IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 16, 2021, accepted April 4, 2021, date of publication April 8, 2021, date of current version April 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071846

On the Construction of Multitype Quasi-Cyclic Low-Density Parity-Check Codes With Different Girth and Length

FARZANEH ABEDI AND MOHAMMAD GHOLAMI
Department of Mathematics, University of Shahrekord, Shahrekord 88186-34141, Iran
Corresponding author: Mohammad Gholami (gholami-m@sci.sku.ac.ir)

ABSTRACT *Multitype quasi-cyclic (QC) low-density parity-check (LDPC) codes* are a class of protograph LDPC codes lifted cyclically from protographs with multiple edges, represented by two *weight* and *slope* matrices. For a given weight-matrix, an approach is proposed to find the *maximum-achievable girth* $g_{\max}$ of the corresponding multitype QC-LDPC codes by some *inevitable chains* having less complexity than the existing methods. This advantage leads to some new patterns of the weight matrices such that the corresponding codes have some improvements in terms of the maximum-achievable girths or the minimum-distance upper-bounds. In continue, for a given weight-matrix with maximum-achievable girth $g_{\max}$, some slope-matrices are constructed by a *depth-first search* algorithm for which the corresponding multitype QC-LDPC codes with even *girth* g, $g \leq g_{\max}$, have smaller lengths, higher rates, or larger minimum-distances than the state-of-the-art achievements. Simulation results show that the constructed codes have some *error-rate* advantages than PEG, random-like, CCSDS, and 802.11n/ac IEEE standard LDPC codes.

INDEX TERMS Circulant permutation matrix, girth, QC-LDPC codes, Tanner graph.

I. INTRODUCTION

Low-density parity-check (LDPC) codes [1], as a main class of error correcting linear codes, can be specified by their sparse parity-check matrices (PCM's) H and their associated *Tanner graphs* TG(H) [2]. Although the minimum-distance of LDPC codes is less than that of the best known linear codes, due to their structures, they are suitable [3] for low-complexity iterative decoding methods, such as *Pearl's belief propagation (BP) algorithm*, adopted in many practical applications.

The performance of LDPC codes of small length may be strongly affected by their cycle properties such as *girth* and *stopping set* [4]. In fact, the girth, i.e., the shortest cycles in the Tanner graph, is one important factor to design LDPC codes [7]–[12], [18] with good error-correcting properties. The progressive-edge-growth algorithm (PEG) construction builds up a Tanner graph, or equivalently a parity-check matrix, for an LDPC code by maximizing the local girth at symbol nodes in a greedy algorithm [33]. In PEG constructions, an algorithm is applied to find a PEG LDPC code with a given target girth, however the real girth is usually less than the target. Moreover, it was shown in [13] that the minimum-distance of an LDPC code can limit the error performance at a high signal-to-noise ratio (SNR) and is also important in understanding the likelihood of undetected errors which are a critical concern in many applications.

Quasi-cyclic (QC) LDPC codes [14], [15] are the most promising class of structured LDPC codes due to their ease of hardware implementations using simple shift registers and excellent performances over noisy channels when they are decoded by message-passing algorithms. The PCM's of QC-LDPC codes are comprised of blocks of circulant matrices, classified by the researchers as type-w QC-LDPC codes if each nonzero block is a combination of at-most w circulant permutation matrices (CPMs). For example, for $1 \leq w \leq 4$, the readers are suggested to refer [7]–[12], [14]–[21] for type-I QC-LDPC codes, [22]–[27], [31] for type-II QC-LDPC codes, [6], [16], [27]–[29] for type-III QC-LDPC codes, and [28], [29] for type-IV QC-LDPC codes. In [6], the benefit of *perfect difference families* (PDF's) and *quasi-perfect difference families* (QPDF's) is considered to define a class of combinatoric-based 4-cycle free type-III and type-IV QC-LDPC codes with shortest possible length.

The associate editor coordinating the review of this manuscript and approving it for publication was Zesong Fei.

