Regular and irregular LDPC code design for bandwidth efficient BICM schemes

Junyi Du†, Liang Zhou†, Zhe Zhang†, Lei Yang† and Jinhong Yuan‡
†National Key Lab on Communication, University of Electronic Science and Technology of China, Chengdu, P. R. China
‡School of Electrical Engineering & Telecommunications, University of New South Wales, Sydney, Australia
Email: jydu@std.uestc.edu.cn, lzhou@uestc.edu.cn, zhang@std.uestc.edu.cn, lei.byron.yang@gmail.com, j.yuan@unsw.edu.au

Abstract—We consider low-density parity-check (LDPC) code design by considering the unequal error protection property in high order modulated bit-interleaved coded modulation (BICM) schemes. The existing work mainly considered the effect of variable node edge assignments on the decoding performance. In this paper, we consider both variable node and check node edge assignments to further optimize the LDPC codes for BICM schemes. To achieve this, we derive new extrinsic information transfer (EXIT) functions for both regular and irregular LDPC code ensembles. Then we employ differential evolution to optimize the code ensembles in terms of the lowest decoding threshold. Finally, we propose a modified progressive edge growth algorithm to design regular and irregular LDPC codes based on the optimized code ensembles. The numerical results show that our designed LDPC codes have better bit error rate performance compared to the codes designed in the existing work.

Index Terms—LDPC; high order modulations; progressive edge growth algorithm.

I. INTRODUCTION

Recently, low-density parity-check (LDPC) code design has been adopted by the 3rd Generation Partnership Project as the channel coding scheme for the 5G enhanced mobile broadband data channel. 5G requires much higher network throughput and a higher spectrum efficiency, which calls for the design of LDPC coded high order modulation schemes. The bit-interleaved coded modulation (BICM) scheme is a flexible and simple implementation scheme for applications of high order modulations [5]. The BICM scheme with Gray labeling can approach the coded modulation (CM) capacity [6]. Therefore, the design of LDPC coded BICM schemes is important in practical communication system.

One of the key features of high order modulated BICM schemes is the unequal error protection (UEP) property [5]. The high order modulation channel is equivalent to several parallel and independent sub-channels with unequal mutual information. This UEP property introduces multi-edge type to the Tanner graph edges. The LDPC codes designed in [2]–[4] are for BPSK modulated schemes where coded bits have the same channel output error probability. These codes are not optimal if we employ them directly to an inherent unequal protection modulation scheme, such as the high order modulated BICM system.

To address this problem, some recent related work focused on designing bit-interleavers for the BICM schemes with given LDPC codes [7]–[11]. They designed the bit-interleaver between the LDPC encoder and the modulator to change the positions of LDPC coded bits according to the UEP property of the BICM scheme. Although this provides improved bit error rate (BER) performance than the consecutive bit mapping schemes, the improvement was limited as the code design and the bit-interleaver design were separated.

To improve the performance further, the recent work [12]–[15] aimed to jointly design the LDPC codes and the bit-interleavers. With respect to the multi-edge type in the Tanner graph, they divided the variable node (VN) degree distributions into sub-distributions on the sub-channels. On top of that, they developed analytical tools, like density evolution and extrinsic information transfer (EXIT) chart, to optimize the code ensembles in terms of lowest decoding threshold. Then the progressive edge growth (PEG) algorithm [4] was employed to design practical finite-length irregular LDPC codes. These designs achieved further improved BER performance than the bit-interleavers designed in [7]–[11], however, they mainly considered the variable node (VN) edge assignments to optimize the code ensembles. In fact, the check node (CN) edge assignments also affect the BER performance.

In the update process of a check node (CN) with degree $d_c$, its output message is determined by the types of the other $d_c − 1$ edges. With multi-edge-type in the Tanner graph, there are multiple assignments of the $d_c − 1$ edges. Different assignments of the $d_c − 1$ edges would make the CN output different messages. Therefore, the BER performance should be further improved by considering both VN and CN edge assignments of the LDPC code ensembles.

