The lower part comprises some matrices $\mathbf{B}_k$s.

Note that, given the practical LDPC code, the length-4 cycles in $\mathbf{\Lambda}_p$ are in general eliminated. Then, we only have to handle the matrix cycle with length $g_c > 4$ in Step 4. A benefit comes with the row replacing operation in Step 4 is that we could construct many $\mathbf{\Omega}^e$s whose degree distributions are more different from each other than the ones obtained without this operation.

IV. Bit-Level Decoder for the GBR

A. Motivation

Consider the performance-optimized $\bar{\mathcal{C}}$ under non-binary BP decoding. While the decoding performance could be very good, the computational complexity is high. In this section, our goal is to propose a low complexity bit-level decoding process for its associated GBR while the bit-level performance can closely approach the optimized symbol-level performance of $\bar{\mathcal{C}}$. To this end, the proposed decoding process for the associated GBR should have both good performance threshold and fast convergence speed (with regard to the number of decoding iterations).

We first notice that there exists the following isomorphism for $\bar{\mathcal{C}}$.

$$\bar{\mathcal{C}} \cong \mathcal{C}^e \cap \left(\mathcal{C}_1^e \times \mathcal{C}_2^e \times \cdots \times \mathcal{C}_N^e\right), \tag{3}$$

where $\mathcal{C}^e$ is the binary code defined with $\mathbf{\Omega}^e$, $\mathcal{C}_j^e$ is the binary code generated by $\mathbf{\Psi}_j$. The above equation implies that to have good performance threshold we may perform the binary BP decoding for the GBR and utilize the parity check relations for both $\mathcal{C}^e$ and $\mathcal{C}_1^e \times \mathcal{C}_2^e \times \cdots \times \mathcal{C}_N^e$. Then to further have fast convergence speed we introduce a hybrid parallel decoding process in Section IV-B, i.e., we allow the binary BP decoder and an extended hard decision decoder working iteratively to decode the GBR. Systematic investigation is also carried out to clearly explain how we achieve our goal and to provide more insights into the benefits of the proposed algorithms.

B. The Hybrid Parallel Decoding Process

Assume that $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_1^T, \bar{\mathbf{x}}_2^T, \ldots, \bar{\mathbf{x}}_N^T)^T$ is transmitted over the binary input channels. We denote $\bar{\mathbf{y}} = (\bar{\mathbf{y}}_1^T, \bar{\mathbf{y}}_2^T, \ldots, \bar{\mathbf{y}}_N^T)^T$ as

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

the received sequence. In the following, for ease of discussion, we refer to the bits in $\bar{\mathbf{x}}$ as *bit nodes*, the bits in $\mathbf{v}^e$ as *extended bit nodes* and the rows of $\mathbf{\Omega}^e$ as *constraint nodes*. Then, same as the definition of bipartite graph, the bit nodes are connected to the extended bit nodes according to the corresponding non-zero entries in $\mathbf{\Psi}_j^T$, $j = 1, 2, \ldots, N$ and the extended bit nodes are connected to the constraint nodes according to the corresponding non-zero entries in $\mathbf{\Omega}^e$. Next, we first give the extended hard decision decoder over binary symmetric channel (BSC). Then we show how to let the extended hard decision decoder and binary BP decoder work iteratively to decode the GBR over binary input Gaussian channel.

Extended Hard Decision Decoder (EHDD): Here, we present an extended iterative hard decision decoder for BSC. Let $\boxplus$ be the bit-wise addition of the vector space over $\mathbb{F}_2$. Then, for a simplex code $\mathbf{v}_j$ [25], we have $\mathbf{v}_j(j_1' \boxplus j_2' \boxplus \cdots \boxplus j_k') = \mathbf{v}_j(j_1') + \mathbf{v}_j(j_2') + \cdots + \mathbf{v}_j(j_k'), j_i' \in \{1, 2, \ldots, q-1\}$ [14]. By utilizing this property and the $\mathbf{\Omega}^e$, we present the iterative decoding procedure below.

Step 1) We denote $\hat{\mathbf{v}}^e$ as the message for the extended bit nodes which is initialized by the value of $\mathbf{\Psi}_j^T \bar{\mathbf{y}}_j$, $j = 1, 2, \ldots, N$ and b as the thresholds to perform the bit-flippings.

Step 2) If $\mathbf{z} = \mathbf{\Omega}^e \hat{\mathbf{v}}^e = \mathbf{0}$ then $\mathbf{v}^e = \hat{\mathbf{v}}^e$. Else, $\mathbf{s} = \mathbf{z}^T \mathbf{\Omega}^e = (\mathbf{S}_j)_{1 \times N}$ (here is the decimal multiplication). For $j' \in \{1, 2, \ldots, q-1\}$, if $\mathbf{s}_j(j') \geqslant b$ and $\mathbf{v}_j^e(j') \neq \mathbf{v}_j^e(j_1') + \mathbf{v}_j^e(j_2') + \cdots + \mathbf{v}_j^e(j_k')$ where $j_i' \in \{1, 2, \ldots, q-1\}$ such that $\mathbf{\Psi}_j(0, j_i') \neq \mathbf{0}$ and $j' = j_1' \boxplus j_2' \boxplus \cdots \boxplus j_k'$, then $\hat{\mathbf{v}}_j^e(j') = 1 + \hat{\mathbf{v}}_j^e(j')$.

Step 3) Stop the procedure when $\mathbf{\Omega}^e \hat{\mathbf{v}}^e = \mathbf{0}$ or the maximum number of iterations is reached. For the trivial case, $\bar{\mathbf{x}}_j = (\mathbf{v}_j^e(1), \mathbf{v}_j^e(2), \ldots, \mathbf{v}_j^e(2^{p-1}))^T$.

For ease of presentation, we denote b as the thresholds for extended bit nodes with different degrees at different iterations, i.e., for an extended bit node with degree-d at iteration-l, set $b > \lfloor d/2 \rfloor$. We also introduce the simplex parity checks to guarantee enhanced decoding performance. Below, we show how to apply the BP algorithm into the decoding of the GBR over binary input Gaussian channel.

Hybrid Parallel Decoder (HPD): The hybrid parallel decoder (HPD) for the GBR consists of two component decoders, i.e., the binary BP decoder and the extended hard decision decoder (EHDD). The BP decoder and the EHDD exchange decoding messages iteratively. We consider one decoding round is finished iff these two decoders have exchanged information once. A (μ, ν) decoding round is a decoding round within which the BP decoder has performed μ times consecutive decoding iterations and the EHDD has performed ν times consecutive decoding iterations. Different from the BSC, we choose to transmit $\mathbf{v}^e$ instead of $\bar{\mathbf{x}}$. Assume BPSK is utilized. We denote $\mathbf{y}^e$ as the received sequence. Then the decoding process is described below.

Step 1) Initialize the message for the vth extended bit node by $\mu_{v,c}^{(0)} = (2/\sigma^2)\mathbf{y}^e(v)$ and the message for the cth constraint node by $\omega_{c,v}^{(0)} = 0$.

Step 2) $\omega_{c,v}^{(l)} = -2\tanh^{-1}(\prod_{i'' \in \mathcal{N}_c \backslash \{v\}} \tanh(-\mu_{i'',c}^{(l-1)}/2))$, where $\mathcal{N}_c$ is set of the extended bit nodes connected to the cth constraint node.

Step 3) $\mu_{v,c}^{(l)} = (2/\sigma^2)\mathbf{y}^e(v) + \sum_{j'' \in \mathcal{M}_v \backslash \{c\}} \omega_{j'',v}^{(l)}$, where $\mathcal{M}_v$ is the set of constraint nodes connected to the vth extended bit node.