Lally presented an explicit construction for type-II QC-LDPC codes with girth six whose PCMs include weight-2 circulant entries [23]. Moreover, some combinatorial methods based on *Sidon sequences* [24] and *perfect cyclic difference sets* [25], [26] are used for the construction of a class of type-II QC-LDPC codes with girth at least 6. Some type-II, III QC-LDPC codes with girth at most ten are presented in [27] based on a search algorithm. Recently, some type-w QC-LDPC codes, $2 \leq w \leq 6$, with girths at least 6, were constructed by some explicit methods in [28].

Type-w, $w \geq 1$, QC-LDPC codes are generalized to define the class of multiple-edge protograph LDPC codes, called briefly *multitype QC-LDPC codes* in this paper which can potentially achieve larger minimum-distances compared to type-I QC-LDPC codes [16], although, there is a hindrance to construct multitype QC-LDPC codes with more flexibility in some parameters, such as the regularity or the girth.

In linear algebra, a *circulant matrix* is a square matrix in which each column vector is rotated one position to the down relative to the preceding column vector. Especially, a *circulant permutation matrix* (CPM) is a circulant matrix in which each column contains precisely a single 1. Clearly, each circulant matrix can be described a sum of some CPMs. A multitype QC-LDPC code with CPM-size m, can be described by its PCM $H = (H_{i,j})$ in which each $H_{i,j}$ is a circulant matrix of size m or the $m \times m$ zero matrix. Corresponding to PCM $H = (H_{i,j})$ of a multitype QC-LDPC code, a *weight-matrix* is associated in which (i, j)th entry indicates the *circulant weight*, i.e., the number of 1 in each column of $H_{i,j}$.

Alternatively, the PCM H of a multitype QC-LDPC code can be shown in the polynomial form $H(x)$ whose (i, j)th entry is the polynomial representation of $H_{i,j}$. In [16], the authors have presented two upper-bounds on the minimum-distance of multitype QC-LDPC codes based on their weight-matrices and the polynomial forms of their PCMs. Moreover, they have shown that against weight matrices with entries 0, 1, the weight matrices with the elements greater than 1 correspond to some multitype QC-LDPC codes haveing larger possible minimum-distances. Especially, for $L = 4, 5$, some patterns for $3 \times L$ weight matrices achieving the minimum-distance upper-bounds have been presented in [16].

For a given weight-matrix W, by the *maximum-achievable girth* $g_{\max} = g_{\max}(W)$, we mean the maximum girth achieved by the set of multitype QC-LDPC codes which have the same weight-matrix W. Using the concept of protograph [30], the authors in [31] have investigated all of the subgraph patterns (inevitable cycles) of multiple-edge protographs which prevent protograph QC-LDPC codes to have girths exceeding than the *maximum-achievable girth* $g_{\max}$, $g_{max} \leq 20$. Then, some constructions of regular multiple-edge protograph QC-LDPC codes with maximum-achievable girth 14 were presented in [31].

In this paper, we redefine each *multitype QC-LDPC code* by two *weight* and *slope* matrices which are useful to pursue each cycle in the Tanner graph by an *admissible chain* and a *linear modular equation* between the slopes. Based on the approach in [31], for a given weight-matrix W, finding the maximum-achievable girth $g_{\max} = g_{\max}(W)$ needs a complexity that increases exponentially when the size of the matrix enlarges linearly, because the patterns corresponding to the girth g, $g < g_{\max}$, must be checked for all submatrices of W. Inspired by the approach used in [10], we propose a new method to find $g_{\max}$ based on *inevitable chains* which are some admissible chains that exist regardless of slope values and CPM-sizes. This approach is useful to generate some new weight matrices by a random search in which the corresponding multitype QC-LDPC codes have larger maximum-achievable girths or minimum-distance upper-bounds when they are compared with the weight matrices in [31].

