Received August 23, 2018, accepted September 16, 2018, date of publication October 1, 2018, date of current version October 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2873058

Optimized Construction of Protograph G-LDPC Codes by Modified EXIT Chart and MACE for New-Generation Wireless Communications

JIANRONG BAO[1,2], **(Member, IEEE), BIN JIANG**[3], **AND CHAO LIU**[1]

[1]Information Engineering School, Hangzhou Dianzi University, Hangzhou 310018, China
[2]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
[3]School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Chao Liu (liuchao@hdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61471152, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY17F010019, in part by the Zhejiang Provincial Science and Technology Plan Project under Grant LGG18F010011, in part by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University, under Grant 2014D02, in part by the Open Research Fund of Zhejiang Provincial Key Laboratory of Solid Hard Disk and Data Security Technology, HDU, under Grant ZJSSKL003, and in part by the Scientific Research Project of Zhejiang Provincial Department of Education under Grant Y201329723.

ABSTRACT In this paper, a new systematic construction of optimized protograph generalized low-density parity-check codes with good decoding threshold and low error floor is proposed. First, a typical code graph is generated by combining an accumulate-repeat-accumulate seed protograph with a Tanner graph extension of a simple linear block node. Subsequently, it is analyzed and optimized theoretically, especially with the puncture mechanism, by a modified extrinsic information transfer chart and an asymptotic weight distribution. The generated graph is then extended to a base matrix by a copy-and-permute procedure, accompanied with the matrix split, which is optimized by a progressive edge growth for good randomness and girth property. Finally, the proposed code matrix is created by replacing "1" in the above base matrix with square circulant sub-matrices, the offsets of which are searched by a quasi-cyclic (QC)-oriented modified approximate cycle extrinsic message degree algorithm to improve the cycle relationship, especially for the compound cycles. Simulation results show that the codes exhibit excellent performance in both error floor and waterfall region on an additive white Gaussian noise channel. Moreover, they are also characterized by faster encoding due to the QC structure as well as lower decoding complexity and less latency, which make them a natural fit for new-generation power constrained wireless communications.

INDEX TERMS Generalized QC LDPC codes, protograph, modified EXIT, MACE, minimum distance.

I. INTRODUCTION

LDPC codes are Shannon capacity approaching channel codes with moderate complexity. Because the binary LDPC codes are much easy for encoding and decoding, we just discuss the efficient design methods of binary LDPC codes below. Initially, they were constructed with random parity check matrices, which led to the dense generator matrices of high encoding complexity. Richardson *et al.* [1], Richardson and Urbanke [2], and Chung *et al.* [3] had investigated that random LDPC codes, with carefully chosen degree profiles, could approach capacity at a close rate, when decoded by the iterative belief propagation (BP) algorithm. But they were not suited for wireless communications, due to the huge encoding complexities in proportion to the square of their large code length. High error floors were mainly caused by unfavorable combinatorial characteristics, such as smallest cycles, stopping sets or trapping sets [1], [4]. So it was necessary to research the structured codewords for efficient encoding and good performance. Recently, the efficiently encodable LDPC codes, *e.g.* the irregular repeat accumulate (IRA) codes and the accumulate repeat accumulate (ARA) codes, were proposed with high performance and low complexity [4]. The ARA codes were designed efficiently by the protographs [5]. They had relatively small multi-edge graphs, from which larger graphs were generated by a copy-and-permute procedure [6]. They were further extended as the complete Tanner graphs by a modified progressive edge growth (PEG) method [7], [8]. Fang *et al.* [9]

had provided a comprehensive survey on the state-of-the-art in protograph LDPC code design and analysis for different channel conditions, including the AWGN, fading, partial response (PR), and Poisson pulse-position modulation and so on, which greatly facilitated research in this field. And a systematic framework based capacity-approaching protograph-based LDPC coding systems were also investigated in [10] for practical uses, where practically short length codes were designed. Since binary protograph LDPC codes were usually punctured, survived check nodes were allocated evenly to all punctured variable nodes, to prevent the formation of a stopping set, even when the amount of punctures was large [11], [12]. Other alternatives were the quasi-cyclic (QC) LDPC codes based on finite geometries [13]. These codes utilized efficient QC structures with cyclic circulants for both generator and check matrices, all possessing easily addressing and accessing with low complexity [14]. The structures were also adopted in the ARA codes and their parity check node extensions [15]. To improve the error floor, many strategies had been proposed, such as constructing the QC LDPC codes with large girths [16]. But large girth was just the sufficient block of wood prerequisite to improve the error floor, because not all short cycles hurt the performance equally. Richardson [17] analyzed the average error probability over memoryless channels under iterative decoding and showed that small stopping sets, especially the smallest ones, and also the trapping sets, were the main reasons of high error floors. Then, Tian *et al.* [18] explained the cycle sets, stopping sets and small distances and suggested an efficient approximate cycle extrinsic message degree (ACE) algorithm to generate the codes with low error floors. But they didn't consider the compound cycles.

Our work differs from these earlier ones in that we focus on both threshold and error floor, with an irregular QC structure for efficient coding. We mainly consider an alternative protograph based code design, which facilitates the short and moderate, encodable codes with both good threshold and low error floor. The optimized protograph is extended with a simple linear block node as the generalized LDPC (G-LDPC) codes [19]. It is optimized with an EXIT chart for good threshold [20], [21], which is improved as modified extrinsic information transfer (EXIT) chart suited for punctured nodes in code graphs. The floor is also analyzed and optimized by the asymptotic weight distributions as in [22] and [23]. The codeword is then designed with protograph extensions [24] and low error floor is achieved with some repetitions of variabe/check nodes in the Tanner graph to enlarge minimum distance [25], [26]. After matrix extension in successive QC LDPC code construction, the offsets of the proposed circulant sub-matrices are searched by a quasi-cyclic (QC) oriented modified approximate cycle extrinsic message degree (MACE) algorithm to improve the cycle relationships. This measure can add enough external independent links, from outer variable nodes rather than those in the inevitable compound cycles, to increase independence of messages in decoding iterations. By accumulators, the proposed codes are jointly concatenated with the minimum shift keying (MSK) like modulation of memory effect easily to fit the nonlinear amplifier in space communications for high power efficiency [27]. The protograph-based LDPC code design technique can also be used for bandwidth-efficient coded modulation with probabilistic shaping by jointly optimizes the LDPC code node degrees and the mapping of the coded bits to the bit-interleaved coded modulation (BICM) bit-channels [28]. For more practical wireless fading channel applications, rate-compatible root-protograph codes as well as their distributed version, can also obtain full diversity in coded modulation based multi-relay quasi-static fading channels efficiently [29], [30]. For a memory channel, a factor graph based method with both code constraints and channel behaviors is also used for the joint *maximum a posteriori* (MAP) detection [31]. Finally, the main contributions of this paper are concluded as follows.

- **Low complexity generation of code graph template**
  An optimized LDPC code graph template is generated by combining an accumulate-repeat-accumulate seed protograph with a Tanner graph extension of a simple linear block node. The template optimization is much simpler than that of a full code graph, which reduces much complexity of the code construction.
- **Joint optimization of modified EXIT chart and asymptotic weight distribution to deal with punctured nodes for good error floors of LDPC codes**
  The scheme analyzes and optimizes theoretically the code properties, especially with puncture mechanism and the asymptotic weight distribution, by a modified EXIT chart, which can properly deal with the punctured nodes in decoding iterations. Thus, the modified EXIT chart and the asymptotic weight distribution are combined together to obtain an overall optimization of both threshold and error floor for the LDPC codes academically.
- **MACE algorithm to improve compound cycle relationship by introducing outer independent links**
  The offsets of the proposed circulant sub-matrices are searched by a quasi-cyclic (QC) oriented MACE algorithm to improve the cycle relationship, especially for the inevitable compound cycles, by introducing new independent messages from outer variable nodes, rather than those in cycles in decoding iterations. And it can overcome the error propagation phenomenon caused by false messages self-feedback between two variable nodes of the same cycle in decoding iterations.

The remainder of the paper is organized as follows. In section II, the concepts of protograph G-LDPC codes with their graph representations are introduced. In this section, the protographs and the extensions are also described and analyzed with their QC structures. In Section III, three optimization techniques, *i.e.*, the puncture mechanism with EXIT chart for threshold, the asymptotic weight distributions for minimum distance and the MACE for circulant offsets, are discussed to improve the codes. The joint code optimization by them are also depicted. Then, in section IV, some extensions of the protographs with a Hamming node are presented for low complexity and storage occupation. Three families of the optimized G-LDPC codes are generated in detail. Subsequently, numerical simulations and result analyses of the codes by our method on an AWGN channel are given to manifest the good performance in section V. Finally, the summary is concluded in section VI.

II. PROTOGRAPH G-LDPC CODES WITH QC MATRICES

An LDPC code is depicted by a $M \times N$ check matrix $\mathbf{H}$, which stands for a code with rate $R = (N - M)/N$, where N, K and $M = N - K$ are the code length, the information and parity check bits per codeword, respectively. It can be represented by a Tanner graph, where there are m check nodes and n variable nodes related to the rows and columns in $\mathbf{H}$. Generally, $\mathbf{H}$ has irregular row and column weight, and it is characterized by the degree assignment sets, which determine the theoretic threshold of the code. It can be optimized by the density evolution (DE) [2] or EXIT chart [20] and so on.

A protograph is similar to a Tanner graph with relatively small number of nodes, except that they can possess parallel edges, *i.e.*, a variable and check node pair connected by more than one edge. It can be extended to a large Tanner graph by a "copy-and-permute" procedure [6]. ARA codes are a class of protograph LDPC codes. Since the variable node extensions for high code rate are quite simple, we mainly discuss the improvement of the code graph for low rate. The performance of the low rate code is mainly improved by pre-coding of the RA code with a series of differential encoders [5]. So it can be improved by replacing a differential encoder with a more powerful Hamming pre-encoder, *i.e.*, node extension (or node doping) [15], as a *generalized* LDPC (G-LDPC) code. It consists of the substitution of selected single parity-check (SPC) node pairs in an Tanner graph with more powerful linear block constraint nodes. So it is initialized by good threshold and substituted by a fraction of SPC nodes with stronger constraint nodes. The choices of the fraction of the doped nodes and the component codes depend on the desired code rate and block length. And the set of component codes used for doping are limited to the low complex codes. After the node doping process, the derived graph of the G-LDPC code is extended to a complete Tanner graph and thus get the related check matrix with same method of the protograph code. An instance of the code design is shown in Fig. 1.