In this paper, we contribute to develop new EXIT functions by considering both VN and CN edge assignments. We propose a CN UEP distribution matrix and a VN UEP distribution matrix to record the CN edge assignments and VN degree distributions, respectively. In addition, we employ differential evolution to optimize the CN and VN UEP distribution matrix in terms of lowest decoding threshold. Finally, we modify the PEG algorithm to design finite-length LDPC codes in accordance with the optimized matrixes.

Compared to our previous work [11], we have fulfilled two new contributions in this paper. First of all, the work in [11] aimed to design the bit-interleaver and the work in this paper focuses on LDPC code design. The designed LDPC codes
have better BER performances as we jointly design the LDPC code and the bit-interleaver. Secondly, we develop generalized EXIT functions for both regular and irregular LDPC code ensembles. The EXIT functions proposed in [11] are derived for regular LDPC code ensemble and are special cases of the EXIT functions developed in this paper.

II. SYSTEM MODEL

The sample block diagram of an LDPC coded BICM scheme is illustrated in Fig. 1. The encoder adopts an LDPC code with length N and rate R. The CN number is \( N_c = N(1 - R) \).\(^1\) The source generates a data sequence \( u \) of independent and identically distributed uniform Bernoulli random variables. Then the sequence \( u \) is encoded by the LDPC encoder to \( b \). After that, the \( M \)-ary QAM modulator maps \( m = \log_2 M \) consecutive bits \( \{b_1, b_2, \ldots, b_m\} \) to a complex symbol \( x \) from the QAM constellation \( \chi \).

The transmitted vector \( x \) is corrupted by complex additive white Gaussian noise (AWGN) \( n = n^t + jn^q \), where \( n^t \) and \( n^q \) are independent Gaussian variables with zero mean and variance \( \sigma_n^2 = N_0/2 \). The receiver gets the channel observation \( y = x + n \).

Based on the channel observation vector \( y \), the demodulator calculates the log-likelihood ratio (LLR) for each coded bit \( b_k \), \( k \in \{1, 2, \ldots, m\} \), as

\[
L_c(b_k) = \log \frac{\sum_{x \in \chi^k} p(y|x)}{\sum_{x \in \chi^k} p(y|x)},
\]

where \( \chi^k \) denotes the subset of symbols \( x \in \chi \) with the value \( b_k \in \{0, 1\} \) at the \( k \)-th position. Then the LDPC decoder decodes the LLR vector \( L_c(b) \) into estimated data vector \( \hat{u} \).

III. UEP IN HIGH ORDER MODULATED BICM SCHEMES

For a BICM scheme with an ideal bit interleaver, the \( M \)-ary QAM modulation channel can be considered as \( m \) independent parallel binary-input sub-channels for simplification [5]. The mutual information of the \( k \)-th sub-channel, \( k \in \{1, 2, \ldots, m\} \), between the channel input \( b_k \) and output \( y \) is calculated by

\[
I(b_k; y) = 1 - E_{b_k,y} \left[ \log_2 \frac{\sum_{x \in \chi^k} p(y|x)}{\sum_{x \in \chi^k} p(y|x)} \right],
\]

where \( \chi^k \) denotes the subset of the constellation \( \chi \) with the \( k \)-th bit position \( b_k \in \{0, 1\} \) and \( p(y|x) \) is the channel conditional probability density function. The expectation \( E_{b_k,y}[\cdot] \) is taken with \( p(b_k) = 1/2 \) and \( p(y|b_k) = 2^{1-m} \sum_{x \in \chi^k} p(y|x) \).

\(^1\)The LDPC code is row full rank in this paper for simplicity.

For a Gray labelled \( M \)-ary QAM, it has \( m \) sub-channels with \( m/2 \) mutual information levels. The \( m \) sub-channels have unequal mutual information and the coded bits are not equally protected [11]. The sub-channels can be classified into two groups in terms of their mutual information. We call the sub-channels with higher mutual information as reliable sub-channel group \( H \) and the other sub-channels as unreliable sub-channel group \( L \). We call the VNs connecting to the sub-channels in the reliable group as reliable VNs and the VNs connecting to the sub-channels in the unreliable group as unreliable VNs. Besides, we call the edges emanating from the reliable VNs as reliable edges \( E_H \) and other edges as unreliable edges \( E_L \).