Step 4) For iteration-μ in a (μ, ν) decoding round, let the hard decision be $\hat{\mathbf{v}}^e$. We apply the EHDD for the following ν times decoding iterations. If $\hat{\mathbf{v}}^e(v) = 1$, $\mu_{v,c}^{(l)} = -|\mu_{v,c}^{(l)}|$, else $\mu_{v,c}^{(l)} = |\mu_{v,c}^{(l)}|$. Then, go to step 2.

Step 5) Stop the procedure when $\mathbf{\Omega}^e \hat{\mathbf{v}}^e = \mathbf{0}$ or the maximum number of iterations is reached. For the trivial case, $\bar{\mathbf{x}}_j = (\mathbf{v}_j^e(1), \mathbf{v}_j^e(2), \ldots, \mathbf{v}_j^e(2^{p-1}))^T$.

We denote $\mathscr{S}_v$ as the set containing all the bit nodes connected to the vth extended bit node. Then $\mathbf{v}^e(v) + \sum_{i' \in \mathscr{S}_v} \bar{\mathbf{x}}(i') = 0$. As a result, if $\bar{\mathbf{x}}$ is transmitted over the binary input Gaussian channel, the initialization of the messages for the extended bit nodes can be performed similarly to the processing rule in **Step 2**. The decoding procedure is the same. Note that when $\mu = 0$, the HPD coincides with the extended hard decision decoder. When $\nu = 0$, the hybrid parallel decoder coincides with the binary BP decoder.

Performance evaluation of the GBR under HPD could be done by utilizing the Monte-Carlo experiments for an "infinite" LDPC code used in [2], [15]. That is, by decoding a simulated "infinite" long code from its associated ensemble, we evaluate the performance in terms of the minimum signal to noise ratio (MSNR), i.e., T_b, for which the average syndrome bit entropy (ASBE) reaches certain value after a number of decoding iterations. Note that, for codes with particular edge connections, e.g., the protograph-based codes whose definition permits the introduction of degree-1 nodes, punctured nodes in the protograph and protograph chains, their decoding performance will be very different from the average performance of the random codes ensemble with the same degree distributions. However, like the codes (with some structures) used in [2], the GBR does not require these particular edge connections. Decoding performance of the GBR could be evaluated in terms of the average performance of its associated random code ensemble. Advantages of this method are twofold. First, it can provide good approximation to the real decoding behavior with regard to both performance limit and decoding iterations [2]. Second, it can easily incorporate different channel models. For the simulation results, we refer the reader to Section V-D.

Moreover, the hybrid parallel decoding process computes the decoding messages at bit-level. Then, by removing the zero columns in each $\mathbf{\Psi}_j$ and $\mathbf{\Omega}^e$, the computational complexity of the check-vector-sum operation for $\mathbf{\Omega}$ relies linearly on the number of the non-zero columns in $\mathbf{\Omega}_i^e$, $i = 1, 2, \ldots, M$. The computational complexity for the simplex parity checks relies linearly on the non-zero columns in $\mathbf{\Psi}_j$, $j = 1, 2, \ldots, N$. We denote the maximum number of the non-zero columns in each $\mathbf{\Omega}_i^e$ by $\phi_e \leqslant q-1$ and the maximum number of the non-zero columns in each $\mathbf{\Psi}_j$ by $\psi_e \leqslant q-1$. Then the computational complexity is dominated by $O(m_s = \max\{\phi_e, \psi_e\})$.

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

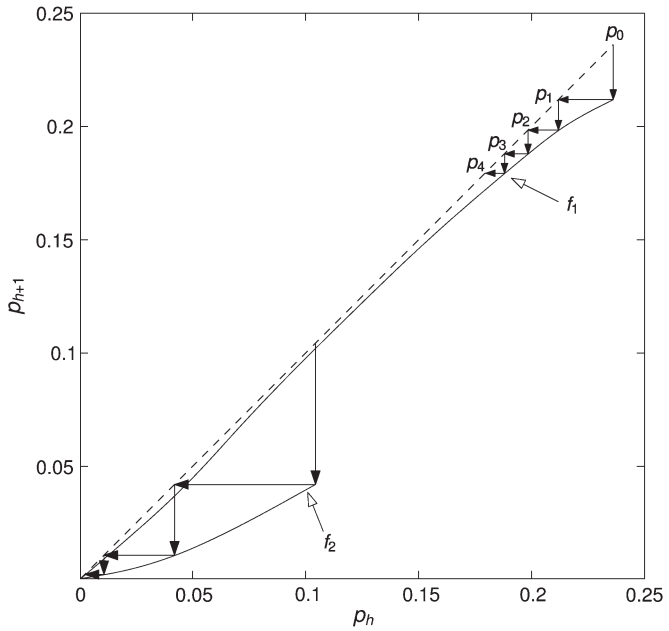

Fig. 3. Consider a performance optimized 8-ary LDPC code of rate 0.265. f_1 is the EXIT chart for its optimized GBR under the binary BP decoder at $E_b/N_0 = -0.1$ dB. f_2 is the EXIT chart for the binary BP decoder at $E_b/N_0 = 4.7$ dB. $p^*_{0,BP} = 0.244$.

C. Bit-Level Decoding Under Different (μ, ν)s

In this subsection we explain how to choose (μ, ν) so that the HPD will converge faster and have lower MSNR compared to its component decoders. Note that the MSNR is obtained by simulating an "infinite" code when the average syndrome bit entropy (ASBE) reaches certain value after a number of decoding iterations. If the ASBE is set to be very small value and the number of decoding iterations is set to be very large number, we refer to the obtained MSNR as the *asymptotic performance threshold*. If the ASBE is set to be small value and the number of decoding iterations is set to be not very large number, the obtained MSNR is an equivalent measure for the convergence speed. In this case, we refer to the MSNR as the *convergence threshold*.

First, we associate the asymptotic performance threshold with a message error probability p^*_0 which is the error rate for the sequence received from the corresponding channel. Then, we adopt the EXIT (extrinsic information transfer) chart based on the message error probability to perform the analysis. This method begins with defining the message error probability function $p_{h+1} = f(p_h, p_0)$ for an iterative decoder, where p_{h+1} is the extrinsic message error probability (EMEP) at the output of the iteration-h, p_h is the extrinsic message error probability (EMEP) at the input of the iteration-h, p_0 is the intrinsic message error probability (IMEP, the message error probability for the sequence received from channel). Then the EXIT chart for a fixed p_0 is obtained by plotting f and $p_{h+1} = p_h$ both in a graph (as shown in Fig. 3, f is obtained by the Monte-Carlo experiments). The decoding steps/iterations are visualized as the arrows starting from p_0 in Fig. 3. For monotonic decoder, the decoding tunnel will be more open as p_0 increases. The decoding tunnel is closed iff $f(p_h, p_0) \geqslant p_h$. Then, p^*_0 is the worst intrinsic message error rate for which the decoding tunnel is open.

In the following, we refer to the binary BP decoder in the HPD as the component BP decoder to avoid confusion. We assume that the GBR for the performance-optimized $\bar{\mathcal{C}}$ is decoded by the binary BP decoder and the HPD, respectively. We denote the EMEP for the binary BP decoder at the output of the iteration-h as $p_{h,BP}, h \in \mathbb{N}$. $p_{0,BP}$ is IMEP for the binary BP decoder. We denote the EMEP for the HPD at the output of the iteration-h as $p_{h,HPD}, h \in \mathbb{N}$. $p_{0,HPD}$ is IMEP for the HPD. Then the IMEP corresponding to the asymptotic performance threshold for the binary BP decoder is denoted by $p^*_{0,BP}$. The IMEP corresponding to the asymptotic performance threshold for the HPD is denoted by $p^*_{0,HPD}$.