In Fig. 1, the optimal construction of the protograph G-LDPC code can be generalized as four steps. Firstly, the optimal seed protograph in Fig. 1(a) is obtained either by the simulated annealing (SA) [6] or directly from the existed ones in [5]. Some variable nodes in the protograph are usually punctured for good performance. In the second and most important step, the SPC node pair in the seed protograph is replaced (doped) with a more powerful linear block node and the result is shown in Fig. 1(b). The generated graph has the same degree profile and the edge connection to the final Tanner graph, which affects the performance. In this step, the edge connection between the linear block node and the original protograph can be adjusted according to the theoretic threshold and error floor of the potential code represented by the derived graph. Some new edges can be added in the derived graph to satisfy the optimal degree profile searched by the EXIT chart technique and thus to get good threshold. After that, the protograph are replicated for several copies, *i.e.*, 2 copies in Fig. 1(c), and then the edges are permuted with the same node sequence among the different copies shown in Fig. 1(d). So parallel edges are eliminated and a standard Tanner graph is derived. In the final step, the new derived graph of Fig. 1(d) is re-sequenced and expanded for q fold of the original protograph to get the final Tanner graph. In this procedure, each variable/check node is turned to a set of q nodes, with each edge is changed as a $q \rightarrow q$ permutation connections $\mathbf{\Pi}_{i,j}$ in Fig. 1(e). It inherits the same degree profile of sub-plot (d) and thus has same theoretic threshold. The threshold for infinite code length are computed using the DE or the EXIT technique over the degree profile from the Tanner graph. Given specific number of variable and check nodes, all possible connections between them are searched over to obtain lowest threshold. In summary, this pragmatic approach is less complex thoroughly than the direct DE based search in code design, where only small scale of seed protographs and the extensions need to be found. In addition, it permits the codes with both good thresholds and low error floors.

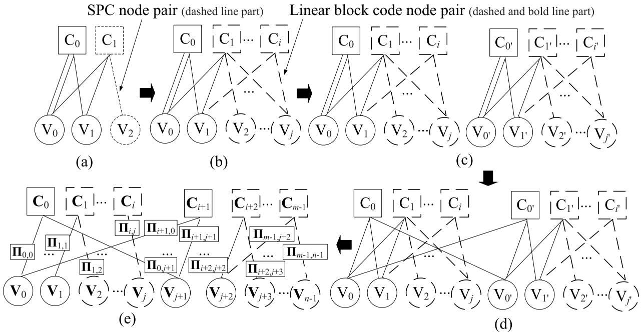


FIGURE 1. G-LDPC Protograph with implied copy and permute expansion. Each permutation matrix is $q \times q$ and each vector node in the final Tanner graph is interpreted to be q nodes of the same type. (a) Seed protograph. (b) Node doping. (c) Copy 2 times. (e) Re-sequence and q-fold extension. (d) Expand (premute and eliminate multi-edges).

Using similar notations in [15], we define $V = \{V_j\}_{j=0}^{n-1}$ be the set of n variable nodes (VNs) and $C = \{C_i\}_{i=0}^{m-1}$ be the set of m check nodes (CNs). So the connections between CNs and VNs in the final Tanner graph of Fig. 1(e) are related to a $m \times n$ base matrix $\mathbf{T}$. The permutation matrices are usually set as the QC form for low coding complexity. Then $\mathbf{T}$ is extended as an adjacency matrix $\mathbf{\Gamma}$ with quasi-cyclic form, where each "1" or "0" in $\mathbf{T}$ is replaced with a cyclic sub-matrix $\mathbf{B}_{i,j}$ or a zero matrix of the same dimension, respectively. The cyclic sub-matrix is usually derived from circular shift of an *identity* matrix. So the relationship between the adjacency matrix $\mathbf{\Gamma}$ and the check matrix $\mathbf{H}$ of a QC LDPC code is determined, once each cyclic sub-matrix $\mathbf{B}_{i,j}$ is confirmed. In this case,

$\mathbf{\Gamma}$ is a $m \times n$ circulant permutation matrix in the form of

$$\Gamma = \begin{bmatrix} \mathbf{B}_{0,0} & \mathbf{B}_{0,1} & \cdots & \mathbf{B}_{0,n-1} \\ \mathbf{B}_{1,0} & \mathbf{B}_{1,1} & \cdots & \mathbf{B}_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{B}_{m-1,0} & \mathbf{B}_{m-1,1} & \cdots & \mathbf{B}_{m-1,n-1} \end{bmatrix}, \tag{1}$$

where $\mathbf{\Pi}_{i,j}$ is either a $q \times q$ circulant permutation matrix or a zero matrix. And the matrix $\mathbf{\Gamma}$ is just generated by substituting the circulant permutation or zero matrices for "1" or "0" in the base matrix, respectively. By this means, short cycles need to be avoided with circulant permutation expansion. $\mathbf{H}$ is an optimized permutation of $\mathbf{\Gamma}$, with the size $mq \times nq$, and it is enlarged by q times in both row and cow. In summary, the check matrix $\mathbf{H}$ of the code can be obtained by the expansion of a base matrix $\mathbf{T}$, and it is in the form of the adjacency matrix $\mathbf{\Gamma}$. It has determined circulant permutation matrix, which need to be optimized for large cycles in the Tanner graph.

In addition, a more randomization measure can be adopted to obtain better performance, where the circulant sub-matrices are divided into much smaller ones randomly. Each none zero square circulant sub-matrix $\mathbf{I}$, *i.e.*, $\mathbf{B}_{i,j}$ in (1), is decomposed as 4×4 (or larger size as 8×8, *etc.*) smaller square circulant matrices $\{\mathbf{I}_i\}$, as long as they guarantee weight-1 of the rows and columns in these sub-matrices $\{\mathbf{B}_{i,j}\}$. Namely, each $\mathbf{B}_{i,j}$ is divided as the smaller 4×4 circulant matrices to increase the randomness of the code for better performance according to Shannon's coding theory [4]. One instance of a 4×4 division is expressed in (2).

$$\mathbf{I} = \begin{bmatrix} \mathbf{I}_1 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I}_2 & 0 \\ 0 & 0 & 0 & \mathbf{I}_3 \\ 0 & \mathbf{I}_4 & 0 & 0 \end{bmatrix} \tag{2}$$

In (2), a sub-matrices $\mathbf{I}$ is divided as a 4×4 sub-sub-matrix $\mathbf{I}_i$ with both row and column weight-1. The detailed position of each $\mathbf{I}_i$ in $\mathbf{I}$ can be obtained by applying the PEG algorithm [7], [8], where each $\{\mathbf{B}_{i,j}\}$ in (2) is replaced by a possible allocated distribution $\{\mathbf{I}_i\}$. And it is written as a new symbol $\mathbf{\Pi}_{i,j}$ in the extended matrix for clear coordination. Therefore, with properly chosen division coefficient t, the final check matrix of the code is expressed as

$$\Gamma = \begin{bmatrix} \Pi_{0,0} & \Pi_{0,1} & \cdots & \Pi_{0,nt-1} \\ \Pi_{1,0} & \Pi_{1,1} & \cdots & \Pi_{1,nt-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Pi_{mt-1,0} & \Pi_{mt-1,1} & \cdots & \Pi_{mt-1,nt-1} \end{bmatrix}. \tag{3}$$

Better performance can be achieved by much higher dimension of divisions, but at the cost of more coding complexity and storage. So the dimension of the division requires tradeoff between performances and resources, as well as the coding complexity and latency.

In summary, the proposed protograph G-LDPC code can be optimally designed in four steps. Step 1). An optimized seed protograph in Fig. 1(a) is searched or chosen in the literature. Step 2). The seed protograph is doped with a more powerful compound node for low rate shown in Fig. 1(b). Also the simple variable node extensions for high rate will be discussed in an example latter. Step 3). The derived graph in Fig. 1(b) is duplicated for several times as in Fig. 1(c) and the parallel edges are permuted and eliminated with the same node sequence among different protograph copies shown in Fig. 1(d). Step 4). The new graph of Fig. 1(d) is re-sequenced and expanded for q fold of the original protograph, with divided QC permutation, to obtain the final Tanner graph. In our work, Step 2) is performed with some edge modifications according to theoretic threshold (by degree profile), error floor (by minimum distance) and Step 4) is operated for optimal error floor (by loop relationship).

III. DESIGN METHOD OF PROTOGRAPH G-LDPC CODES

LDPC codes are mainly evaluated by threshold and error floor. The former is determined by the waterfall region, while the latter is affected by the minimum distance and the loops in Tanner graphs. To design the optimal protograph G-LDPC codes with good threshold and low error floor, they are theoretically analyzed and generated by the EXIT chart for good threshold, as well as the asymptotic weight distribution for low error floor. The QC oriented *modified* ACE (MACE) is also proposed to search the optimal circulant offsets of the QC sub-matrices of finite length codes for low error floor in coding practice. Finally, they are described as follows respectively.

A. CODE OPTIMIZATION BY EXIT CHART AND MINIMUM DISTANCE

The decoding threshold of protograph G-LDPC codes are mainly determined by the optimal degree profiles from the base protographs. Proper edge connections and even some incremental edges in the protograph extension with either the variable node extensions or the linear block node for doping can be partly evaluated for good threshold by the EXIT chart. In this procedure, the punctured and degree-1 variable nodes are also adjusted for the EXIT analysis. So the protograph codes can be analyzed and jointly designed with the asymptotic weight distribution as follows.

1) THRESHOLD ANALYSES AND OPTIMIZATION BY EXIT CHARTS

An LDPC code is characterized by the degree assignment sets $\{d_v(i)\}_{i=1}^{N}$ and $\{d_c(j)\}_{j=1}^{M}$, where $d_v(i)$ is the degree of the i-th variable node and $d_c(j)$ is the degree of the j-th check node [1]. For an LDPC code, the variable and check degree profile, *i.e.*, $\lambda(x)$, $\rho(x)$, which influences the performance of the code, can be depicted as

$$\lambda(x) = \sum\nolimits_{i=2}^{d_v} \lambda_i x^{i-1}, \quad \rho(x) = \sum\nolimits_{i=2}^{d_c} \rho_i x^{i-1}, \tag{4}$$

where d_v and d_c are the maximum variable and check node degree. The degree profile expresses the ratio of edges connected to the variable or check nodes, respectively, where λ_i

is the fraction of the edges connected to the variable nodes of degree i, while ρ_j is related to the check nodes of degree j. It determines the theoretic thresholds of the codes.