In [20], some slope values corresponding to type-I QC-LDPC codes with girth at most 14 have been generated by a greedy search algorithm being generalized in [31] to construct some type-II QC-LDPC codes with girth at most 14. In this paper, we present a *depth-first search algorithm* which for a given weight-matrix W and even integer g, $g \leq g_{\max}(W)$, efficiently generates a proper slope-matrix such that the corresponding multitype QC-LDPC code has girth g with the CPM-size as small as possible. As the outputs of the algorithm, some multitype QC-LDPC codes will be constructed having some improvements in the parameters, such as rates, lengths, minimum-distances, and cycle distributions than the codes in [7], [8], [16], [18], [20], [23]–[25], [29], [31]. Finally, simulation results show better bit-error-rate performance of the constructed codes than PEG [33], random-like [3], and codes in [16], [19], [25], [31], [34]–[36].

The outline of the paper is as follows: In Section II, we discuss the structure of multitype QC-LDPC codes and two upper-bounds on the minimum-distance are given. Then the existence of cycles in the Tanner graph of a QC-LDPC code will be investigated by a necessary and sufficient condition. In Section III, this condition is used to define inevitable chains which are useful to find maximum-achievable-girth of a specific weight-matrix by an efficient algorithm. Then, some weight-matrices with good upper-bounds on minimum-distance are found for different rates and girths. In Section IV, we apply a depth-first algorithm to search the slope-vectors corresponding to multitype QC-LDPC codes with a given weight-matrix and different girths (not greater than the maximum-achievable girth), such that the constructed codes have CPM-sizes as small as possible. In continue, algorithm outputs are given by introducing some examples and tables. Some error-rate comparisons are given between the constructed multitype QC-LDPC codes and some other known constructions in Section V, and finally, the conclusion is given in Section VI.

II. MULTITYPE QC-LDPC CODE

Let m, s be some positive integers with $0 \leq s < m$. By the CPM with shift value s, denoted by $\mathcal{I}_m^s$, or $\mathcal{I}^s$ when m is known, we mean the $m \times m$ permutation matrix $(a_{i,j})_{1 \leq i,j \leq m}$, in which $a_{i,j} = 1$ if and only if $i - j = s \bmod m$. Moreover, we accept this notation that I^∞ is the $m \times m$ zero matrix. Now, for positive integers J, L, $J < L$, let $W = (w_{i,j})$ be a $J \times L$ matrix of some nonnegative integers. By a (J, L) multitype QC-LDPC code with *weight-matrix* W and CPM-size m, where m is greater than all elements of W, we mean the QC-LDPC code of length mL with PCM $H = (H_{i,j})_{1\le i\le J,1\le j\le L}$, in which $H_{i,j}$ is the $m \times m$ zero matrix, when $w_{i,j} = 0$, or otherwise the sum of $w_{i,j}$ CPMs of size m, i.e., $H_{i,j} = \mathcal{I}^{s_{i,j}^{(1)}} + \mathcal{I}^{s_{i,j}^{(2)}} + \cdots + \mathcal{I}^{s_{i,j}^{(w_{i,j})}}$, for some shift values $0 \le s_{i,j}^{(1)} < s_{i,j}^{(2)} < \cdots < s_{i,j}^{(w_{i,j})} < m$. In this case, if $s_{i,j} = (s_{i,j}^{(1)}, s_{i,j}^{(2)}, \ldots, s_{i,j}^{(w_{i,j})})$ is the vector of shift values appeared in $H_{i,j}$, then the matrix $S = (s_{i,j})$ is called the *slope-matrix*. Clearly, if $w_{i,j} = 0$, then $H_{i,j}$ is the zero matrix of size m and $s_{i,j} = ()$ is the empty vector.

For each $w \ge 1$, type-w QC-LDPC codes are a subclass of multitype QC-LDPC codes in which each element of the weight-matrix $W = (w_{i,j})$ is at most w, i.e., $w_{i,j} \le w$. In the following, we give an example of a type-IV QC-LDPC code with girth 6.