For a CN with degree \( d_c \), its output LLR value on an edge is mainly determined by the minimum input LLRs from other \( d_c - 1 \) edges. In addition, unreliable VNs have the largest probability to send out small LLR values. Thus, the decoding performance is mainly determined by the assignment of unreliable VNs to CNs [7]. In the following part of this paper, we aim to design the optimal assignment of unreliable VNs to the CNs in terms of lowest decoding threshold.

IV. LDPC CODE DESIGN

In this section, we aim to design finite-length LDPC code according to the UEP property. We first derive EXIT functions by taking into account both VN and CN edge assignments to calculate the decoding threshold of an LDPC code ensemble. Then we employ the differential evolution algorithm and the developed EXIT functions to obtain the optimized LDPC code ensembles in terms of decoding threshold. At last, we propose a PEG algorithm to generate practical finite-length LDPC code according to the optimized ensemble.

A. EXIT functions

For the irregular LDPC code ensemble design, it is shown that a concentrated CN degree can simplify the design procedure with negligible loss in optimality [2]. In this paper, we apply the same simplification and fix the CN degree of the irregular LDPC code as \( d_c \). The VNs have \( V \) different degrees and the degree distribution is \( (d_1, d_2, \ldots, d_v) \), where \( d_i < d_j \) with \( i < j \). For a regular LDPC code ensemble, we denote its CN degree and VN degree as \( d_c \) and \( d_v \), respectively.

In the following part of this section, we develop the EXIT functions of the irregular LDPC code ensembles. A regular LDPC code ensemble is a special case of an irregular LDPC code ensemble where the VN degree is unique. The EXIT functions developed in this paper are also suitable for regular LDPC code ensembles. Therefore, the developed EXIT functions in [11] are special cases of the proposed EXIT functions in this paper.

1) VN decoder (VND) EXIT functions: Considering a VN with degree \( d_i \) and mapped to a sub-channel with the \( i \)-th level of mutual information, where \( j \in \{1, 2, \ldots, V\} \) and \( i \in \{1, 2, \ldots, m/2\} \). Its VND function is

\[
I_{E,VND}(I_A, d_j, \sigma_i) = J \left( \frac{(d_j - 1)[I^{-1}(I_{E,VND})]^2 + \sigma_i^2}{\sqrt{(d_j - 1)[I^{-1}(I_{E,VND})]^2 + \sigma_i^2}} \right),
\]
where $J(\cdot)$ is defined in [16], and $I_{A,VND}$ denotes the a priori information from the CND. The term $\sigma^2_i$ is the equivalent noise variance of the $i$-th sub-channel

$$\sigma_i = J^{-1}(I(b_i;Y)).$$

There are $\frac{m}{2} \times V$ individual different VND functions in total. A VN UEP distribution matrix $\mathbf{P} = [p_{i,j}]_{\frac{m}{2} \times V}$ is used to model the distributions of different degree coded bits on different sub-channels [8]. The element $p_{i,j}$ represents the proportion of the VNs with degree $d_i$ that are mapped to the sub-channels with the $i$-th level mutual information, where $i \in \{1,2,\ldots,m/2\}$ and $j \in \{1,2,\ldots,V\}$. The elements $p_{i,j}$ are subject to the following constraints

$$\begin{align*}
0 \leq p_{i,j} &\leq 1 \\
\sum_{i=1}^{m/2} p_{i,j} &= \lambda_j, \\
\sum_{j=1}^{V} p_{i,j} &= 2/m
\end{align*}$$

(1)

where $\lambda_j$ is the percentage of the $d_j$-degree VN among all VNs. The percentage of sub-channels with the $i$-th level mutual information is $2/m$. The percentages of reliable VNs and unreliable VNs are equal to $(m-2)/m$ and $2/m$ in Gray labelled $M$-ary QAM, respectively.