Since protograph codes are recognized as a mixture of inner repetition codes and outer check codes, the optimal degree profiles can be searched by an EXIT chart [20] under the constraints of the variable and check nodes rawly fixed by the protograph. With some notations in [20], I_A and I_E are defined as the *a priori* and extrinsic log-likelihood ratios (L-values) of the messages passing through between the variable and check nodes in iterative decoding. For the variable node decoders (VNDs) and the check node decoders (CNDs), there are corresponding definition of $I_{A,V}$, $I_{E,V}$, $I_{A,C}$ and $I_{E,C}$. They are computed via numerical simulations according to the EXIT chart technique to get the thresholds as in [15]. Given average check node degree $\bar{d}_c$, the rate R, the $I_{E,V}$ of the VNDs with multi-degrees can be calculated by the linear combination of $I_{E,V}[d_v(i)]$ and the coefficients are

$$b_i = \lambda_i d_v(i)/[(1-R)\bar{d}_c], \tag{5}$$

where $\bar{d}_c$ is the linear combination of all check degree assignment set $\{d_c(j)\}$ with the fraction coefficients $\{c_i\}$. Given $\bar{d}_c$, the degree profile is searched and the VND curve of $I_{E,V}$ is plotted to match the CND curve for optimal degree profile.

The modifications of the edge connections and some incremental edges in the derived protograph with node extension affect the degree profile (λ_i, ρ_j) of the code, which lead to different threshold calculated by the EXIT chart. So proper edge connections and tentatively incremental edges are allowed in the code protograph design. By this method, a typical example of the mutual information (MI) transfer in a Tanner graph of an LDPC code is shown in Fig. 2. The MI is exchanged between the VNDs of variable nodes and the CNDs of check nodes. Due to good performance brought by the punctured or degree-1 variable nodes in the protograph code, the EXIT is used to fit these two special cases. However, the conventional EXIT chart [20] can't be used to analyze the LDPC codes with either variable nodes of degree-1 or the punctured variable nodes, which are usually appeared in protograph LDPC codes. Similar to the improvement in the protograph EXIT (PEXIT) method in [21], the original EXIT analysis is then modified to overcome such deficiencies. And it is depicted as follows.

- **Punctured variable nodes**
  The punctured variable nodes in an LDPC code deteriorate the decoder, if the punctured portion of the variable nodes involves stopping sets. In the LDPC decoding, a punctured variable node is independent of the channel MI, *i.e.*, I_{ch}, So the channel MI for the punctured node is set as $I_{ch} = 0$ in the MI computation of the EXIT chart.
- **Variable nodes of degree-1**
  In BP algorithm, the MI from a variable node to a check node is the sum among the the channel message and the message from other check nodes other than the target one. For a variable node of degree-1, the output message to the related check node brings the same metric of the channel observation allocated to the node, due to the nature of the BP algorithm. By this rule, the MI from any variable node of degree-1 to the associated check node is always the channel messages. So there is $I_{E,V}^{(1)} = \cdots = I_{E,V}^{(r)} = I_{ch}$, where r is the iteration number. The fraction of variable nodes with degree-1 in an LDPC code implies that there is a term in the calculation of total MI with the item $\lambda_i \cdot I_{ch}$. Since I_{ch} is lower than 1 for any binary-input AWGN channel, the average MI of the variable nodes can not achieve the maximum value, *i.e.*, $I_{E,V} = 1$, as described in [21]. And the EXIT analysis related to variable nodes of degree-1 is evaluated by the MI between the *a posteriori* probability estimation produced from each variable node and the codeword symbols.

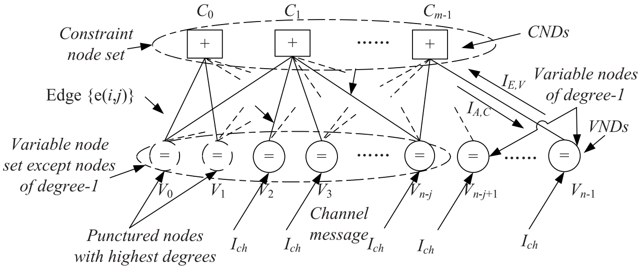


FIGURE 2. Mutual information transfer in a Tanner graph of an LDPC code.

From above analysis, the MI transfer in the variable node of degree-1 is unidirectional and only the MI transfer from the degree-1 variable node to the check node is alowed. So the MI transfer of the whole code will not be convergence to 1 ultimately, which can't be used to analyze the convergence of decoding. However, the MI transfer equilibrium can be persisted between two VND and CND sub-system of bidirectional MI exchange, which contributes the all MI transfer in the iterative decoding. And the unidirectional MI of the degree-1 variable node, *i.e.*, I_{ch}, can just be treated as the *a prior* information for the related check nodes, similar to that of the channel messages. Therefore, the classic EXIT chart method is modified to fit such case. As shown in Fig. 2, the MI transfer is only carried out between the CNDs with the constraint node set, and the VNDs with the variable node set excluding the nodes of degree-1. And the MI in the VNDs of the variable nodes of degree-1, *i.e.*, I_{ch} is set as the initial *a prior* information for the check nodes, which connect the degree-1 variable nodes. Finally, the modified EXIT chart algorithm for an LDPC code with variable nodes of degree-1 on an AWGN channel are given as follows.

- **Initialization: *a prior* information from channel**
  Initialize the channel input $I_{ch} = J(\sigma_{ch})$, with

  $$\sigma_{ch}^2 = 8R \cdot E_b/N_0, \tag{6}$$

  where R is the code rate of the protograph code and E_b/N_0 represents the bit signal-to-noise ratio associated to the channel input to the variable node. If the variable node is punctured, there is $I_{ch} = 0$.

- **MI update(V→C): from variable to check** For variable node order $j = 0, \cdots, N-s$ and check node order $i = 0, \cdots, M-1$, there is

$$I_{E,V}(i,j) = J\left(\sqrt{\sum_{s=1,s\neq i}^{d_v} a_{s,j}(J^{-1}(I_{A,V}(s,j)))^2 + (J^{-1}(I_{ch}))^2}\right), \tag{7}$$

where $N, M, s, a_{s,j}$ are the code length, check node number, number of the punctured nodes and the fraction of the edges from variable node j (excluding the punctured nodes) to check node i, respectively. The $I_{A,V}(s,j)$ is got from the previous iterations, which is the MI $I_{E,C}(s,j)$ from the related check node. If $a_{i,j} = 0$, $I_{E,V}(i,j) = 0$. Then in each iterations, the MI from variable nodes to check nodes are

$$I_{E,V} = \sum\nolimits_{k=2}^{d_v} b_k I_{E,V}(i,j)^{(k)}, \tag{8}$$

where b_k is the percentage of the variable nodes of degree-k in all variable node degrees and $I_{E,V}(i,j)^{(k)}$ is the average MI from variable nodes of degree-k.

- **MI update(C→V): from check to variable** With the same definition of i, j, there is

$$I_{E,C}(i,j) = 1 - J\left(\sqrt{\sum_{s=1,s\neq j}^{d_c} a_{i,s}(J^{-1}(1 - I_{A,C}(i,s)))^2}\right), \tag{9}$$

where $N, M, s, a_{i,s}$ are the same in MI Update(V→C). The $I_{A,C}(s,j)$ is got from the previous iterations, which is also the MI $I_{E,V}(s,j)$ from the related variable node. If $a_{i,j} = 0$, $I_{E,V}(i,j) = 0$. Then in each iterations, the MI from check nodes to variable nodes are

$$I_{E,C} = \sum\nolimits_{k=2}^{d_v} c_k I_{E,C}(i,j)^{(k)}, \tag{10}$$

where c_k is the percentage of the check nodes of degree-k in all check node degrees and $I_{E,C}(i,j)^{(k)}$ is the average MI from check nodes of degree-k.

- **Iterate the two MI update until** $I_{E,V} \approx 1$.

In the above algorithm, we also perform the MI updates of variable nodes with degree-1 in the decoding, which guarantees the converge of MI to an exact 1 in theory. In addition, to verify the modified EXIT chart, we give an example of the analysis of a typical protograph G-LDPC code. Consider an instance of a rate $R = 1/6$ G-LDPC code ensemble with the base matrix, whose degree profiles are $\lambda(x) = 0.2 + 0.1x + 0.25x^4 + 0.45x^8$ and $\rho(x) = 0.6x^2 + 0.4x^3$. According to the Modified EXIT chart method, the EXIT curve is plotted in Fig. 3.

From Fig. 3, the optimal threshold for the code is -0.813dB, which is very closed to that of -0.813dB by the DE and a little better than that of -0.791dB by the PEXIT [21]. Since the theoretical threshold for a code of rate 1/6 is -1.07. there is still a gap of 0.257dB between our result and the true threshold. So there is still effort to search the more optimized code degree profile for better threshold.

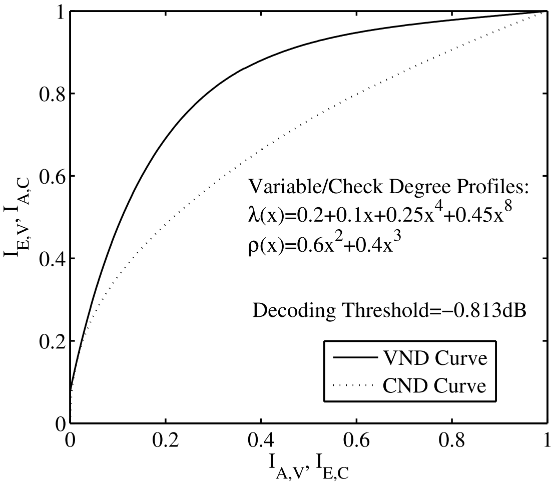


FIGURE 3. EXIT chart for a typical protograph LDPC code.