Next, we consider the case when $p_{0,HPD}$ and $p_{0,BP}$ are the same and close to $p^*_{0,BP}$. As a result, the decoding tunnel for the binary BP decoder under performance-optimized GBR is very narrow, as shown in Fig. 3. However, the beginning part of the tunnel is wider than most of the other parts, which means that the first a few decoding iterations will make the message error probability fall quicker than most of the other decoding iterations. For the HPD with a fixed (μ, ν), the component BP decoder does the first μ times decoding iterations in the kth, $k \in \mathbb{N}^*$ decoding round. The IMEP for the component BP decoder (in the kth decoding round) is $p_{0,HPD} = p_{0,BP}$, since the component BP decoder always uses the same channel inputs in each iteration. Then the EHDD does the following ν times decoding iterations over the BSC with IMEP being equal to $p_{(k-1)(\mu+\nu)+\mu,HPD}$. This means that the EXIT chart for the component BP decoder in the kth decoding round is the same as the one for the component BP decoder in the $(k+1)$th decoding round. In addition, the EXIT charts for the EHDD in different decoding rounds are different since the IMEPs for the EHDD in different decoding rounds are different. Further, in each decoding round, the ν times decoding iterations over the BSC will always start from the beginning point of its associated EXIT chart.

In general, we assume that the decoding tunnel for the EHDD within the first decoding round is open at the beginning part. This assumption is reasonable because we allow the component BP decoder to do the decoding first. Then, we could have $p_{\mu+\nu,HPD} \leqslant p_{\mu+\nu+\Delta_1,BP} < p_{\mu+\nu,BP}, \Delta_1 \in \mathbb{N}^*$. To have a better understanding, we give an example in Fig. 4 where the decoding iterations in the first decoding round are visualized. In this example, we choose a (μ, ν), i.e., $\mu = 7$ and $\nu = 2$, such that $p_{\mu+\nu,HPD} < p_{\mu+\nu+5,BP}$. The decoding tunnel for the component BP decoder in the second decoding round is plotted in Fig. 5. It can be seen that $p_{\mu+\nu+1,HPD} < p_{\mu,BP} = p_{\mu,HPD}$, which means that the $(\mu+1)$th decoding iteration in the component BP decoder (in the second decoding round) achieves a lower message error probability compared to its μth decoding iteration (in the first decoding round). However, $p_{\mu+\nu,HPD} < p_{\mu+\nu+1,HPD}$, i.e., the HPD is in general not a monotonic decoder. This is mainly due to the fact that some LLRs are changed to their additive inverses while their magnitudes remaining the same (at the end of the first decoding round) and the magnitudes of some of these LLRs are small. Then, some more errors may be caused by the channel inputs. It is also the reason why the decoding tunnel for the component BP decoder from $p_{\mu+\nu+1,HPD}$ to $p_{\mu+\nu+2,HPD}$ is slightly

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

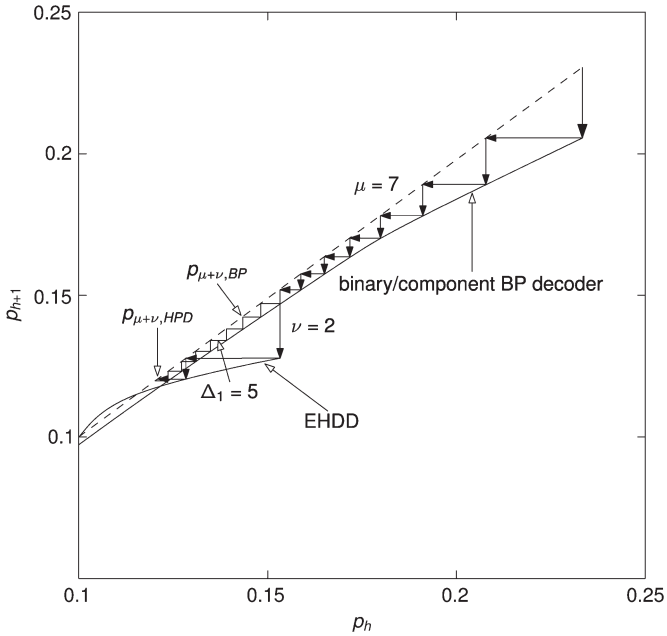

Fig. 4. The first decoding round of the HPD at $E_b/N_0 = 0.1$ dB for the code in Fig. 3.

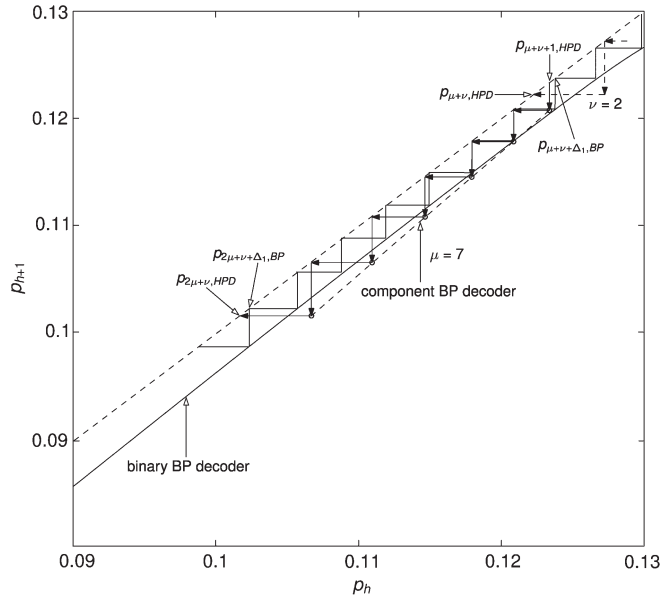

Fig. 5. The EXIT chart for the component BP decoder in the second decoding round at $E_b/N_0 = 0.1$ dB for the code in Fig. 3.

tighter than the corresponding tunnel for the binary BP decoder. In the meantime, we observe that the values of the LLRs contribute to incorrect decodings in the component BP decoder are also smaller than that in the binary BP decoder, which makes the decoding tunnel for component BP decoder from $p_{\mu+\nu+3,HPD}$ to $p_{2\mu+\nu,HPD}$ wider than the corresponding tunnel for the binary BP decoder. As k increases, the decoding tunnel for the EHDD will be more open. Then, we further expect that $p_{\mu+\nu,HPD} \leqslant p_{\mu+\nu+\Delta_k,BP} < p_{\mu+\nu,BP}, \Delta_k \in \mathbb{N}^*$ with $\Delta_{k-1} < \Delta_k$, i,e., the HPD is monotonic with respect to k. Then, determining the best values of μ and ν amounts to maximizing Δ_k for a fixed number of decoding iterations. In our simulations, with properly chosen (μ, ν), $p^*_{0,HPD}$ could also be very close to $p^*_{0,BP}$.

TABLE I
MSNRs FOR DIFFERENT (μ, ν)s. p_μ IS THE PERCENTAGE OF THE NUMBER OF DECODING ITERATIONS PERFORMED BY THE BINARY BP DECODER WITHIN A (μ, ν) DECODING ROUND

T_b (dB)	0.75	0.62	0.67	1.57	1.03	2.11
(μ, ν)	$(1, 0)$	$(16, 4)$	$(17, 3)$	$(1, 3)$	$(5, 15)$	$(0, 1)$
p_μ	100%	80%	85%	25%	25%	0%

When $p_{0,HPD}$ and $p_{0,BP}$ is not close to $p^*_{0,BP}$. The decoding tunnel for the binary BP decoder will become wider. However, when $p_{h,BP}$ is small, the convergence speed of the binary BP decoder will also become slow. In the meantime, considering the HPD, the decoding tunnel for the EHDD will become wider as the decoding proceeds. Then, we expect that the HPD could also have faster converge speed than the binary BP decoder does in small message error probability region.