For the reliable edges $E_R$ with a priori information $I_{A,VND}^R$ from the CND, the VND EXIT function is

$$I_{E,VND}^R(I_{A,VND}^R, \mathbf{P}, \sigma) = \frac{\frac{m}{2} - 1}{\sum_{i=1}^{m/2} \sum_{j=1}^{V} p_{i,j}d_j I_{E,VND}^R(I_{A,VND}^R, d_j, \sigma_i)}.$$  

(2)

For the unreliable edges $E_L$ with a priori information $I_{A,VND}^L$ from the CND, the VND EXIT function is

$$I_{E,VND}^L(I_{A,VND}^L, \mathbf{P}, \sigma) = \frac{\sum_{j=1}^{V} \sum_{i=1}^{m/2} p_{i,j}d_j I_{E,VND}^L(I_{A,VND}^L, d_j, \sigma)}{\sum_{j=1}^{V} \sum_{i=1}^{m/2} p_{i,j}d_j}.$$  

(3)

The extrinsic information $I_{E,VND}^R$ and $I_{E,VND}^L$ is conveyed to the CND as its a priori information $I_{A,CND}^R$ and $I_{A,CND}^L$, respectively.

2) CND EXIT functions: There are two types of edges in the Tanner graph: reliable edges $E_R$ from reliable VNs and unreliable edges $E_L$ from unreliable VNs. For a CN with degree $d_c$, there are $d_c + 1$ different edge assignments, as the number of unreliable edges ranges from 0 to $d_c$. We denote $\mathbf{Q}$ to model the CN UEP distribution of the CNs. We call a CN as a type-$i$ CN if it has $i - 1$ unreliable edges, where $i \in \{1,2,\ldots,d_c+1\}$. Let $q_i$ denote the fraction of the type-$i$ CNs. The elements $q_i$ are subject to

$$\begin{align*}
0 \leq q_i &\leq 1 \\
\sum_{i=1}^{d_c+1} q_i(i-1) &\sum_{j=1}^{V} p_{m/2,j}d_j/(1-R),
\end{align*}$$

(4)

where $R$ is the rate of the code ensemble. The term $\sum_{i=1}^{d_c+1} q_i(i-1)N_c$ denotes the number of the unreliable edge emanating from CNs. The term $\sum_{j=1}^{V} p_{m/2,j}d_jN$ denotes the number of the unreliable edges emanating from the VNs. Since the unreliable edge number emanating from both sides should be equal and $\sum_{i=1}^{d_c+1} q_i(i-1)N_c = \sum_{j=1}^{V} p_{m/2,j}d_jN$. The second equation in (4) is achieved based on the condition that $N_c = N(1-R)$.

For an edge emanating from a CN with degree $d_c$, its reliability is determined by the edge assignments of the other $d_c - 1$ edges. As a result, there are $d_c$ types of the edges emanating from the CNs. We call an edge as a type-$k$ edge if its connected CN has $k$ unreliable edges among the other $d_c - 1$ edges, where $k \in \{0,1,\ldots,d_c-1\}$. According to the a priori information $I_{E,R}^R$ from the reliable edge $E_R$ and $I_{E,L}^R$ from the unreliable edge $E_L$, we can derive its extrinsic information $I_{E,CND}^R$ of type-$k$ edges as (5).

Consider a type-$i$ CN, which is assigned to $(i-1)$ unreliable edges $E_L$. It provides type-$(i-2)$ extrinsic information $I_{E,CND}^{R-2}$ and type-$i$ extrinsic information $I_{E,CND}^{i-1}$ to the connected $(i-1)$ unreliable VNs and $(d_c - i + 1)$ reliable VNs, respectively. We present an example of a type-4 CN in Fig. 2. The CN provides $I_{E,CND}^0$ to the three connected reliable VNs and $I_{E,CND}^1$ to the three connected reliable VNs.