Therefore, the proper edge connections and incremental edges in the derived protograph with node extension are set in the design of the code for good threshold calculated by the EXIT chart. The modified EXIT analysis provides a much simpler measure for the protograph LDPC code threshold evaluation, when compared with the DE and GA method. The reduction in computational complexity is due to the nature of the EXIT chart, which only tracks a single parameter, *i.e.*, the MI, rather than the whole probability distribution or the probability mean in the DE [1], or the Gaussian Approximation (GA) [3] approach, respectively. So the EXIT chart greatly improve the efficiency of threshold evaluation and effectively used in code design.

B. OPTIMIZED CIRCULANT OFFSETS BY THE MACE ALGORITHM

BP algorithm possesses good feature of maximum likelihood decoding for LDPC codes, on condition that there is no loop in the Tanner graph. This is mainly due to the iterative decoding, while loops in a Tanner graph cause self-feedback message propagation and thus deteriorate the performance. However, with limited code length, an LDPC code can't be constructed without any loop. So in the final code check matrix in (3), the position of the non-zero sub-matrix $\mathbf{\Pi}_{i,j}$ can be decided by the existing PEG algorithm to get the good loop length properties. Or a sub-optimal code design is considered that there are less cycles or inevitable loops with large circumferences (*i.e.*, the number of edges in a closed loop) in the Tanner graph. The notation girth is the smallest circumference of all loops in a Tanner graph, and it is given in literature to evaluate code performance [4]. It is one of the key factor in optimal code design.

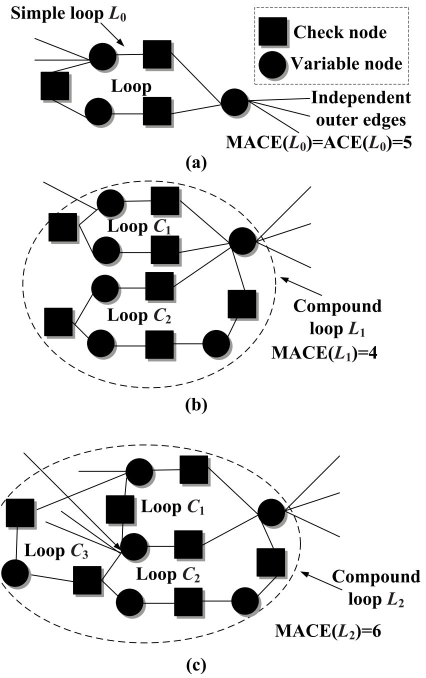


FIGURE 4. Typical compound loops with *MACE*. (a) Original ACE Algorithm only considers simple loops. (b) Two loops with 1 common node. (c) Two loops with 1 common edge.

TABLE 1. The diagram of the MACE algorithm.

Initializations: $m \times n$ sub-matrixes for searching the QC offsets are required. For all variable and check nodes, $p(\mu_t)$ is the *MACE* between any node μ_t and the root node v_0 in the tree search of the code. When μ_t is a variable node, $MACE(\mu_t)$ is the degree of μ_t minus 2. Otherwise $MACE(\mu_t)$ is 0.

Main Procedure:

```
for ($i$ =$n$-1;$i \leq$ 0; $i$- -)
begin
for ($j$=0; $j \leq m$-1; $j$++)
  begin
  redo:
  Randomly generate $v_i$ according to the non-zero offset positioning in the first row of the QC sub-matrix $\Gamma$ which is not a zero matrix.
  MACE detection of $v_i$:
    $p(\mu_t) \leftarrow \infty$ for all variable and check nodes.
    $p(v_i) \leftarrow MACE(v_i)$ for all variable and check nodes.
    Find possible simple & compound loops in current structure.
     Loops are searched by the tree spread of a Tanner graph. Given one variable node as the root, track the connected edges from it to any possible nodes by the Depth-First-Search graph algorithms [32] and record the node set once there is a loop. If there are different node loop sets with the same root node, the sets belong to the same compound loop, otherwise they are single loops. Common nodes in a compound loop are marked for successive procedure.
    For simple loops:$MACE(v_i) = ACE(v_i) = d_i - 2$. $MACE(c_i) = 0$. The MACE of a length $2d$ loop is $\sum_i (d_i - 2)$. $v_i$ and $c_i$ are the $i$-th variable and check nodes, respectively.
    For compound loops: Suppose the check node set of the compound loop as $C(v_i)$, in which all check nodes are connected to the same variable node $v_i$ in the common edges of the compound loops. The number of $C(v_i)$ is represented as $N(C(v_i))$. So the MACE of the compound loop is calculated as $\sum_i (d_i - 2) - \sum_k [N(C(v_k)) - 2]$.
   for ($k$ =1; $k \leq d_{ACE}$; $k$++)
     begin
      for any active node $w_s$ in level $(k-1)$.
        begin
        Find its children set $Ch(w_s)$.
        for every child $\mu_t \in Ch(w_s)$.
          begin
          $p_{temp} \leftarrow p(w_s) + MACE(\mu_t)$.
          $s_{temp} = p_{temp} + p(\mu_t) - MACE(v_i) - MACE(\mu_t)$.
          if ($s_{temp} < \eta_{ACE}$)
              Exit with failure.
          else if ($p_{temp} \geq p(\mu_t)$)
              Deactivate $\mu_t$ in level-$k$ with current parent $w_s$.
          else
              $p(\mu_t) \leftarrow p_{temp}$
          end
          end
        end
        Exit with success.
        if ($ACE < \eta_{ACE}$ for a cycle of length $2d_{ACE}$ or less)
          goto redo.
        end
      end
   end
end
```

QC LDPC codes usually can't avoid all loops [4]. Fortunately, the relationships of the loops are also crucial to the code performance, since independent nodes connected to a loop introduce reliable and independent messages from external nodes outside of the loop [18]. Following this criterion, the optimal circulant offsets of the non-zero sub-matrix $\mathbf{\Pi}_{i,j}$ in (3) are searched by the ACE algorithm [18] under the constraints of the degree profiles and the QC framework. But in the original ACE algorithm, there are only single loops as shown in Fig. 4(a). But compound loops, as shown in Fig. 4(b)&Fig. 4(c), need accurate calculation of real ACE values. Here the *compound loop* is a group of loops with common edges or nodes. The real ACE value for a variable node, involved in several loops, should be corrected by subtracting the number of the edges in the compound loop. Only the outer independent edges contribute to the effective and reliable messages in the iterative decoding.

Therefore, the offsets in the sub-matrices of the QC LDPC codes are searched by the new QC oriented *MACE* algorithm. At first, there are some notations [18] as follows.

EMD: An extrinsic constraint node of a variable-node set is a constraint node singly connected to the set. The EMD of a variable node set is the number of extrinsic constraint nodes.

Approximate Cycle EMD (ACE): The ACE of a length $2d$ loop is $\sum_i (d_i - 2)$, where d_i is the degree of the i-th variable of the cycle. The ACE of a degree-d variable node is $(d-2)$, and the ACE of any constraint node is 0.

Modified ACE (MACE): The MACE of a loop is the independent connections from outer compound loops. It is calculated by the ACE value subtracting the number of the edges within the compound loops, and the *MACE* of any constraint node is 0 too.

Finally, the MACE algorithm is proposed in Tab. 1. The optimal circulant offsets of the non-zero sub-matrix $\mathbf{\Pi}_{i,j}$ in (3) are found by the MACE algorithm and the parameter (d_{ACE}, η_{ACE}) is optimally selected by multiple tentative experiments with accurate ACE values under compound loops.

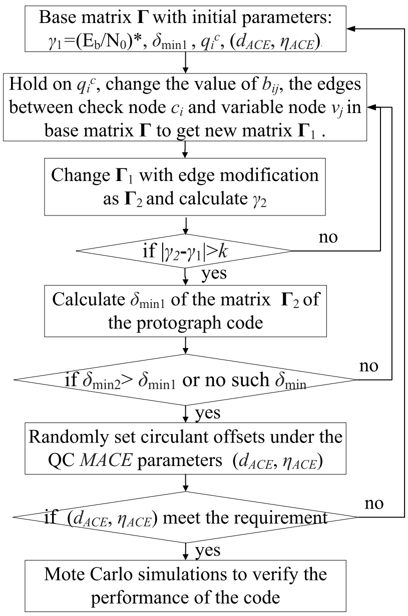

Base matrix Γ with initial parameters: γ_1=(Eb/N0)*, δ_{min1}, q_i^c, (d_{ACE}, η_{ACE})
Hold on q_i^c, change the value of b_{ij}, the edges between check node c_i and variable node v_j in base matrix Γ to get new matrix Γ_1 .
Change Γ_1 with edge modification as Γ_2 and calculate γ_2
if $|\gamma_2$-$\gamma_1|>k$
no
yes
Calculate δ_{min1} of the matrix Γ_2 of the protograph code
if δ_{min2}> δ_{min1} or no such δ_{min}
no
yes
Randomly set circulant offsets under the QC MACE parameters (d_{ACE}, η_{ACE})
if (d_{ACE}, η_{ACE}) meet the requirement
no
yes
Mote Carlo simulations to verify the performance of the code

FIGURE 5. Block diagram of a joint optimized construction of an LDPC code.

TABLE 2. The detailed steps of the joint construction.

Main Procedure of the Jointly Optimal Code Construction.
Step 1). Given a $m \times n$ base matrix Γ, where b_{ij} is the number of the edges between the check node (CN) c_i and variable node (VN) v_i. Suppose the pre-coding flag as l_p, which is equal to 1 when there is the pre-coding structure in the protograph corresponding to matrix Γ. And the 1st row and cow of matrix Γ is fixed to keep the pre-coding gain. Namely, the edge between CN c_1 and VN v_1 is fixed. The *ACE* parameter pair (d_{ACE}, η_{ACE}) is also initialized set here according to similar coding experience of the same code length. Then, if l_p is equal to 1, goto step 2), otherwise goto step 3).
Step 2). Set the degree of CN c_2 and c_3, *i.e.*, d_2^c and d_3^c, in Γ, as q_2^c and q_3^c, or $q_2^c + 1$ and $q_3^c - 1$, respectively, where the degree of every check node is kept unchanged. Then randomly choose the value of b_{ij} to get Γ_1, given that $\sum_{j=1}^{n} b_{ij} = d_i^c$. Then, goto step 4).
Step 3). Set $d_1^c = q_1^c$, $d_2^c = q_2^c$, $d_3^c = q_3^c$. Choose b_{ij} according to the method in step 2) and get Γ_2. Then goto step 4).
Step 4). Calculate the threshold γ_2 of Γ_2. If $|\gamma_2 - \gamma_1| < k$, goto Step 5), else if $l_p = 1$, goto Step 2) and else if $l_p = 0$, goto Step 3).
Step 5). Calculate the minimum distance ratio δ_{min2} of the matrix Γ_2. If $\delta_{min2} > \delta_{min1}$ or δ_{min2} is not exist, simulate the performance of Γ_2 by the *EXIT* chart and goto Step 6). Otherwise, if $l_p = 1$, goto Step 2), else if $l_p = 0$, goto Step 3).
Step 6). Given the tentative QC MACE parameter (d_{ACE}, η_{ACE}), calculate the circulant offsets for every QC sub-matrix. When (d_{ACE}, η_{ACE}) is not met, goto Step 1) with degraded QC ACE parameter pair, undergo the above steps, until all circulant offsets are searched and satisfied for the QC ACE parameters. Finally, the joint algorithm is over and the codeword is sent for BP decoding.