To provide more insights, we define $p_\mu = (100 \times \mu)/(\mu + \nu)\%$. Then, when $p_\mu = 0\%$, the HPD coincides with the EHDD. When $p_\mu = 100\%$, the HPD coincides with the binary BP decoder. We consider a rate $R = 0.5311$ irregular non-binary LDPC code over $\mathbb{F}_8$. Among the constructed $\mathbf{\Omega}^e$s, we choose the one with the smallest p^*_{BP}. If the maximum number of decoding iterations is set to be 60, Table I gives the converge thresholds (MSNRs) for different (μ, ν)s. It can be seen that, to obtain low MSNRs, the binary BP decoder should do most of the decoding iterations. Moreover, with carefully chosen (μ, ν), the HPD could have lower MSNR than the binary BP decoder does. It is worth mentioning that, with the simplex constraints, the EHDD could have better asymptotic performance threshold than that without the constraint. Then, the decoding tunnel for the EHDD will be open at higher EMEP, i.e., μ could be assigned with a smaller value when $p_{0,HPD}$ is close to $p^*_{0,BP}$. As a result, the HPD is expected to have better MSNR than that without the simplex constraints. In the following, for ease of discussion, we refer to the MSNR for a $\mathbf{\Omega}^e$ as the lowest MSNR corresponding the best choice of (μ, ν) among a range of values under a fixed maximum number of decoding iterations.

D. Bit-Level Decoding Under Different $\mathbf{\Psi}$s

When the decoding of $\mathbf{v}^e$ over $\mathbf{\Omega}^e$ is accomplished, we have to get every $\bar{\mathbf{x}}_j$ from $\mathbf{v}^e_j$. To guarantee $\bar{\mathbf{x}}_j$ being successfully recovered from $\mathbf{v}^e_j$, we first provide the following conditions for the extended generator matrices.

Theorem 6: Consider the GBR with extended generator matrices set $\mathbf{\Psi} = \{\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_N\}$. For all $j \in \{1, 2, \ldots, N\}$ and $q = 2^p \geqslant 4$,

1) if $\mathrm{wt}(\mathbf{\Psi}_j) > (q/2) - 1$, every bit in $\bar{\mathbf{x}}_j$ can be recovered from $\mathbf{v}^e_j$.
2) If $\mathrm{wt}(\mathbf{\Psi}_j) = (q/2) - 1$, $\bar{\mathbf{x}}_j$ can be recovered with probability of $1 - ((q-1)/\binom{q-1}{(q/2)-1})$.

Proof: Recall that $\mathbf{\Phi}$ is a $p \times (q-1)$ matrix and

$$V = \{\mathbf{0}, \mathbf{\Phi}(0,1), \mathbf{\Phi}(0,2), \ldots, \mathbf{\Phi}(0, q-1)\}$$

is a vector space of dimension-p. We denote

$$V^e_j = \{\mathbf{0}, \mathbf{\Psi}_j(0,1), \mathbf{\Psi}_j(0,2), \ldots, \mathbf{\Psi}_j(0, q-1)\}$$

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

as the set formed by the column vectors of $\mathbf{\Psi}_j$. Then

$$\mathrm{wt}(\mathbf{\Psi}_j) = \left|V_j^e\right| - 1.$$

We denote

$$V' = \left\{\mathbf{\Phi}(0,1), \mathbf{\Phi}(0,2), \ldots, \mathbf{\Phi}(0,2^{p-1})\right\}$$

as the set of all unit vectors. Then the non-zero vectors in V and V_j^e can be formulated by the additions of the vectors in V'.

If $|V_j^e|$ is larger than the size of the $(p-1)$-dimensional subspace of V, then $\mathrm{rank}(\mathbf{\Psi}_j) = p$. Every bits in $\bar{\mathbf{x}}_j$ can be recovered. The size of the $(p-1)$-dimensional subspace can be calculated by $\sum_{i=1}^{p-1}\binom{p-1}{i} + 1 = 2^{p-1}$. Then if $\mathrm{wt}(\mathbf{\Psi}_j) > \sum_{i=1}^{p-1}\binom{p-1}{i} = 2^{p-1} - 1$, $\bar{\mathbf{x}}_j$ can be recovered from $\mathbf{v}_j^e$. If $\mathrm{wt}(\mathbf{\Psi}_j) = \sum_{i=1}^{p-1}\binom{p-1}{i}$, the rank of $\mathbf{\Psi}_j$ is either p or $p-1$. Then the probability that $\bar{\mathbf{x}}_j$ can be recovered equals the probability that the non-zero vectors in $\mathbf{\Psi}_j$ do not form a $(p-1)$-dimensional subspace, which depends on the number of the $(p-1)$-dimensional subspaces. To calculate the number of the $(p-1)$-dimensional subspaces of the V, we first introduce the Gaussian binomial coefficient over finite field $\mathbb{F}_q$ [25]

$$\binom{n}{k}_q = \frac{[n]_q!}{[k]_q![n-k]_q!},\ k \leqslant n,$$

where $[n]_q! = [1]_q[2]_q \ldots [n]_q$ with

$$[m]_q = \frac{1-q^m}{1-q} = \sum_{0\leqslant i<m} q^i = 1 + q + q^2 + \cdots q^{m-1}, 1 \leqslant m \leqslant n.$$

Then the number of the $(p-1)$-dimensional subspaces over $\mathbb{F}_2$ is calculated by $\binom{p}{p-1}_2 = \sum_{i=1}^{p} 2^{i-1}$. The probability that $\bar{\mathbf{x}}_j$ can be recovered when $\mathrm{wt}(\mathbf{\Psi}_j) = \sum_{i=1}^{p-1}\binom{p-1}{i}$ is

$$1 - \binom{p}{p-1}_2 \Big/ \binom{q-1}{\mathrm{wt}(\mathbf{\Psi}_j)}.$$

■

Note that $2^{p-1} = \sum_{i=1}^{p-1}\binom{p-1}{i} + 1 \geqslant p, \forall p \geqslant 2$. In addition, if $\bar{\mathbf{x}}_j$ can be recovered, $\mathrm{wt}(\mathbf{\Psi}_j)$ is at least the size of a basis of a dimension- p vector space over $\mathbb{F}_2$, i.e., $\mathrm{wt}(\mathbf{\Psi}_j) \geqslant \log_2 q$, $j = 1, 2, \ldots, N$. Thus, the least number of non-zero columns required for each $\mathbf{\Psi}_j$ is p, which serves as a necessary condition for the successful decoding of $\bar{\mathbf{x}}$.

When $\mathrm{wt}(\mathbf{\Psi}_j)$ is large enough, $\bar{\mathbf{x}}_j$ could be recovered from $\mathbf{v}_j^e$ with certainty. In addition, as $\mathrm{wt}(\mathbf{\Psi}_j)$ increases, more extended bits are introduced, which will gives birth to flexible rates for the GBR. We denote the code rate of $\bar{\mathcal{C}}$ by R. The length of $\mathbf{v}^e$ is $M_s = \sum_j \mathrm{wt}(\mathbf{\Psi}_j)$. Then, we define the *extended rate* for the GBR as $R_e = NpR/M_s$. Since in general $Np \leqslant M_s \leqslant N(q-1)$, $(pR/(q-1)) \leqslant R_e \leqslant R$. That is, the larger M_s the smaller R_e (for the same $\bar{\mathcal{C}}$). Then, the proposed decoding procedure under different $\mathbf{\Psi}$ (or R_e) is capable of dealing with different channel conditions. To explain this, we consider the HPD and the non-trivial GBRs for a 16-ary LDPC code with $R = 1/2$. Assume that the decoder only try to recover $\bar{\mathbf{x}}_j$ when $\bar{\mathbf{v}}_j^e$ is successfully decoded. We denote the proportion of unrecovered $\bar{\mathbf{x}}_j$s by p_u, the bit error rate for $\mathbf{v}^e$ under the HPD by p_v and the number of decoding iterations by l. As shown in Table II, for different R_e under the signal-to-noise rate (SNR) value of interest, p_v will largely decrease as the decoding procedure proceeds.