We denote two length-$d_c$ vectors $\mathbf{r} = (r_0, r_1, \ldots, r_{d_c-1})$ and $\mathbf{u} = (u_0, u_1, \ldots, u_{d_c-1})$ to record the distribution of extrinsic information types for reliable VNs and unreliable VNs, respectively. The elements $r_k$ and $u_k$ respectively represent the fraction of the type-$k$ extrinsic information $I_{E,CND}^k$ sent to reliable VNs and unreliable VNs, where $k \in \{0,1,\ldots,d_c-1\}$.

For the reliable VNs, they can obtain the type-$k$ extrinsic information only from the type-$(k+1)$ CNs. Consequently, let $r_k = (d_c - k)q_{k+1}$. For the unreliable VNs, they can obtain the type-$k$ extrinsic information only from the type-$(k+2)$ CNs. Consequently, let $u_k = (k+1)q_{k+2}$. The total number of reliable edges emanating from CNs is $\sum_{k=0}^{d_c-2} r_k = 1/R \sum_{k=1}^{m/2-1} \sum_{j=1}^{V} p_{i,j}d_j$ and the total number of unreliable edges emanating from CNs is $\sum_{k=0}^{d_c-2} u_k = 1/R \sum_{k=1}^{m/2-1} \sum_{j=1}^{V} p_{i,j}d_j$, we can normalize $r_k$ and $u_k$ from $\mathbf{Q}$ as

$$\begin{align*}
\frac{r_k}{1/R \sum_{k=1}^{m/2-1} \sum_{j=1}^{V} p_{i,j}d_j}
\frac{u_k}{1/R \sum_{k=1}^{m/2-1} \sum_{j=1}^{V} p_{i,j}d_j},
\end{align*}$$

(6)

$$I_{E,CND}^R(I_{A,CND}^R, I_{A,CND}^L) = 1 - J \left( \sqrt{(d_c - k - 1)(J^{-1}(1 - I_{A,CND}^R))^2 + k(J^{-1}(1 - I_{A,CND}^L))^2} \right).$$

(5)
and
\[ u_k = \frac{(k + 1)q_k+2}{1 - \frac{1}{R} \sum_{j=1}^{V} p_m/2,j d_j}. \]  

The CND EXIT function for the reliable edges \( E_H \),
\[ I_{E,CND}^H(I_{A,CND}^H, I_{A,CND}^L) = \sum_{k=0}^{d_r-1} r_k I_{E,CND}^L(I_{A,CND}^H, I_{A,CND}^L) \]  

The EXIT function for the unreliable edges \( E_L \),
\[ I_{E,CND}(I_{A,CND}^H, I_{A,CND}^L) = \sum_{k=0}^{d_r-1} u_k I_{E,CND}^L(I_{A,CND}^H, I_{A,CND}^L). \]

The extrinsic information \( I_{E,CND}^H \) and \( I_{E,CND}^L \) are fed back to the VND as its a priori information \( I_{A,VND}^H \) and \( I_{A,VND}^L \), respectively.

**B. Optimization of code ensembles**

We employ differential evolution [17] and the derived EXIT functions to find the optimized code ensembles in terms of lowest decoding threshold. We first explain the employed parameters in the following. The VN UEP distribution \( \mathbf{P} \) has a size of \( \frac{m}{2} \times V \). We set the population number scaling with the size of \( \mathbf{P} \), such as \( 10 \times \frac{m}{2} \times V \). To generate each candidate \( \mathbf{P} \), we first generate \( V - 1 \) elements of \( \mathbf{P} \), s.t. \( 0 \leq p_i \leq 1 \), using differential evolution. Then we calculate the other \( V + 1 \) elements of \( \mathbf{P} \) to satisfy the constraints in (1).

The size of the CN UEP distribution \( \mathbf{Q} \) is \( d_r + 1 \), we set the population number scaling with the size of \( \mathbf{Q} \), such as \( 10 \times (d_r + 1) \). To generate each candidate \( \mathbf{Q} \), we first generate \( d_r - 1 \) elements of \( \mathbf{Q} \), s.t. \( 0 \leq q_i \leq 1 \), using differential evolution. Then we calculate the other elements of \( \mathbf{Q} \) to satisfy the constraints in (4).