C. THE JOINT CODE OPTIMIZATION CONSTRUCTION ALGORITHM

The minimum distance analysis of an LDPC code is also a key factor in the code design. It can be evaluated by the asymptotic weight distribution in the optimal code protograph design theoretically for low error floor, which has been discussed and used for optimal protograph design in [26]. Based on the the modified EXIT chart, the asymptotic weight distribution and the MACE techniques mentioned above, a jointly optimized protograph LDPC code construction is proposed for good threshold and low error floor. According to the protograph principle, an optimization of a small scale protograph template leads to good performance as well as reduced construction complexity. And an optimized protograph template of an ARA code can be the proper seed protograph, which can be extended to a high or low rate by increasing variable nodes or doping check nodes. Then a base matrix $\mathbf{\Gamma}$ with $m \times n$ circulant matrices are designed under the parameters of the iterative threshold γ, *i.e.*, $(E_b/N_0)^*$, check node degree (*prior*: q_i^c and *posterior*: d_i^c), the typical minimum distance ratio (δ_{min}) and the MACE parameter (d_{ACE}, η_{ACE}). The notations or variables "γ, *i.e.*, $(E_b/N_0)^*$, check node degree (*prior*: q_i^c and *posterior*: d_i^c), the typical minimum distance ratio (δ_{min})" are defined and calculated as in [26]. Finally, the block diagram of the joint construction of a protograph LDPC code is shown in Fig. 5 and depicted in Tab. 2.

In the joint code design algorithm, k is a factor to adjust the threshold, which is a positive constant. For large k, it is easy to find the code with better threshold (waterfall region) than that of the original code. But it usually has worse error floor, which in turn increases the complexity of the iterative search. To another extreme, small k also leads to poor threshold and worse error floor. So k should be compromised for both the error floor and the threshold, set as 0.4 dB and so on, through some tentative simulations.

IV. THREE IMPROVED PROTOGRAPH G-LDPC CODES

To construct a protograph code with good threshold and low error floor, the optimal construction of the seed protograph and the doped node, as well as the edge connections are crucial. They can be accomplished by a joint optimization of the EXIT chart, the asymptotic weight analysis and the MACE techniques mentioned above. ARA codes are characterized by good thresholds and low error floor, as well as concise graph representations. So one of them is adopted here as a seed protograph, which is doped with a linear block code and then splitted and subsequently permuted with QC form for low coding complex. Here we mainly introduce 3 families of the optimal protograph QC G-LDPC codes (short for G-LDPC codes) with simple edge modification (code type G^I), doping extension (code type G^{II}), and doping extension with edge amendment (code type G^{III}), respectively. The former one has high rate ($R \geq 1/2$), while the latter two have low rate ($R < 1/2$).

G-LDPC Code Type G^I (simple edge modification): for this code, the edges in the optimized ARA protographs (the seed protographs) are modified under the prediction of an EXIT chart for better threshold than that of the original protograph. We mainly modify the seed protograph by increasing or eliminating several connections to form a much closed relationship of message propagations, thus it also increase the randomness of the code. The seed protograph here is chosen

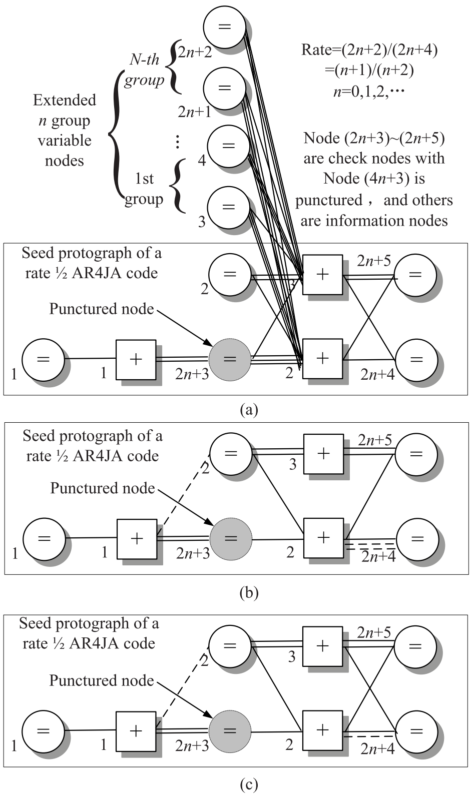


FIGURE 6. Protograph of a G-LDPC Code G^I with simple edge modification. (a) Original AR4JA protograph. (b) modified AR4JA seed protograph: G^I (A) code. (c) modified AR4JA seed protograph: G^I (B) code.

from the famous AR4JA code in CCSDS standard [34]. Some possible tentative edge modifications are made as in Fig. 6, along with the thresholds calculated by the EXIT chart technique. In Fig. 6(a), the original AR4JA protograph [34] is given with the seed protograph of a rate 1/2 AR4JA code. Others being the same (extension of code rate by additional variable nodes in the top of Fig. 6(a).), the modification mainly lies in the seed protograph. From Fig. 6(b)&Fig. 6(c), edges are modified as two type, *i.e.*, G^I Type A (short for $G^I(A)$) and G^I Type B (short for $G^I(B)$), respectively. Some modified edges are also displayed in dashed lines pointed out in the seed protographs, where they are tentatively reversed and testified by the EXIT chart for good threshold.

Finally, the corresponding thresholds are calculated by the EXIT technique and shown in Tab. 3. From it, the thresholds of the proposed G-LDPC codes are a little better than that of the AR4JA code family. However, it is just the theoretic results with infinite code length. In practices, the thresholds are affected by the factors of code length, cycles in the Tanner graph and so on. So the pragmatic codes need to be constructed with all factors considered.

G-LDPC Code Type G^{II} (doping extension): for this code, a SPC node pair of the optimized rate-1/3 ARA protograph (the seed protograph, *threshold* $= -0.048dB$ [5]) is doped by a more powerful linear block node, *i.e.*, a Hamming node, under the analysis of an EXIT chart for low threshold.

TABLE 3. The thresholds of the G-LDPC codes G^I (dB).

Code Type	*Threshold*	*Capacity*	*Gap*
$G^I(A)$ 1/2	0.365	0.187	0.178
$G^I(A)$ 2/3	1.269	1.059	0.210
$G^I(A)$ 4/5	2.334	2.040	0.294
$G^I(B)$ 1/2	0.369	0.187	0.182
$G^I(B)$ 2/3	1.266	1.059	0.213
$G^I(B)$ 4/5	2.331	2.040	0.298

The code construction is similar to that in [15] and is given as an instance latter. However, the doping techniques can be improved for better threshold, where a better Hamming node is used to extend the seed protograph. The new doping is done by the same seed protograph and a different Hamming node H(7,4,3). Here, H(7,4,3) stands for a Hamming code with code length 7, information length 4 and check bit length 3 per codeword, respectively. The corresponding parity check matrix is listed in (11).

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Similar to [15], we retain the order of the information and check node to generate a punctured check matrix, where the information variable node with largest degree is punctured. The resulting protograph of the G-LDPC code G^{II} is shown in Fig. 7. In Fig. 7(a), the SPC node pair in the seed protograph of the optimized rate-1/3 ARA code is doped by the Hamming node H(7,4,3) just like in [15]. By this mean, the rate of the protograph drops to rate-1/6. Higher code rates can be achieved by puncturing the variable nodes with smallest degree in the doped protograph. Namely, the variable node $p_2 - p_5$ with degree-1 can be punctured for high rate. For a much higher rate 1/5 or 1/4 code, either the degree-1 variable node p_5 or the node set p_5, p_4 is punctured. This treatment reduces the degree-1 variable node for high rate, and improves the error floor, which is easily affected by the degree-1 nodes [15]. For example, the code rate 1/4 for the protograph is generated by puncturing any one of remain variable nodes with degree-1 for the ease of eliminating the multi-edges. Besides, other low rate code can also be constructed with the combination of the different optimized ARA code protograph and other Hamming nodes (e.g. H(15,7,8) and so on) including their shortened form. Then, take the rate 1/6 code for example, the protograph can be expanded with 2 copies to eliminate the multi-edges as the derived graph, which is shown in Fig. 7(b).