TABLE II
HYBRID PARALLEL DECODER UNDER DIFFERENT $\mathbf{\Psi}$ S. M_s IS THE LENGTH OF $\mathbf{v}^e$, m_s IS THE MAXIMUM NUMBER OF NON-ZERO COLUMNS IN EACH $\mathbf{\Omega}_i^e$ AND EACH $\mathbf{\Psi}_j$, p_v IS THE BIT ERROR RATE FOR $\mathbf{v}^e$, p_u IS THE PERCENTAGE OF UNRECOVERED $\bar{\mathbf{x}}_j$, l IS THE NUMBER OF DECODING ITERATIONS, E_b/N_0 IS THE CHANNEL SNR AND T_b IS THE MSNR FOR $\mathbf{v}^e$

R_e	0.15		0.36	
E_b/N_0	0.9dB		1.8dB	
l	p_v	p_u	p_v	p_u
20	5.09e-02	9.5e-01	1.19e-05	2.03e-02
40	6.87e-03	1.05e-01	0	2.03e-02
60	4.68e-04	5.20e-03	0	2.03e-02
80	9.43e-05	1.37e-04	0	2.03e-02
T_b for $\mathbf{v}^e$	0.27dB		0.77dB	
m_s	15		6	

On the other hand, when m_s is not large enough, p_u will not converge to arbitrary small value as the number of decoding iterations grows. A certain proportion of $\bar{\mathbf{x}}_j$s will remain unrecovered no matter how many decoding iterations are performed. It is suggested that the proposed decoding procedure for the non-trivial GBR requires lower R_e than the decoding of the trivial GBR does in general. As a result, according to Theorem 6, we have the sufficient extended rate condition for the proposed decoding of GBR as

$$\frac{pR}{q-1} \leqslant R_e \leqslant \frac{2pR}{q}.$$

E. Bit-Level Decoding Compared to the Symbol-Level Decoding of $\bar{\mathcal{C}}$

According to Section IV-D, to successfully recover each $\bar{\mathbf{x}}_j$, $\mathrm{wt}(\mathbf{\Psi}_j)$ should be large enough. Then the proposed decoding under different $\mathbf{\Psi}$s could deal with different channel conditions. Another benefit come with large M_s is that, by decoding the GBR for a performance-optimized $\bar{\mathcal{C}}$, the low complexity bit-level decoding could closely approach the symbol level decoding for the optimized $\bar{\mathcal{C}}$. More specifically, considering a practical code $\bar{\mathcal{C}}$ with optimized degree distributions, its performance under the non-binary BP decoder could be very good. In the mean time, the computational complexity is also high. On the other hand, if m_s is not limited to small values, we could have many $\mathbf{\Omega}^e$s with large M_ss by the practical construction introduced in Section III-F. Among these matrices, we choose the one with the lowest MSNR and large girth. Then, by decoding the corresponding GBR with the HPD, the low complexity bit-level decoding could perform similarly to the symbol-level decoding of the optimized $\bar{\mathcal{C}}$.

To explain this, we consider a performance-optimized irregular 8-ary LDPC code with $R = 0.5311$ which is decoded by the QSPA. Its associated GBRs are decoded by the HPD. As shown in Table III, the GBR could have lower MSNR than the EBR does. In the mean time, we could also establish a

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

TABLE III
THE BIT-LEVEL DECODING COMPARED TO THE SYMBOL-LEVEL DECODING. m_s IS THE MAXIMUM NUMBER OF NON-ZERO COLUMNS IN EACH $\mathbf{\Omega}_i^e$ AND EACH $\mathbf{\Psi}_j$, T_b IS THE MSNR IN dB AND g_s IS THE GIRTH

T_b	0.59	0.62	0.71	0.73	1.31	0.67	2.11	3.2
R_e	$\mathbf{H}$	0.228	0.228	0.31	0.37	0.228 ($\mathbf{\Omega}$)	0.228 (EHDD)	0.31 (EHDD)
m_s	$\times$	7	7	6	5	7	7	6
g_s	8	12	12	12	12	8	12	12

GBR with MSNR close to that of its optimized mother code $\bar{\mathcal{C}}$. Then, the low complexity bit-level decoding for the GBR could perform closely to the symbol-level decoding of $\mathcal{C}$. For the detailed simulation results, we refer the reader to Section V-A.

F. Code Optimization for the GBR

It has been shown that, by increasing each wt($\mathbf{\Psi}_j$) and finding the $\mathbf{\Omega}^e$ with lowest MSNR and large girth, the HPD could achieve enhanced decoding performance. Moreover, for a fixed q, the computational complexity will also increase as m_s grows. In this subsection, we provide a simple algorithm to optimize the GBR (to make each wt($\mathbf{\Psi}_j$) large enough and optimize the girth and degree distributions of $\mathbf{\Omega}^e$) while allowing a trade-off between the decoding performance and computational complexity for a fixed q. First, we assume that the mother matrix $\mathbf{\Lambda}_p$ is constructed by the modified progressive-edge-growth (PEG) algorithm. One also can construct $\mathbf{\Lambda}_p$ by other random methods or with some structures. Let the matrix labels for the corresponding $\bar{\mathbf{H}}$ be chosen according to Corollary 5.[1] The rate for $\mathbf{\Lambda}_p$ of size $M \times N$ is $R = 1 - M/N$. Then, we assume that the $\mathbf{\Omega}$s for different values of q can be constructed by either fixing the $M_s = N(q-1)$ and changing $\mathbf{\Lambda}_p$s or fixing the $\mathbf{\Lambda}_p$ and changing the M_ss. Below, we provide the details.

Step 1) Let $q = 2^p$, $p > 1$. Given the mother matrix $\mathbf{\Lambda}_p$, we construct the equivalent binary parity check matrix $\bar{\mathbf{H}}$ by filling $\mathbf{\Lambda}_p$ with the optimized matrix labels of size $p \times p$ according to Corollary 5. Then we construct the $\mathbf{\Omega}$ based on $\bar{\mathbf{H}}$.

Step 2) Let g_{s_1} be an even number. We find the matrix cycles in $\mathbf{\Omega}$ with lengths smaller than g_{s_1} (that will result in bit-level cycles with lengths smaller than g_{s_1}) and set the rows across the associated matrix labels to be zero vectors. Then, same as the method in Section III-F, we construct many $\mathbf{\Omega}^e$s by filling these zero rows with matrices $\mathbf{B}_k$s (without checking the girth when placing a $\mathbf{B}_k$ in the zero rows).

Step 3) Let $c > (q/2) - 1$ be a non-zero integer. Among the matrices constructed in **Step 2**, we find the ones with $c \geqslant m_s > (q/2) - 1$.

Step 4) Let $g_{s_2} \leqslant g_{s_1}$ be an even number. Let t be a real number. We search among the matrices constructed in **Step 3** for the one with smallest MSNR (also not exceeding t) and girth not smaller than g_{s_2}. The resulting matrix is denoted by $\mathbf{\Omega}^e$. If such matrix can not be found, $p = p + 1$ and go to **Step 1**.

Note that, for short block length codes, we drop the MSNR examinations in **Step 4** and only choose a matrix in **Step 3** with suitable m_s and large girth as the resulting $\mathbf{\Omega}^e$. If c is set to be $q-1$ and q is fixed, the algorithm produces a $\mathbf{\Omega}^e$ with the lowest MSNR for a given $\mathbf{\Lambda}_p$. As shown in Section IV-E, in this case we expect that the bit-level performance could closely approach the optimized symbol-level performance. If $(q/2) - 1 < c < q - 1$ and q is fixed, the resulting $\mathbf{\Omega}^e$ may have a higher MSNR while the decoding complexity is lower. By allowing p to increase, the above steps could also be utilized to design binary codes with different lengths and girths while permitting the MSNR to be optimized.

[1]For the mother matrix $\mathbf{\Lambda}_p$, how to optimize the degree distributions has been studied in [5], [7]. The optimization of the matrix labels has been studied in [2], [12], [26]. The authors in [2], [12], [26] propose several optimization methods based on the equivalent binary LDPC codes. The degree distributions for the resulting $\bar{\mathbf{H}}$ can be efficiently calculated according to [20].