Both the crossover probability and differential weight for \( \mathbf{P} \) and \( \mathbf{Q} \) are set to 0.5. The maximum generation number is set to 1000. We do not observe much improvement in terms of decoding threshold by increasing the generation number further. The design algorithm is described as below:

1. For \( i = 1 \) to 1000, go through Steps 2-6.
2. Generate \( 10 \times \frac{m}{2} \times V \) candidates of \( \mathbf{P} \) in the \( i \)-th generation using differential evolution. For each candidate \( \mathbf{P}_i^{(k)} \), where \( k = 1, 2, \ldots, 10 \times \frac{m}{2} \times V \), go through Steps 3-5.
3. For \( j = 1 \) to 1000, go through Steps 4-5.
4. Generate \( 10 \cdot (d_r + 1) \) candidates of \( \mathbf{Q} \) in the \( j \)-th generation using differential evolution, go to Step 5.
5. Calculate the decoding threshold based on \( \mathbf{P}_i^{(k)} \) and the candidate \( \mathbf{Q} \). Record the optimal \( \mathbf{Q}_i^{(k)} \) in terms of lowest decoding threshold.
6. Record the optimal \( \mathbf{P}_i \) and its corresponding optimal \( \mathbf{Q}_i \) in terms of the lowest decoding threshold in the \( i \)-th generation.
7. Output the recorded optimal matrixes \( \mathbf{P}^* \) and \( \mathbf{Q}^* \).

The obtained VN UEP distribution matrix \( \mathbf{P}^* \) and CN UEP distribution matrix \( \mathbf{Q}^* \) represent the optimal code ensemble in terms of the lowest decoding threshold, while the code length is supposed to be infinite. In the next part of this section, we aim to design finite-length LDPC code based on \( \mathbf{P}^* \) and \( \mathbf{Q}^* \).

**C. PEG algorithm for UEP schemes**

The conventional PEG algorithm is proposed to design LDPC codes for BPSK schemes [4]. In this section, we aim to modify the PEG algorithm and design practical finite-length LDPC code for high order modulation schemes. The design principle is to generate LDPC code to ensure its VN UEP distribution and CN UEP distribution equal to \( \mathbf{P}^* \) and \( \mathbf{Q}^* \), respectively.

For the aspect of VNs, we denote \( \varphi_i \) as the degree of the \( i \)-th VN, where \( i \in \{1, 2, \ldots, N\} \). The VNs are ordered in a nondecreasing order in terms of their degrees and \( \varphi_i \leq \varphi_k \), if \( 1 \leq i < k \leq N \). For VN \( v_i \), its degree is chosen according to the searched matrix \( \mathbf{P}^* \). We let \( \varphi_i = d_j \), where \( j \in \{1, 2, \ldots, V\} \) and \( \sum_{k=1}^{j} \sum_{l=1}^{V} p_{i,k}^* N < i \leq \sum_{k=1}^{j+1} \sum_{l=1}^{V} p_{i,k}^* N \). Fig. 3 presents the VN degree distribution of the length-\( N \) LDPC code. Note that, the VNs mapped to the second sub-channel are unreliable VNs. We mark the unreliable VNs as black in Fig. 3 and we record their positions.

For the aspect of CNs, we denote \( \omega_i \) as the unreliable edge number of the \( i \)-th CN, where \( i \in \{1, 2, \ldots, N_c\} \). The CNs are ordered in a nondecreasing order in terms of their connected unreliable edge number. For CN \( c_i \), its unreliable edge number is chosen according to \( \mathbf{Q} \). We let \( \omega_i = j \), where \( j \in \{0, 1, \ldots, d_c\} \) and \( \sum_{i=1}^{j} \sum_{l=1}^{N_c} q_{i,l}^* N_c < i \leq \sum_{i=1}^{j+1} \sum_{l=1}^{N_c} q_{i,l}^* N_c \). Fig. 3 presents the CN unreliable edge number distribution of the \( N_c \) CNs.