In Fig. 7(b), the variable nodes of sequence "1" and "2" are information bit nodes and others are check bit nodes. And the code can be extended to a generalized quasi-cyclic matrix by replace all "1" with square circulant matrix and all "0" with square zero matrix with the same dimension just as in (1). By the way, the punctured node with large degree is assigned to the information bit node and the

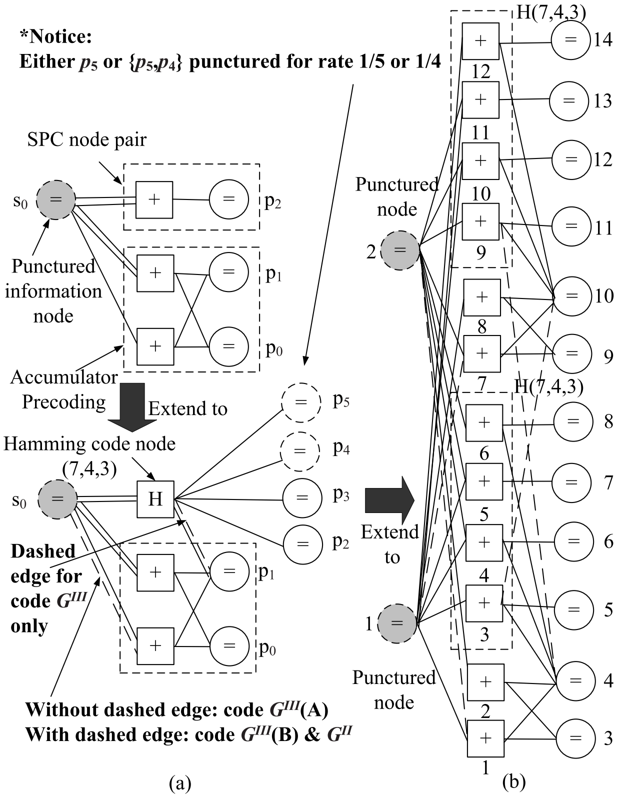

*Notice:
Either p_5 or $\{p_5, p_4\}$ punctured for rate 1/5 or 1/4
SPC node pair
s_0
p_2
p_1
p_0
Punctured information node
Accumulator Precoding
Extend to
Hamming code node (7,4,3)
H
p_5
p_4
p_3
p_2
p_1
p_0
Dashed edge for code G^{III} only
Without dashed edge: code G^{III}(A)
With dashed edge: code G^{III}(B) & G^{II}
(a)
H(7,4,3)
Punctured node
Extend to
Punctured node
1
2
3
4
5
6
7
8
9
10
11
12
13
14
H(7,4,3)
(b)

FIGURE 7. Protograph of G-LDPC Code G^{II} & G^{III} with rate 1/3 seed ARA protograph doped by a Hamming node and the derived graph. (a) Protograph of G-LDPC Code G^{II} & G^{III}. (b) Dervied graph of G^{II} & G^{III}.

rate-1/3 AR4A code [5] is selected for good performance. However, the designed code meets the problems of rank deficiency in the matrix of the protograph, which requires complex measures [14] to finish encoding. So a smallest divided square sub-sub-matrix in the linear dependent block sub-matrix vector can be eliminated to obtain the full check matrix rank only affecting the performance trivially. Because very limited variable and check nodes are influenced due to the small dimension of the sub-sub-matrix and it has been verified by several numerical simulations. Then the code is produced in the form of the standard QC form with full matrix rank, which simplifies both encoding and decoding since full QC form is more regular and easy to be implemented. Then, the offsets in the sub-matrix are searched optimally by the proposed *MACE* algorithm to obtain good loop relationship for better error floor.

Finally, the thresholds for three code rate are calculated by the modified EXIT chart and shown in Tab. 4. From Tab. 4, the theoretic thresholds of the proposed G-LDPC Codes G^{II} are slightly better than that of the G-LDPC code family C^{I} & C^{II} of [33]. Also, the practical thresholds are affected by the factors of code length and so on.

G-LDPC Code Type G^{III} (doping extension with edge modification): for this code, similar measures of generating the code for the G-LDPC Code type G^{III} are took except that there is double edges between the compound node **H** and one node p_1 shown in Fig. 7(a). This modification improves the precoding effect from variable node p_1. In the code G^{III} design, the same seed protograph, doped Hamming code H(7,4,3) and rate increase method are adopted too. Then the thresholds are calculated by the modified EXIT chart and shown in Tab. 5. From Tab. 5, the thresholds of the proposed G-LDPC code type G^{III} are slightly better than that of the G-LDPC code type C^{I} & C^{II} in [33].

TABLE 4. Thresholds of the G-LDPC codes G^{II} (dB).

Code Type	*Threshold*	*Capacity*	*Gap*
G^{II} 1/6	-0.29	-1.07	0.78
G^{II} 1/5	-0.53	-0.96	0.43

TABLE 5. Thresholds of the G-LDPC codes G^{III} (dB).

Code Type	*Threshold*	*Capacity*	*Gap*
$G^{III}(A)$ 1/6	-0.22	-1.07	0.85
$G^{III}(A)$ 1/5	-0.31	-0.96	0.65
$G^{III}(B)$ 1/6	-0.26	-1.07	0.81
$G^{III}(B)$ 1/5	-0.50	-0.96	0.46

V. NUMERICAL SIMULATIONS AND RESULT ANALYSES

In experiments, the LDPC coded BPSK systems in an AWGN channel are simulated. Three proposed G-LDPC codes are generated with modification in their seed protograph and edges in Section IV. The contrast codes are original AR4JA codes in CCSDS standard [34], the G-LDPC codes in [15] and some short LDPC codes in [9] and [36]–[38]. Proper code length and rate are chosen according to the existed contrast codewords. Then, Mote Carlo bit-error-rate (BER) simulations are performed and they run until either a specified number (*i.e.*, 20 in high SNRs and 80 in low SNRs) of error frames occur or a total 6 million trials have been run. Simultaneously, the maximum iterations are 100 for each decoding. Finally, the BER of the codes by proposed method, with their counterparts, are simulated and shown as follows.

A. G-LDPC CODE TYPE G^{I}

In this case, the codes with information length $K = 1024$ and the rate 1/2, 2/3 and 4/5 are chosen and simulated in comparison with the AR4JA code. The ACE parameters (d_{ACE}, η_{ACE}) of the above three codes with rate 1/2, 2/3, 4/5 are (8,4), (6,3), (6,3)(for both $G^{I}(A)$&$G^{I}(B)$), respectively. The BER is shown in Fig. 8. The codes by the proposed method gain a little performance improvement in both good threshold (about 0.05-0.1dB at BER of 10^{-6}) and low error floor, when compared with that of the AR4JA codes in [34].

From Fig. 8, with code rate larger than or equal to 1/2, the generated codes all obtain slightly gain with improving the edge connections by the proposed joint construction method. It is caused by not only the design of the protograph with better theoretic threshold, but also by minimum distance optimization with asymptotic weight analyses. Moreover, the circulant offsets of the QC structure of the codes are

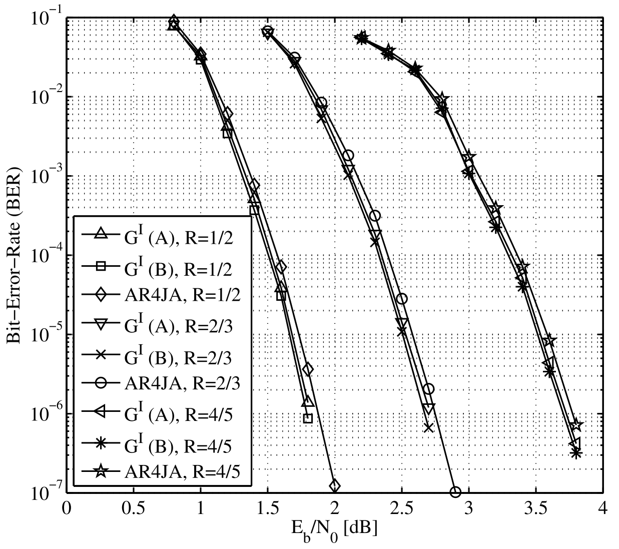


FIGURE 8. BER of the codes ($G^I(A)$&$G^I(B)$ vs AR4JA codes [34]) with information length $K = 1024$, and rate 1/2, 2/3 and 4/5.

also optimized with the MACE algorithm, where more outer independent connections of the inevitable cycles in the Tanner graphs are introduced to obtain self-correction. So it overcomes the cyclic message self-propagations, which improves both thresholds and error floors.

B. G-LDPC CODE TYPE G^{II}

In this case, the codes with information length {1024,1784 (only for rate 1/6 code)}, rate {1/6, 1/5} are chosen and simulated for comparison with the G-LDPC codes in [15]. The code with information bit length 1784 is designed and compared with that of the Turbo code in [35]. The ACE parameters of the two rate {1/6, 1/5} codes are {(8,4), (6,3)}, respectively. Then, the BER is shown in Fig. 9, where the codes constructed by the proposed method obtain a little gain (about 0.15dB at BER of 10^{-6} for rate 1/6, 0.03dB at BER of 10^{-6} for rate 1/5) for both good threshold and low error floor, when compared with that of the codes in [15]. The proposed codes also outperform the Turbo code in [35] due to the poor error floor of the latter.

From Fig. 9, with code rate less than 1/2, the designed codes achieve slightly gain by replacing the shorted Hamming node H(6,3,3) with a complete Hamming node H(7,4,3) in the node doping. The proposed code performs better than the Turbo code at high SNR region, since the latter has obvious error floor effect. The proposed codes are provided with both good thresholds and low error floors, because they are designed with the EXIT techniques in their seed protograph for better thresholds, and the asymptotic weight analyses for better minimum distance. Furthermore, the circulant offsets of the sub-QC matrices of the codes are also optimized with the MACE algorithm, where the stop sets are improved to overcome the interdependent cyclic message self-propagations for both good thresholds and error floors.

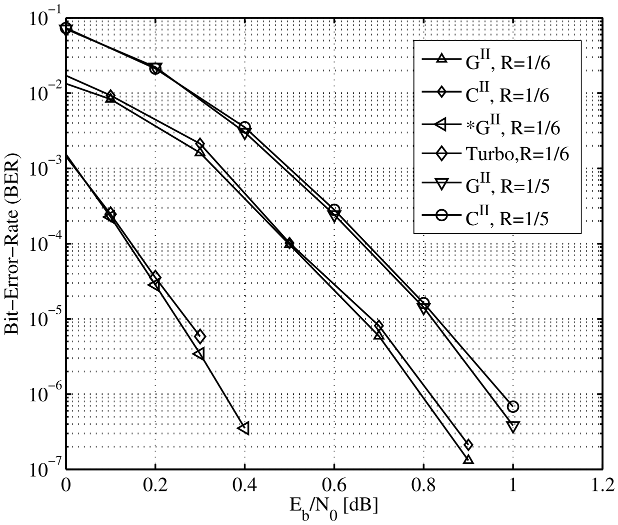


FIGURE 9. BER of the codes (G^{II} vs C^{II} [15] and the Turbo code [35]) with rate 1/6 and 1/5. The information length K=1784 of the $*G^{II}$ code is the same to the Turbo code, while all others are 1024.