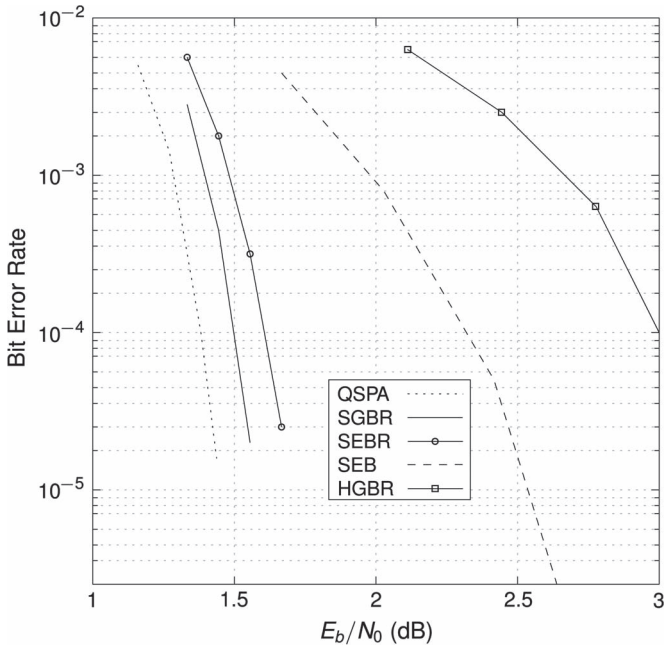

Fig. 6. Performance comparison between different representations for the non-binary LDPC code over $\mathbb{F}_8$ of rate $R = 0.5311$. The block length is 12000 bits, maximum 40 iterations, $\mu = 16$ and $\nu = 4$.

V. SIMULATION

A. Different Binary Forms of a Non-Binary LDPC Code

We present the simulation results for different representations of a non-binary LDPC code under different decoders. No undetectable error is observed in our simulations. We denote $M_s = \sum_j \text{wt}(\mathbf{\Psi}_j)$ as the length of $\mathbf{v}^e$. Consider the code over $\mathbb{F}_8$ of rate $R = 0.5311$. The block length 12000 bits. Degree distributions and MSNRs for $\mathbf{H}$ and $\mathbf{\Omega}^e$ are displayed in Table V. In addition, $\mathbf{v}^e = \mathbf{v}$ and $\mathbf{\Omega}_i^e \neq \mathbf{\Omega}_i$ for some i, i.e., M_ss for $\mathbf{\Omega}$ and $\mathbf{\Omega}^e$ are the same. The girth of $\mathbf{\Omega}$ is 8 and the girth of $\mathbf{\Omega}^e$ is 12. The MSNR for $\mathbf{\Omega}^e$ is $E_b/N_0 = 0.62$ dB. The MSNR for $\mathbf{\Omega}$ is $E_b/N_0 = 0.67$ dB, while the capacity limit is $E_b/N_0 = 0.30$ dB. We consider the binary input Gaussian channel. Then, the comparison is shown in Fig. 6, where HGBR (hard decision decoder for the GBR) is the extended hard decision decoder for $\mathbf{\Omega}^e$, SGBR (soft decision decoder for the GBR) is the hybrid parallel decoder for $\mathbf{\Omega}^e$, QSPA is the

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

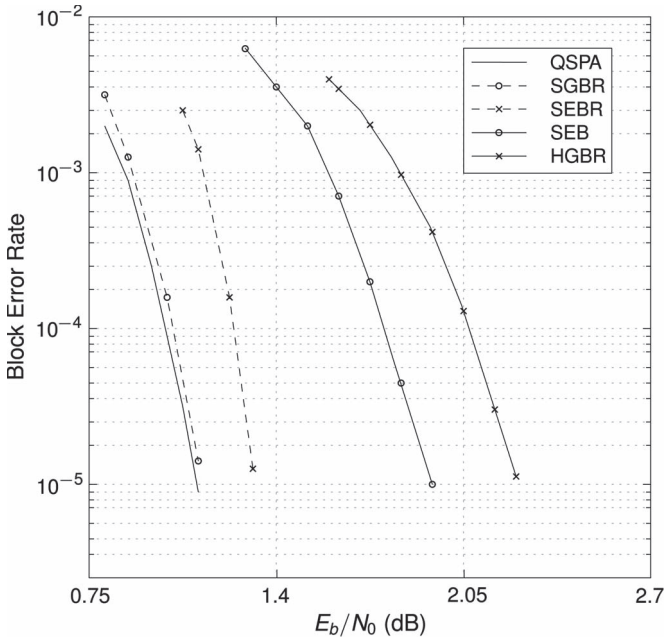

Fig. 7. Performance comparison between different representations for the non-binary LDPC code of rate half over $\mathbb{F}_{16}$. The block length is 2048 bits, maximum 200 iterations, $\mu = 16$ and $\nu = 4$.

q-ary sum-product decoder for $\mathbf{H}$, SEB (soft decision decoder for the equivalent binary LDPC code) is the binary BP decoder for $\bar{\mathbf{H}}$ and SEBR (soft decision decoder for the extended binary representation) is the hybrid parallel decoder for $\mathbf{\Omega}$. QSPA is used as the benchmark for both performance and complexity. Due to the short length bit-level cycles in $\bar{\mathbf{H}}$, SEB suffers from a performance loss of about 1 dB. In our simulation, the performance gap between SGBR and QSPA is within 0.2 dB while the computational complexity of SGBR is much lower.

Consider the non-binary LDPC code of rate half over $\mathbf{F}_{16}$ characterized by $\lambda(x) = 0.303x + 0.337x^2 + 0.04x^3 + 0.113x^4 + 0.122x^6 + 0.085x^{12}$ and $\rho(x) = 0.85x^5 + 0.15x^6$. The associated GBR of this code is optimized by the algorithm in Section IV-F. The block length is 2048 bits. We give the performance comparison between different representations in Fig. 7. In this example, the decoding performance of the GBR is very similar to that of the non-binary code.

B. $\mathbf{\Omega}$s and $\mathbf{\Omega}^e$s With Different Girths

In this subsection, based on the optimization in Section IV-F, we give comparative results for $\mathbf{\Omega}^e$s and $\mathbf{\Omega}$s with different girths and M_ss ($M_s = \sum_j \mathrm{wt}(\mathbf{\Psi}_j)$) which are displayed in Table IV. Consider the (3,6)-regular non-binary LDPC code over $\mathbb{F}_q$ with 120 coded symbols. We denote g_s as the girth of $\mathbf{\Omega}^e$ and assume the hybrid parallel decoder is adopted. For different p, we give the performance comparison in Fig. 8. The GBR with $M_s = 3321$ performs the best due to the optimization on the girth and large M_s.

C. Comparison of Codes From Literature

Consider the non-binary LDPC code of rate-half over $\mathbf{F}_{16}$ in Section V-A. We compare the performance of its GBR with the performance optimized non-binary cycle LDPC codes (optimized under similar assumptions) and the girth optimized binary LDPC codes in the literature. In Fig. 9, SPB59 is the sphere packing bound for block length-2048 bits. The codes from [11] is the non-binary cycle code with length 5376 bits. The code from [12] is the non-binary cycle code with length 2048 bits. The code from [18] is the non-binary cycle code with length 3000 bits. These codes are decoded by the FFT-QSPA. The code from [9] is the (3,6) QC-LDPC code with length 2294 bits. The code from [10] is the PEG-LDPC code with length 2694 bits. These codes are decoded by the binary BP decoder. The GBR under HPD for the $\mathbb{F}_{16}$ code has achieved a maximum 0.8 dB (at BER $= 10^{-4}$) performance gain compared to the optimized non-binary cycle LDPC codes with lower computational complexity.