The edges can be established from CNs to VNs or from VNs to CNs. We observed that the constructed code has a better BER performance if we establish the edges from CNs to VNs for regular LDPC codes. On the other hand, the
constructed code performs better if we establish the edges from VNs to CN for irregular LDPC codes. Based on our observations, establishing edges from CNs to VNs for regular LDPC code and from VNs to CNs for irregular LDPC code can design LDPC codes with good code structure. In this paper, we present two different construction methods for regular and irregular LDPC codes.

For regular LDPC codes, the design algorithm is described as below:

1. We denote $H$ to record the LDPC Tanner graph.
2. For CN $c_i$ with degree $d_c$, denote $a_i$ and $b_i$ to record the established reliable and unreliable edges of $c_i$, $i = 1,2,\ldots,N_c$. Let $a_i = 0$ and $b_i = 0$, go through Steps 3-6.
3. For $t = 1$ to $d_c$, establish $d_c$ edges to the VNs, go through Steps 4-6.
4. For each VN $v_k$, $k = 1,2,\ldots,N$. Denote $\gamma_k$ as an indicator. Initialize $\gamma_k = 0$. If $v_k$ is reliable and $a_k < d_v - \omega_k$, let $\gamma_k = 1$. If $v_k$ is unreliable and $b_k < \omega_k$, let $\gamma_k = 1$. Go to Step 5.
5. Group the VNs $v_k$ with $\gamma_k = 1$. Find the VNs in the group with the largest distance to the CN $c_i$. If there are multiple candidates, randomly choose a VN $v_l$. Let $h_{i,t} = l$. Go to Step 6.
6. If the VN $v_l$ is reliable, let $a_l = a_l + 1$. If the VN $v_l$ is unreliable, let $b_l = b_l + 1$.
7. Output $H$ as the generated LDPC code.

For irregular LDPC codes, the design algorithm is described as below:

1. We denote $a_i$ and $b_i$ to record the established reliable and unreliable edges of the $i$-th CN $c_i$ respectively. The initial value $a_i = 0$ and $b_i = 0$, $i \in \{1,2,\ldots,N_c\}$. We denote $H$ to record the LDPC Tanner graph.
2. For VN $v_l$ with degree $\varphi_i$, $i = 1,2,\ldots,N$, go through Steps 3-6.
3. For $t = 1$ to $\varphi_i$, establish $\varphi_i$ edges to the CNs, go through Steps 4-6.
4. For each CN $c_k$, $k = 1,2,\ldots,N_c$. Denote $\beta_k$ as an indicator. Initialize $\beta_k = 0$. If $v_l$ is reliable and $a_k < d_c - \omega_k$, let $\beta_k = 1$. If $v_k$ is unreliable and $b_k < \omega_k$, let $\beta_k = 1$. Go to Step 5.
5. Group the CNs $c_k$ with $\beta_k = 1$. Find the CNs in the group with the largest distance to the VN $v_l$. If there are multiple candidates, randomly choose a CN $c_l$. Let $h_{l,t} = l$. Go to Step 6.
6. If the VN $v_l$ is reliable, let $a_l = a_l + 1$. If the VN $v_l$ is unreliable, let $b_l = b_l + 1$.
7. Output $H$ as the generated LDPC code.

The generated $H$ records the LDPC Tanner graph edges. We can easily obtain a practical LDPC code based on the recorded edges in $H$.

V. NUMERICAL RESULTS

In this section, we evaluate the decoding thresholds and BER performances of our designed LDPC codes and compare them to the those designed in the existing published works. We employ the proposed EXIT functions to obtain optimized VN UEP distribution $P_v^*$ and CN UEP distribution $Q^*$ for both regular and irregular LDPC code ensembles. In the simulations, we set the parameters exactly the same with the existing works for the consideration of fairness. We consider the Gray labelled 16-QAM in accordance with the work [6] and [13]. On top of that, we set the code length to 10000 for regular LDPC code as the same with the code designed in [6] and 8192 for irregular LDPC code as the same with the code designed in [13].