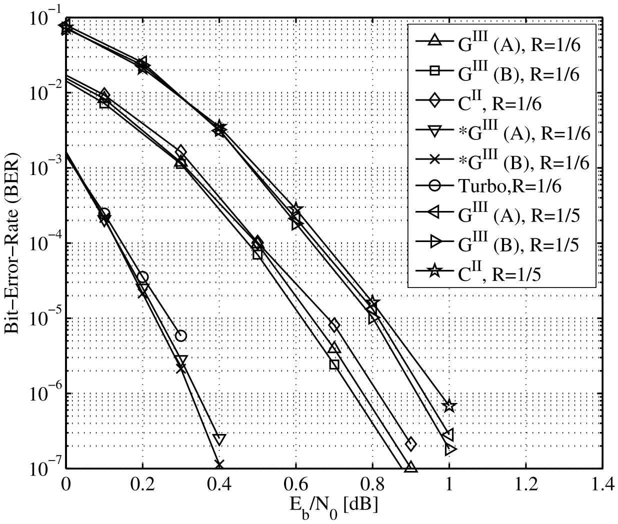


FIGURE 10. BER of the codes (G^{III} vs C^{II} [15] and the Turbo code [35]) with rate 1/6 and 1/5. The information bit length K=1784 of the $*G^{III}(A)$& $*$ $G^{III}(B)$ codes is the same of the Turbo code, while all others are 1024.

C. G-LDPC CODE TYPE G^{III}

In this case, the simulation parameters of the codes for comparison are the same to that in the above part "G-LDPC Code Type G^{II}". The ACE parameters of the above two rate {{1/6, 1/5} codes are {(6,3), (6,3) }(for $G^{III}(A)$), {(8,4), (6,3) }(for $G^{III}(B)$)}, respectively. The BER is shown in Fig. 10, where the codes designed by the proposed scheme obtain a little better (about 0.2dB at BER of 10^{-6} for rate 1/6, 0.05dB at BER of 10^{-6} for rate 1/5) in low error floor, when compared with that of the QC G-LDPC codes in [15]. But they still outperform the Turbo code [35] a little in both thresholds and error floors.

From Fig. 10, the proposed codes obtain a little better in error floor than that of the C^{II} G-LDPC codes [15].

This is mainly due to the structure of the edge relationship with replacement of degree-1 variable node with degree-1 variable node in the Hamming complex nodes shown in the dashed line of Fig. 7. It also can be explained with minimum distance that more repetition rate of a variable node leads to a much larger minimum distance, which helps improve the error correction. The code also outperforms the Turbo code in error floor at high SNR, since the latter has unavoidable poor error floor in natural, which is caused by small minimum distance due to the structure of two convolutional component code with an interleaver between them. In addition, the codes gain better error floor than that of the G^{II} G-LDPC codes, since more high degrees are made by the new joint connection between the complex Hamming node and the accumulator node to improve the asymptotic weight distribution. To sum up, the main reason of their good performance lies in the fact of the enhanced minimum distance. They are optimized with both good thresholds and low error floors by the EXIT techniques in their seed protograph for good thresholds, and with asymptotic weight analyses for better minimum distances. Moreover, the circulant offsets of the QC sub-matrices of the codes are optimized with the MACE algorithm, to improve the stop sets (or trap sets) to eliminate the self-propagations of the interdependent cyclic messages. Therefore, the proposed codes achieve both good thresholds and low error floors.

D. SHORT CODES

Since compound cycles easily appear in short codes, the short QC LDPC codes are simulated and compared with current QC LDPC codes. In these simulations, the proposed codes are $G^{I}(A)$ for high rate and $G^{II}\&G^{III}$ for low rate. The contrast QC LDPC code (code length=1000, rate=4/5) [36] and the code (800, 1/4) [37] are chosen for comparison. The BER is shown in Fig. 11, where the codes designed by the proposed scheme obtain a little gains (about 0.1 dB at BER of 10^{-6} for rate 4/5, 0.2-0.3 dB at BER of 10^{-4} for rate 1/4) in low error floor, when compared with the counterparts in [36] and [37].

From Fig. 11, the proposed protograph G-LDPC codes G^{I} and G^{III} all obtain a little improvement in error floor than that of the QC-LDPC codes [36], [37]. This is mainly due to the fact that it is hard to remove all cycles in short codes. The compound loops are inevitable to occur especially in codes with short length. Also the counterparts have not been considered the relationship of loops, *i.e.*, the ACE affection, which is also the key factor of performance. In addition, the puncture contribute to the performance improvement too by virtually increasing the code constraints in decoding. So the proposed code design can get better performance for the structured QC LDPC codes, especially with short code length.

The proposed codes are also compared with currently well designed protograph codes. In the simulations, the proposed codes are $G^{I}(A)$ for high rate and $G^{III}(B)$ for low rate, which possess rather excellent performance by the proposed scheme. The contrast new rate compatible protograph LDPC codes (short for newRC codes) with information length K=1000, rate={1/3, 2/5, 1/2, 2/3, 4/5} [38], and the best rate-1/2 optimized protograph code (short for CVN1 code) with K=1000 and VN degrees at least 1 [9] are chosen for comparison. The BER is obtained by numerical simulations in Fig. 12, where the codes designed by the proposed scheme obtain a little gains, *i.e.*, 0.2 dB at BER of 10^{-6} for rate 2/3, and 0.05-0.1 dB at BER of 10^{-6} for other rates, with low error floor, when compared with the counterparts in [9] and [38].

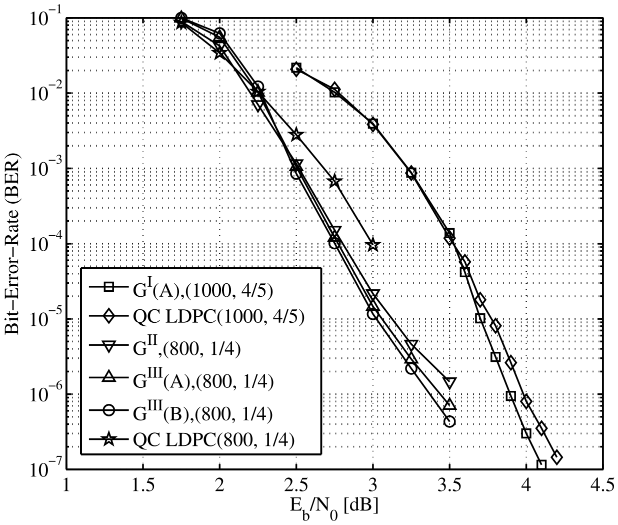


FIGURE 11. BER of the codes (G^{I} vs the QC LDPC code (1000, 4/5) [36] and $G^{II}\&G^{III}$ vs the QC LDPC code (800, 1/4) [37]).

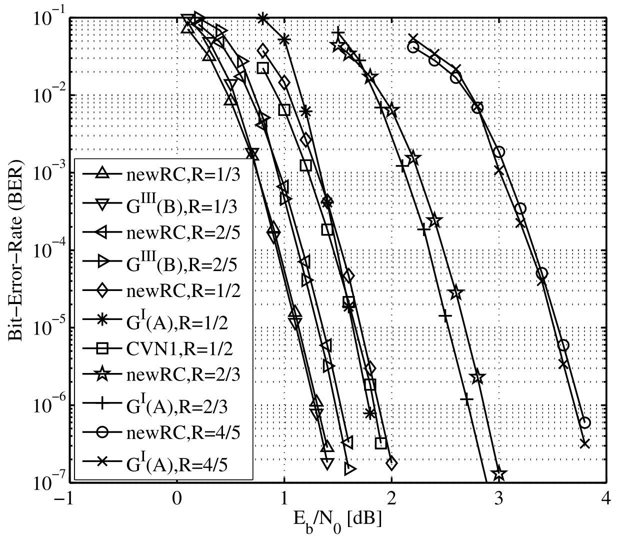


FIGURE 12. BER of the proposed codes, the newRC codes (K=1000, rate={1/3, 2/5, 1/2, 2/3, 4/5}) [38] and the CVN1 code (K=1000, rate=1/2) [9].

From Fig. 12, the proposed protograph G-LDPC codes $G^{I}(A)$ and $G^{III}(B)$ all outperform the well designed protograph codes in [9] and [38] slightly under the same code parameters. This mainly lies in the fact that it is also hard to remove all cycles in the short length codes, especially the compound loops. In addition, the design method of the counterparts has not consider the relationship of loops, *i.e.*, the ACE affection, which is also one of the key factors for code performance. Usually, short LDPC codes under specifically optimized degree profiles can not avoid all cycles. The cycles in the code's Tanner graph impair the performance, because the false message propagation easily leads to error propagation in the cycles. However, this phenomenon can be corrected just by introducing enough independent messages outside the cycles. In other words, enough independent links from outer variable nodes can remedy the cycle effect and reach an ideal performance of maximum likelihood. Therefore, the proposed method can be used to produce good short LDPC codes.

E. SHORT SUMMARY

in the above simulations, more performance is improved by puncturing the variable nodes with large degrees, because the punctured node equivalently brings more check equations and equivalent variable nodes with large degree, thus equivalently reduces the code rate and increases maximum variable node degree for better thresholds. So two phenomena never analyzed in literatures before are explained. One is that the punctured protograph LDPC codes without highest seeming variable degrees outperform their unpunctured counterparts. Another is that in all punctured protograph LDPC nodes with proper design, more performance gains are brought, when the nodes with highest degrees are punctured. They have been explained in Section III and are concluded as follows. Through careful design, exempted from cycle problems, the puncture of the node with large degree brings huge check relationship for the remainder variable nodes, which are connected with the same check nodes. More new equivalent virtual check nodes (or check equations) are increased for enhancing the reliability of the message propagation for the variable nodes connected to the virtual check nodes.