Example 2.1: For $m = 19$, the following matrices W and S can be considered as the weight and slope matrices of a girth-6 $(2, 3)$ type-IV QC-LDPC code with PCM H, respectively.

$$W = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 4 & 2 \end{pmatrix}, \quad S = \begin{pmatrix} (0,1) & (0,2,5) & (0) \\ () & (0,1,7,11) & (3,8) \end{pmatrix},$$

$$H = \begin{pmatrix} \mathcal{I}+\mathcal{I}^1 & \mathcal{I}+\mathcal{I}^2+\mathcal{I}^5 & \mathcal{I} \\ 0 & \mathcal{I}+\mathcal{I}^1+\mathcal{I}^7+\mathcal{I}^{11} & \mathcal{I}^3+\mathcal{I}^8 \end{pmatrix}.$$

Hereinafter, to reduce the space limitation, sometimes, the weight-matrix $W = (w_{i,j})$ of a (J, L) multitype QC-LDPC code is denoted by two designs $\mathcal{B} = [B_1, B_2, \ldots, B_L]$ and $\mathcal{W} = [W_1, W_2, \ldots, W_L]$, called *block-design* and *weight-design*, respectively, where $B_j = [i : w_{i,j} \neq 0]$ and $W_j = [w_{i,j} : w_{i,j} \neq 0]$ are two lists containing of row indices and values of nonzero elements in jth column of W, respectively. For example, in Example 2.1, the weight-matrix W can be denoted by $\mathcal{B} = [[1], [1, 2], [1, 2]]$ and $\mathcal{W} = [[2], [3, 4], [1, 2]]$. In continue, we discuss some properties of multitype QC-LDPC codes, such as minimum-distances, girths, and CPM-sizes.

A. MINIMUM-DISTANCE

In [16], two upper-bounds on minimum-distance of multitype QC-LDPC codes have been presented which can be used when there is no way to find the actual minimum-distance of a QC-LDPC code in a real-time. For multitype QC-LDPC codes with a given weight-matrix W, the following upper-bound on minimum-distance is presented based on W.

$$d_{\min}(C) \leqslant \min_{\substack{S \subseteq \{1,\ldots,L\} \\ |S| = J+1}}^{*} \sum_{i\in S} perm(W_{S\setminus i}), \qquad (1)$$

where operator $\min^*$ gives back the minimum value of all nonzero entries in a list of values and index $S \setminus i$ for a matrix W means the submatrix of W that contains only the columns whose indices appear in the set $S \setminus i$. Moreover, a permanent of an $m \times m$ square matrix $B = (b_{i,j})$ is defined as $perm(B) = \sum_{\sigma} \prod_{j\in\{1,\ldots,m\}} b_{j,\sigma(j)}$, where the summation is over all $m!$ permutations σ on the set $\{1, \ldots, m\}$.

Corresponding to the PCM H of a multitype QC-LDPC code with CPM-size m, weight-matrix $W = (w_{i,j})$, and slope-matrix $S = (s_{i,j})$ ($s_{i,j} = (s_{i,j}^{(1)}, \ldots, s_{i,j}^{(w_{i,j})})$), the polynomial form of the PCM, $H(x) = (h_{i,j}(x))$, is associated, in which $h_{i,j}(x) = \sum_{t=1}^{w_{i,j}} x^{s_{i,j}^{(t)}} \in \frac{\mathbb{F}_2[x]}{< x^m - 1 >}$ for $w_{i,j} \neq 0$ and $h_{i,j}(x) = 0$ otherwise. Also the weight of each polynomial $h_{i,j}(x)$, denoted by $wt(h_{i,j}(x))$, is defined as the number of nonzero terms in $h_{i,j}(x)$. Based on the PCM $H = H(x)$, represented in *polynomial* form, another upper-bound on minimum-distance is reported in [16] as follows.

$$d_{\min}(C) \leqslant \min_{\substack{S \subseteq \{1,\ldots,L\} \\ |S| = J+1}}^{*} \sum_{i\in S} wt(perm(H_{S\setminus i}(x))). \qquad (2)$$

Hereinafter, we denote the upper-bounds on minimum-distance given by (1) and (2) by the notations $U_{d_{min}}(W)$ and $U_{d_{min}}(H)$, respectively. By (1), if the weight-matrix is fully-one, then the constructed (J, L) type-I QC-LDPC codes has a minimum-distance at most $(J + 1)!$. This can be generalized easily to a fully-w weight-matrix as presented by Proposition 2.2 (for $w = 2$, this is the same with the bound in [23]). Before, for positive integers $J, L, w, J < L$, by $W(J, L; w)$ we mean a $J \times L$ matrix that each entry equals to w.