For the $(3,6)$-regular LDPC code ensemble in Gray labelled 16-QAM BICM scheme, the optimized CN UEP distribution $Q_{\text{reg}}$ is shown in Table I. Employing the EXIT functions, the decoding threshold $E_b/N_0$ is 3.37 dB for LDPC code ensemble optimized by our method. To compare with the work [6], we further employ the proposed PEG algorithm to generate a regular LDPC code with the same length 10000. Its CN UEP distribution is equal to $Q_{\text{reg}}$. We also employ the conventional PEG algorithm [4] to construct a length-10000
TABLE I
UEP DISTRIBUTIONS FOR REGULAR AND IRREGULAR LDPC CODE ENSEMBLES WITH GRAY LABELLED 16-QAM.

<table>
<thead>
<tr>
<th>Regular code ensemble</th>
<th>Irregular code ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q_{\text{ir}} )</td>
<td>( P_{\text{ir}} )</td>
</tr>
<tr>
<td>( d_v )</td>
<td>( d_v )</td>
</tr>
<tr>
<td>3</td>
<td>group H</td>
</tr>
<tr>
<td>5</td>
<td>group L</td>
</tr>
<tr>
<td>6</td>
<td>group L</td>
</tr>
</tbody>
</table>

The (3, 6)-regular LDPC code. Its associated decoding threshold is 3.55 dB. Based on this regular LDPC code, we design two different bit-interleavers according to the algorithms proposed in [9] and [11], respectively. To be noticed, the bit-interleaver designed by [11] can only approach the optimized CN UEP distribution \( Q_{\text{reg}} \). Its decoding threshold is 3.45 dB.

Fig. 4 shows the BER simulations of regular LDPC coded 16-QAM BICM schemes. Compared to the LDPC code designed by [6] and the conventional PEG algorithm [4], our designed LDPC code has about 0.1 dB and 0.13 dB gain at the BER of \( 10^{-5} \), respectively. Compared to the bit-interleaver based schemes [9] and [11], our designed LDPC code has about 0.16 dB and 0.05 dB gain at the BER of \( 10^{-5} \), respectively. The numerical results verify the effectiveness of our proposed PEG algorithm for regular LDPC code.

For the irregular LDPC code ensembles in Gray labelled 16-QAM BICM scheme, we set \( V = 9 \) and the VN degree ranging from 2 to 10. The size of the VN UEP distribution \( P_{\text{ir}} \) is \( 2 \times 9 \). The optimized VN UEP distribution \( P_{\text{ir}} \) and CN UEP distribution \( Q_{\text{ir}} \) are shown in Table I. The calculated decoding threshold \( E_b/N_0 \) of our proposed ensemble is 2.47 dB.

To further compare the BER performance to the recent work [13], we employ the proposed PEG algorithm to generate an irregular LDPC code with the same length 8192. Its VN UEP distribution and CN UEP distribution are equal to \( P_{\text{ir}} \) and \( Q_{\text{ir}} \), respectively. Besides, we employ the optimized edge degree distribution with a maximum VN degree 9 described in Table I in [18]. Then we design a length-8192 LDPC code based on the distributions employing PEG algorithm. Fig. 5 shows the BER simulations of irregular LDPC coded 16-QAM BICM schemes on AWGN channels. Compared to the LDPC code designed by [13], our designed code has about 0.1 dB gain at the BER of \( 10^{-5} \). Compared to the LDPC code generated by [18], our designed code has about 0.16 dB gain at the BER of \( 10^{-5} \). The numerical results verify the effectiveness of our proposed PEG algorithm for irregular LDPC code.

VI. CONCLUSION

In this paper, we considered the LDPC code design for high order modulated QAM modulated BICM schemes. We first showed the inherent UEP property of the QAM modulation and classified the edges in the Tanner graph into two types: reliable edges and unreliable edges. We then proposed multi-edge type EXIT functions to calculate the decoding threshold for BICM schemes with both regular and irregular LDPC code ensembles. We further employed differential evolution to obtain globally optimized LDPC ensembles in terms of lowest decoding threshold. Finally, we proposed an improved PEG algorithm to generate both regular and irregular finite-length LDPC codes. Numerical results proved the effectiveness of our proposed EXIT functions and PEG algorithm to design both regular and irregular LDPC codes.

REFERENCES