TABLE IV
DIFFERENT OUTPUTS FROM SECTION IV-F. q IS THE FIELD SIZE, g_s IS THE GIRTH, M_s IS THE LENGTH OF $\mathbf{v}_j^e$ AND $(q/2) - 1$ IS THE SUFFICIENT CONDITION FOR THE SUCCESSFUL DECODING FROM THEOREM 6

$\mathbf{v}_j^e$	$q = 2^p$	g_s	M_s	m_s	$\frac{q}{2} - 1$
$\mathbf{v}_j^e \prec \mathbf{v}_j$	8	6	491	5	3
	16	12	1043	9	7
$\mathbf{v}_j^e \preccurlyeq \mathbf{v}_j$	32	10	3321	29	15
$\mathbf{\Omega}$	8	6	840	7	3
	16	6	1800	15	7
	32	8	3720	31	15

TABLE V
MSNRS FOR DIFFERENT DEGREE DISTRIBUTIONS

$\mathbf{H}$		$\mathbf{\Omega}^e$	
$\lambda(x)$	$\rho(x)$	$\lambda(x)$	$\rho(x)$
$0.153x$	$0.010x^4$	0.134x	$0.001x^3$
$0.261x^2$	$0.029x^5$	$0.254x^2$	$0.001x^4$
$0.138x^3$	$0.074x^6$	$0.178x^3$	$0.036x^5$
$0.051x^4$	$0.178x^7$	$0.081x^4$	$0.145x^6$
$0.047x^5$	$0.272x^8$	$0.027x^5$	$0.260x^7$
$0.046x^6$	$0.248x^9$	$0.005x^6$	$0.269x^8$
$0.026x^7$	$0.136x^{10}$	$0.001x^7$	$0.180x^9$
$0.007x^8$	$0.044x^{11}$	$0.002x^8$	$0.080x^{10}$
$0.001x^9$	$0.008x^{12}$	$0.011x^9$	$0.023x^{11}$
$0.001x^{17}$		$0.024x^{10}$	$0.004x^{12}$
$0.004x^{18}$		$0.031x^{11}$	
$0.012x^{19}$		$0.025x^{12}$	
$0.021x^{20}$		$0.014x^{13}$	
$0.029x^{21}$		$0.005x^{14}$	
$0.031x^{22}$		$0.001x^{15}$	
$0.026x^{23}$		$0.004x^{18}$	
$0.019x^{24}$		$0.009x^{19}$	
$0.015x^{25}$		$0.017x^{20}$	
$0.015x^{26}$		$0.023x^{21}$	
$0.019x^{27}$		$0.025x^{22}$	
$0.021x^{28}$		$0.022x^{23}$	
$0.020x^{29}$		$0.019x^{24}$	
$0.016x^{30}$		$0.017x^{25}$	
$0.011x^{31}$		$0.017x^{26}$	
$0.006x^{32}$		$0.016x^{27}$	
$0.003x^{33}$		$0.014x^{28}$	
$0.001x^{34}$		$0.011x^{29}$	
		$0.007x^{30}$	
		$0.003x^{31}$	
		$0.001x^{32}$	
		$0.001x^{33}$	
T_b	0.59dB	0.62dB	

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

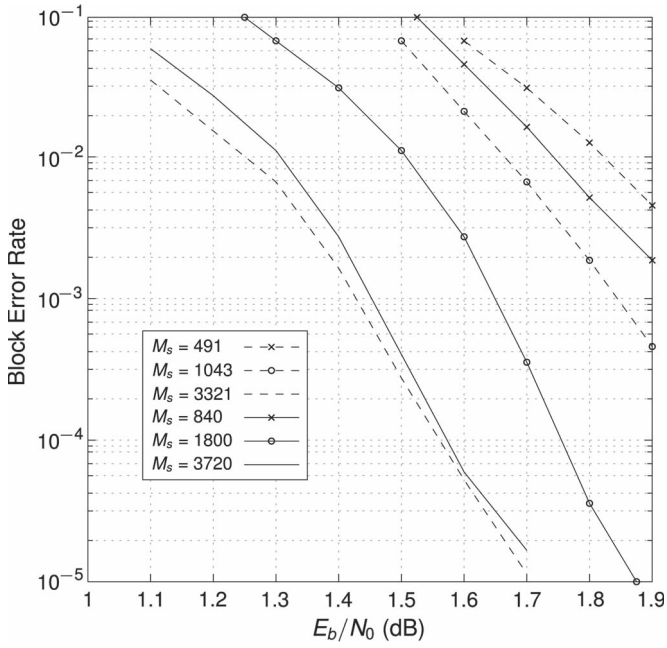

Fig. 8. Performance comparison between different outputs in Table IV.

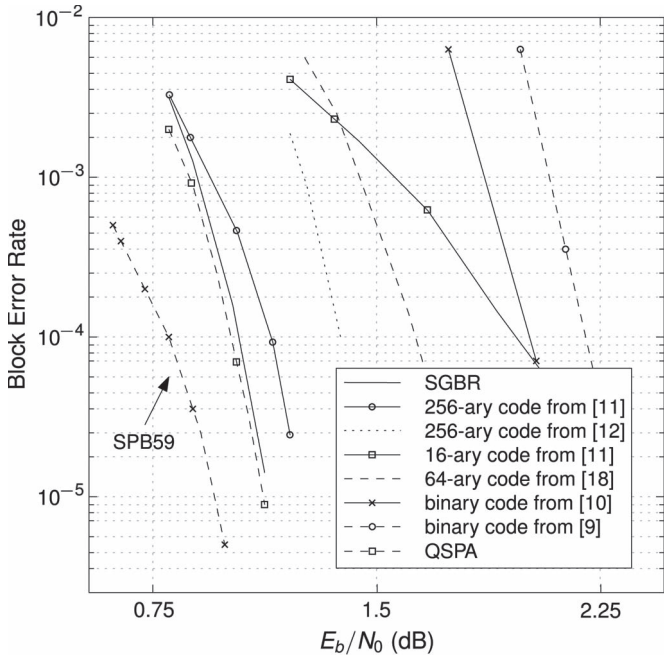

Fig. 9. The GBR compared with codes from literature.

D. Decoding Under Different (μ, ν)s

In this subsection, we compare the decoding performance under different (μ, ν)s with the Monte-Carlo (MC) experiment for "infinite" code with regard to the average syndrome bit entropy (ASBE). We consider the non-binary code over $\mathbb{F}_8$ in Section II-A. In Fig. 10, we give the ASBE versus the number of decoding iterations for different (μ, ν)s at $E_b/N_0 = 1.4$ dB. M_s for the GBR is 21000. The size of the bits set for the "infinite" code is 90000. It can be seen that the Monte-Carlo experiment could provide good approximation to the real decoding behavior.

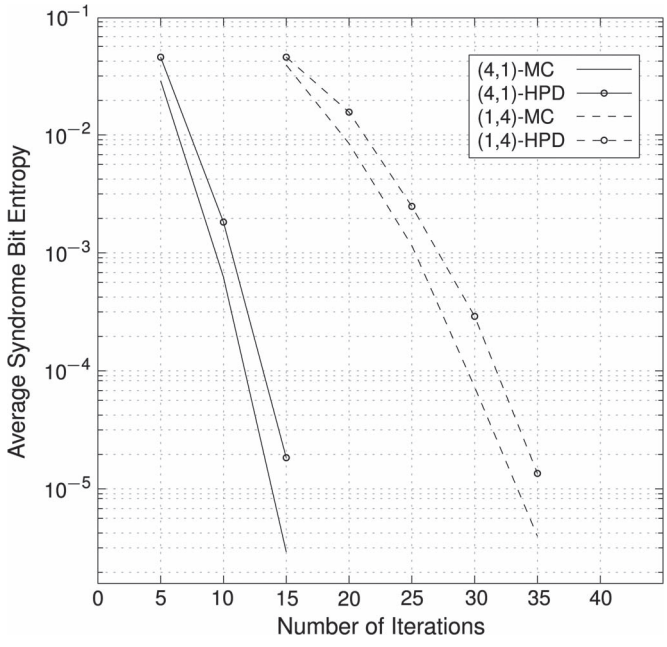

Fig. 10. The decoding under different (μ, ν)s at $E_b/N_0 = 1.4$ dB.