These three techniques for code design improve the performance efficiently. The EXIT chart is used to analyze the theoretic threshold of the codes. But the threshold is only valid when the input/output metrics of the nodes meet the Gaussian distributions and the code length is infinite for such result. So it is just an unconstrained performance bound, which is used to evaluate the code construction whether it is optimized well or not. Since it is quite easy to calculate the threshold of a seed protpgraph, it simplifies the analysis of the threshold, rather than the more complex prediction calculation in DE or GA [1]–[3]. The asymptotic weight analysis is applied too for the distribution calculation, thus more proper code structure is generated for low error floor with even better minimum distance. Since completely eliminating all loops in the Tanner graph of a QC LDPC code are quite difficult, especially for a code with short code length, the code performance still has some disparity, when compared with that of the theoretic threshold. But by careful design, most length-6 short loops can be eliminated to achieve maximum likelihood performance of the BP algorithm. Actually, the proposed codes achieve good performance mainly by reducing small stop sets or trapping sets due to the MACE algorithm. Namely, given unavoidable cycles in the Tanner graphs of the codes, especially the short cycles, more independent outer variable nodes are arranged to be connected with them. So independent and reliable outer metrics are introduced into the cycles and thus the false self-feedback messages in the cycles are corrected. Also the decoding of these codes can be flexibly implemented in a FPGA based LDPC decoder architecture, which supports run-time flexibility over any set of QC LDPC codes [39]. Additionally, an off-line design flow have supported for a chosen code selections and it obtains a high level of design-time and run-time flexibility [39].

In code design, the proposed codes use optimized seed protographs to generate the code frameworks. The seed protographs are simple in calculation of the code parameters as thresholds and so on. Also, the complexity of the optimized methods in code design, such as the modified EXIT and the MACE, are almost the same to those of the P-EXIT and the ACE. In the modified EXIT, we just add the iterative message propagations of degree-1 nodes other than that of the P-EXIT. The increase of the calculations is trivial, since only limited additions are introduced. In the MACE, we just extend single cycles to compound cycles, which are similarly calculated in global search of ACE values. So the complexities of the code design are similar to those of currently low-complex punctured ARA protograph codes [5], which use the above corresponding methods. Therefore, the designs of the codes also have the same low complexity as ARA codes. In code application, the proposed protograph G-LDPC code mainly uses a punctured pattern and QC structure, where the variable nodes with highest degree are punctured in Fig. 6 and Fig. 7. When compared with traditional un-punctured codes, such as the short-length QC LDPC codes in [36] and [37], the proposed codes have low maximum column weight in their check matrices as ARA codes. Therefore, the complexities of the proposed codes are similar to those of the ARA codes, which have been considered as low complexity.

Therefore, these three techniques in the optimized protograph G-LDPC code design, *i.e.*, the modified EXIT chart, the asymptotic weight analyses and the MACE, are united as a systematic code construction scheme. The produced codes are effective for practical wireless communications.

VI. CONCLUSION

In this paper, we present a new construction of protograph G-LDPC codes with good decoding thresholds and low error floors. The Tanner graph of the code is generated by an extension of a protograph doped by a simple hamming code. The base matrix is analyzed and optimized with the modified EXIT chart for good threshold and the asymptotic weight distributions for minimum distance and also the low error floor. Meanwhile, an MACE algorithm is also suggested to search the actual ACE values for optimal circulant offsets of the QC sub-matrices. In addition, G-LDPC codes with different code lengths and rates are flexibly designed with different protograph templates and the extended variable or the doped nodes. Simulation results show that the resulting codes achieve a little better performance, when compared with current codes, especially the short length codes. The proposed codes also possess good features of encodable and variable rate puncturable QC structure and thus can be efficiently applied in the coding practice with volatile wireless channels. Therefore, the proposed Protograph G-LDPC codes can be efficiently applied in new-generation power constraint wireless communications with good performance and low complexity.

REFERENCES

[1] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[2] T. J. Richardson and R. L. 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.

[3] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[4] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. New York, NY, USA: Cambridge Univ. Press, 2009, pp. 261–291.

[5] A. Abbasfar, K. Yao, and D. Divsalar, "Accumulate repeat accumulate codes," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Jun./Jul. 2004, pp. 509–513.

[6] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL INP, Tech. Rep. 42-154, JPL, Pasadena, CA, USA, Aug. 2003, pp. 42–154. [Online]. Available: https://tmo.jpl.nasa.gov/progress_report/42-154/154C.pdf

[7] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[8] N. Bonello, S. Chen, and L. Hanzo, "Construction of regular quasi-cyclic protograph LDPC codes based on vandermonde matrices," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2583–2588, Jul. 2008.

[9] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, "A survey on protograph LDPC codes and their applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1989–2016, 4th Quart., 2015.

[10] T. Van Nguyen, A. Nosratinia, J. P. Fonseka, K. Kiasaleh, and M. Torlak, "Design of capacity-approaching protograph-based LDPC coding systems," Ph.D. dissertation, Dept. Elect. Eng., Univ. Texas Dallas, Dallas, TX, USA, Dec. 2012.

[11] H. Y. Park, J. W. Kang, K. S. Kim, and K. C. Whang, "Efficient puncturing method for rate-compatible low-density parity-check codes," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 3914–3919, Nov. 2007.

[12] H. Pishro-Nik and F. Fekri, "Results on punctured low-density parity-check codes and improved iterative decoding techniques," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 599–614, Feb. 2007.

[13] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[14] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[15] G. Liva, W. E. Ryan, and M. Chiani, "Quasi-cyclic generalized LDPC codes with low error floors," *IEEE Trans. Commun.*, vol. 56, no. 1, pp. 49–57, Jan. 2008.

[16] J. Chen, R. M. Tanner, J. Zhang, and M. P. C. Fossorier, "Construction of irregular LDPC codes by quasi-cyclic extension," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1479–1483, Apr. 2007.

[17] T. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annu. Allerton Conf. Commun., Control Comput.*, Sep. 2003, pp. 1426–1435.

[18] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, Aug. 2004.

[19] J. Boutros, O. Pothier, and G. Zemor, "Generalized low density (Tanner) codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 1999, pp. 441–445.

[20] 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.

[21] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2007, pp. 3250–3254.

[22] D. Divsalar, "Ensemble weight enumerators for protograph LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seattle, WA, USA, Jul. 2006, pp. 1554–1558.

[23] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 3, Nov./Dec. 2005, pp. 5–10.

[24] D. Divsalar, S. Dolinar, and C. Jones, "Low-rate LDPC codes with simple protograph structure," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Sep. 2005, pp. 1622–1626.

[25] X. Hu, M. P. C. Fossorier, and E. Eleftheriou, "On the computation of the minimum distance of low-density parity-check codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2004, pp. 767–771.

[26] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[27] J. Bao, Y. Zhan, L. Yin, and J. Lu, "Design of efficient joint eIRA-coded MSK modulation systems for space communications," *IEEE Trans. Aerosp. Electon. Syst.*, vol. 48, no. 2, pp. 1636–1642, Apr. 2012.

[28] F. Steiner, G. Böcherer, and G. Liva, "Protograph-based LDPC code design for shaped bit-metric decoding," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 397–407, Feb. 2016.

[29] Y. Fang, S. C. Liew, and T. Wang, "Design of distributed protograph LDPC codes for multi-relay coded-cooperative networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7235–7251, Nov. 2017.

[30] Y. Fang, Y. L. Guan, G. Bi, L. Wang, and F. C. M. Lau, "Rate-compatible root-protograph LDPC codes for quasi-static fading relay channels," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2741–2747, Apr. 2016.

[31] G. Colavolpe, "On LDPC codes over channels with memory," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1757–1766, Jul. 2006.

[32] T. H. Cormen *et al.*, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2001, pp. 603–611.

[33] G. Liva and W. E. Ryan, "Short low-error-floor tanner codes with Hamming nodes," in *Proc. IEEE Military Commun. (MILCOM)*, Oct. 2005, pp. 208–213.

[34] CCSDS, "Research and development for space data system standards:Low density parity check codes for use in near-earth and deep space applications-experimental specification," CCSDS Orange Book, Washington, DC, USA, INFORMATIONAL REPORT 131x1o2e2, Sep. 2007 [Online]. Available: http://ccsds.cosmos.ru/publications/archive/131x1o2e2.pdf

[35] Working Group: SLS-SLP, "TM synchronization and channel coding–summary of concept and rationale," CCSDS Green Book, Washington, DC, USA, INFORMATIONAL REPORT 130.1-G-1, Nov. 2012, no. 2. [Online]. Available: https://public.ccsds.org/Pubs/130x1g2.pdf

[36] I. E. Bocharova, F. Hug, R. Johannesson, and B. D. Kudryashov, "High-rate QC LDPC codes of short and moderate length with good girth profile," in *Proc. 7th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, 2012, pp. 150–154.

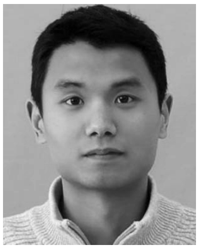
[37] H. Park, S. Hong, J.-S. No, and D.-J. Shin, "Design of multiple-edge protographs for QC LDPC codes avoiding short inevitable cycles," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4598–4614, Jul. 2013.

[38] T. V. Nguyen and A. Nosratinia, "Rate-compatible short-length protograph LDPC codes," *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 948–951, May 2013.

[39] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A flexible FPGA-based quasi-cyclic LDPC decoder," *IEEE Access*, vol. 5, pp. 20965–20984, 2017.

JIANRONG BAO (S'06–M'11) received the B.S. degree in polymeric materials and engineering and the M.S.E.E. degree from the Zhejiang University of Technology, Hangzhou, China, in 2000 and 2004, respectively, and the Ph.D.E.E. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2009.

He was a Post-Doctoral Researcher with Zhejiang University from 2011 to 2013 and also with Southeast University, Nanjing, China, from 2014 to 2017. In 2015, he joined Columbia University, New York, NY, USA, as a Visiting Scholar. He is currently a Professor with the Information Engineering School, Hangzhou Dianzi University, Hangzhou. He is also a Visiting Scholar with the National Mobile Communications Research Laboratory, Southeast University. His main research interests include modern wireless communications, cognitive radio, information theory and coding, communication signal processing, synchronization and equalization, and wireless sensor network.



BIN JIANG received the B.S.E.E. and M.S.E.E. degrees from the School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China, in 1999 and 2004, respectively.

He is currently with the School of Communication Engineering, Hangzhou Dianzi University. His main research interests include wireless communications, signal processing, information theory, and channel coding.

CHAO LIU received the B.S.E.E. and Ph.D.E.E. degrees from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2000 and 2005, respectively.

He is currently an Associate Professor with the School of Information Engineering, Hangzhou Dianzi University, Hangzhou, China. His research interests include modern wireless communication and coding, and multi-in multi-out multi-user detection.

• • •