VI. Conclusion

In this paper, we consider the performance-optimized non-binary LDPC code over general linear group, i.e., $\bar{\mathcal{C}}$. We first propose a generalized binary representation (GBR) for $\bar{\mathcal{C}}$. The main advantage of the GBR is that it can be optimized with regard to both girth and irregular code profile (primarily the irregular code profile). As to the decoding of the GBR, we develop a hybrid parallel decoding process which could have both good performance threshold and fast convergence speed. Simulations show that the bit-level decoding performance of the GBR could closely approach the symbol-level decoding performance of the optimized $\bar{\mathcal{C}}$ while the computational complexity is only $O(m_s)$ where $m_s < q$.

References

[1] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
[2] M. C. Davey and D. J. MacKay, *Error-Correction Using LDPC Codes*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[3] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.
[4] F. Brannstrom, L. Rasmussen, and A. Grant, "Convergence analysis and optimal scheduling for multiple concatenated codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3354–3364, Sep. 2005.
[5] G. Li, I. Fair, and W. Krzymien, "Density evolution for nonbinary LDPC codes under Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 997–1015, Mar. 2009.
[6] L. Sassatelli and D. Declercq, "Nonbinary hybrid LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5314–5334, Oct. 2010.
[7] V. Savin, "Non-binary LDPC codes over the binary erasure channel: Density evolution analysis," in *Proc. 1st ISABEL*, Oct. 2008, pp. 1–5.
[8] Y. Wang, S. Draper, and J. Yedidia, "Hierarchical and high-girth qc LDPC codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4553–4583, Jul. 2013.
[9] C. Spagnol, M. Rossi, and M. Sala, "Quasi-cyclic LDPC codes with high girth," *CoRR*, vol. abs/0906.3410, 2009.
[10] G. Zhang and X. Wang, "Girth-12 quasi-cyclic LDPC codes with consecutive lengths," *CoRR*, vol. abs/1001.3916, 2010.

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.

[11] J. Huang, S. Zhou, J. Zhu, and P. Willett, "Group-theoretic analysis of cayley-graph-based cycle gf(2p) codes," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1560–1565, Jun. 2009.
[12] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular (2,dc)-LDPC codes over gf(q) using their binary images," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1626–1635, Oct. 2008.
[13] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over gf(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.
[14] V. Savin, "Binary linear-time erasure decoding for non-binary LDPC codes," in *Proc. IEEE ITW*, Oct. 2009, pp. 258–262.
[15] L. P. Sy, V. Savin, and D. Declercq, "Extended non-binary low-density parity-check codes over erasure channels," in *Proc. IEEE ISWCS*, 2011, pp. 121–125.
[16] B. Smith, M. Ardakani, W. Yu, and F. Kschischang, "Design of irregular LDPC codes with optimized performance-complexity tradeoff," *IEEE Trans. Commun.*, vol. 58, no. 2, pp. 489–499, Feb. 2010.
[17] Y. Yu and W. Chen, "Design of low complexity non-binary LDPC codes with an approximated performance-complexity tradeoff," *IEEE Commun. Lett.*, vol. 16, no. 4, pp. 514–517, Apr. 2012.
[18] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Split non-binary LDPC codes," in *Proc. IEEE ISIT*, 2008, pp. 955–959.
[19] X. Wang and X. Ma, "A class of generalized LDPC codes with fast parallel decoding algorithms," *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 531–533, Jul. 2009.
[20] Y. Yu, W. Chen, and L. Wei, "Design of convergence-optimized non-binary LDPC codes over binary erasure channel," *IEEE Wireless Commun. Lett.*, vol. 1, no. 4, pp. 336–339, Aug. 2012.
[21] A. Bhatia, A. Iyengar, and P. Siegel, "Enhancing binary images of non-binary LDPC codes," in *Proc. IEEE Global Telecommun. Conf.*, 2011, pp. 1–6.
[22] V. Savin, "Fourier domain representation of non-binary LDPC codes," in *Proc. IEEE ISIT*, 2012, pp. 2541–2545.
[23] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*. New York, NY, USA: Cambridge Univ. Press, 1986.
[24] X. Ma and B. Bai, "A unified decoding algorithm for linear codes based on partitioned parity-check matrices," in *Proc. IEEE ITW*, Sep. 2007, pp. 19–23.
[25] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland Publ. Comp., 1977.
[26] Y. Yu, W. Chen, J. Li, and B. Geller, "Cooperative decoder design for non-binary LDPC code with coefficients selection," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2013, pp. 1868–1873.

Yang Yu received the B.S. and M.S. degrees from Southwest Jiao Tong University, Chengdu, China, in 2005 and 2008, respectively. He is currently working toward the Ph.D. degree in the Network Coding and Transmission Laboratory, Shanghai Jiao Tong University, Shanghai, China. His current research interests include channel coding theory and network coding.

Wen Chen (M'03–SM'11) received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 1990 and 1993, respectively, and the Ph.D. degree from The University of Electro-Communications, Tokyo, Japan, in 1999. From 1999 to 2001, he was a Researcher with the Japan Society for the Promotion of Science. In 2001, he joined the University of Alberta, Canada, starting as a Postdoctoral Fellow with the Information Research Laboratory and continuing as a Research Associate in the Department of Electrical and Computer Engineering. Since 2006, he has been a Full Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is also the Director of the Institute for Signal Processing and Systems. His research interests include network coding, cooperative communications, cognitive radio, and MIMO-OFDM systems.

Jun Li (M'09) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009. From January 2009 to June 2009, he was a Research Scientist with the Department of Research and Innovation, Alcatel-Lucent Shanghai Bell. From June 2009 to April 2012, he was a Postdoctoral Fellow at the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. Since April 2012, he has been a Research Fellow at the School of Electrical Engineering, The University of Sydney, Sydney, Australia. His research interests include network information theory, channel coding theory, wireless network coding, and cooperative communications. Dr. Li served as a Technical Program Committee Member for several international conferences such as APCC2009, APCC2010, VTC2011 (Spring), ICC2011, TENCON2012, APCC2013, VTC2014 (Fall), and ICC2014.

Xiao Ma received the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2000. From 2000 to 2002, he was a Postdoctoral Fellow with Harvard University, Cambridge, MA, USA. From 2002 to 2004, he was a Research Fellow with City University of Hong Kong. He is currently a Professor with the Department of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include information theory, channel coding theory, and their applications to communication systems and digital recording systems. Dr. Ma is a member of the IEEE. He was a corecipient, with A. Kavčić and N. Varnica, of the 2005 IEEE Best Paper Award in Signal Processing and Coding for Data Storage. He was a recipient of the Microsoft Professorship Award from Microsoft Research Asia in 2006.

Baoming Bai received the B.S. degree from Northwest Institute of Telecommunication Engineering, Xi'an, China, in 1987, and the M.S. and Ph.D. degrees in communication engineering from Xidian University, Xi'an, in 1990 and 2000, respectively. From 2000 to 2003, he was a Senior Research Assistant with the Department of Electronic Engineering, City University of Hong Kong. Since April 2003, he has been with the State Key Laboratory of Integrated Services Networks, School of Telecommunication Engineering, Xidian University, where he is currently a Professor. In 2005, he was a Visiting Scholar with the University of California, Davis. His research interests include information theory and channel coding, wireless communication, and quantum communication.

Authorized licensed use limited to: NANJING UNIVERSITY OF SCIENCE AND TECHNOLOGY. Downloaded on January 11,2021 at 11:50:39 UTC from IEEE Xplore. Restrictions apply.