Proposition 2.2: For type-w QC-LDPC codes with weight-matrix $W = W(J, L; w)$, we have $U_{d_{\min}}(W) = w^J(J+1)!$. Moreover, for the following $J \times L$ weight-matrices

$$W^{(1)} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 2 & 2 & \cdots & 2 \\ \vdots & \vdots & \ddots & \vdots \\ J & J & \cdots & J \end{pmatrix}, \quad W^{(2)} = \begin{pmatrix} 1 & 2 & \cdots & L \\ 1 & 2 & \cdots & L \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \cdots & L \end{pmatrix},$$

we have $U_{d_{\min}}(W^{(1)}) = (J + 1)!J!$ and $U_{d_{\min}}(W^{(2)}) = J! \sum_{i=1}^{J+1} \frac{(J+1)!}{i}$.

Proof: To view $U_{d_{\min}}(W) = w^J(J + 1)!$, the reader can refer to [23]. For the rest, we note that the permanent of each submatrix $W^{(1)}_{S\setminus\{i\}}$, $S \subseteq \{1, 2, \ldots, L\}$, $|S| = J + 1$, is the same as the permanent of J first columns of $W^{(1)}$ which is equal to $(J!)^2$, because the permanent of the $J \times J$ fully-one matrix is $J!$. Then, $\sum_{i\in S} \text{perm}(W_{S\setminus i}) = (J + 1)(J!)^2 = (J + 1)!J!$. For the weight-matrix $W^{(2)}$, the minimum value of the summation in (1) is achieved for $S = \{1, \ldots, J + 1\}$. In this case, $W^{(2)}_{S\setminus\{i\}}$, for each $i \in S$, is the submatrix of $W^{(2)}$ in which the columns are fully-j vectors of length J, $1 \le j \le J+1$, $j \neq i$. Therefore, $\text{perm}(W^{(2)}_{S\setminus\{i\}}) = J!\frac{(J+1)!}{i}$ and the proof is completed. ■

Propositions 2.2 shows that there are weight matrices, different from the all-one weight-matrices, such that the minimum-distance upper-bounds are larger than $(J+1)!$, in which J is the number of rows. In general, employing weight matrices with many large entries leads to some QC-LDPC codes that have potentially large minimum-distances. On the other hand, from the bound given by (2), larger minimum-distances within a given $J \times L$ weight-matrix can be achieved by constructing multitype QC-LDPC codes with appropriate slope-matrices.

B. GIRTH

Let H be the PCM of a (J, L) multitype QC-LDPC code with weight-matrix $W = (w_{i,j})$ and slope-matrix $S = (s_{i,j})$, $s_{i,j} = (s_{i,j}^{(1)}, \ldots, s_{i,j}^{(w_{i,j})})$. The following proposition gives a necessary and sufficient condition for the existence of cycles in TG(H).

Proposition 2.3: [6] Each cycle of length $2l$, $l > 2$, in TG(H) corresponds to a chain (i_0, j_0, t_0); (i_1, j_0, t_1); (i_1, j_1, t_2); (i_2, j_1, t_3); $\ldots$; $(i_{l-1}, j_{l-1}, t_{2l-2})$; (i_l, j_{l-1}, t_{2l-1}); $(i_l, j_l, t_{2l}) = (i_0, j_0, t_0)$ in which $w_{i_k,j_k} \neq 0$, $w_{i_k,j_{k+1}} \neq 0$, $1 \leq i_k \leq J$, $1 \leq j_k \leq L$, $1 \leq t_{2k} \leq w_{i_k,j_k}$, and $1 \leq t_{2k+1} \leq w_{i_{k+1},j_k}$, for each $0 \leq k \leq l-1$, such that the following conditions are satisfied:

I. If $i_k = i_{k+1}$, then $t_{2k} \neq t_{2k+1}$ and if $j_k = j_{k+1}$, then $t_{2k+1} \neq t_{2k+2}$.
II. If $\Delta_k := s_{i_k,j_k}^{(t_{2k})} - s_{i_{k+1},j_k}^{(t_{2k+1})}$, then we have

$$\sum_{k=0}^{l-1} \Delta_k = 0 \pmod{m}. \tag{3}$$

Hereinafter, by an *admissible chain* of length $2l$, we mean a chain (i_0, j_0, t_0); (i_1, j_0, t_1); $\ldots$; (i_l, j_{l-1}, t_{2l-1}); $(i_l, j_l, t_{2l}) = (i_0, j_0, t_0)$ satisfied in Condition I of Proposition 2.3. Moreover, an *inevitable chain* is an admissible chain for which Condition II always holds, for all slope-matrices, i.e., the equality in Eq. 3 is established (instead of a modular relation), independently from the slope values. For example, corresponding to each weight-matrix having submatrix $\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$, we have the following inevitable chain of length 12, shown by Fig. 1.

$$\begin{aligned}&(2,2,2);(1,2,1);(1,1,1);(1,1,2);(1,2,1);(2,2,2);\\&(2,2,1);(1,2,1);(1,1,2);(1,1,1);(1,2,1);(2,2,1).\end{aligned} \tag{4}$$

In fact, corresponding to the inevitable chain in (4), the following relation holds, independently from the slope values.

$$(s_{2,2}^{(2)} - s_{1,2}^{(1)}) + (s_{1,1}^{(1)} - s_{1,1}^{(2)}) + (s_{1,2}^{(1)} - s_{2,2}^{(2)}) + (s_{2,2}^{(1)} - s_{1,2}^{(1)}) + (s_{1,1}^{(2)} - s_{1,1}^{(1)}) + (s_{1,2}^{(1)} - s_{2,2}^{(1)}) = 0.$$

In mathematics, a multiset is a modification of the concept of a set that, unlike a set, allows for multiple instances for each of its elements. For the admissible chain $\mathcal{A}$ satisfied in Condition I of Proposition 2.3, let $\mathcal{M}_1(\mathcal{A}) = \{(i_k, j_k, t_{2k}) \,|\, 0 \leq k \leq l-1\}$ and $\mathcal{M}_2(\mathcal{A}) = \{(i_{k+1}, j_k, t_{2k+1}) \,|\, 0 \leq k \leq l-1\}$ be the multisets containing indices listed in the first and second parts of Δ_k, respectively. To determine

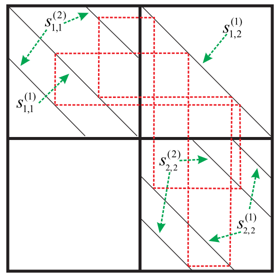

FIGURE 1. An inevitable cycle corresponding to the inevitable chain in weight-matrix W_1.

whether $\mathcal{A}$ is inevitable, it is sufficient to check $\mathcal{M}_1(\mathcal{A}) = \mathcal{M}_2(\mathcal{A})$. For example, for the admissible chain $\mathcal{A}$ shown by Eq.4, we have the following multisets:

$$\begin{aligned}&\mathcal{M}_1(\mathcal{A})\\&= \{(2,2,2),(1,1,1),(1,2,1),(2,2,1),(1,1,2),(1,2,1)\},\\&\mathcal{M}_2(\mathcal{A})\\&= \{(1,2,1),(1,1,2),(2,2,2),(1,2,1),(1,1,1),(2,2,1)\}\end{aligned}$$

which are equal to each other. Then, $\mathcal{A}$ is an inevitable chain.

C. CPM-SIZE

In [5], the following lower-bound on the CPM-size of multitype QC-LDPC codes with girth 6 is presented.

Lemma 2.4: For $J \times L$ weight-matrix $W = (w_{i,j})$, let

$$\mathcal{X} = \max\{2\sum_{j=1}^{L}\binom{w_{i,j}}{2}; \quad i \in \{1,\ldots,J\}\},$$

$$\mathcal{Y} = \max\{2\sum_{i=1}^{J}\binom{w_{i,j}}{2}; \quad j \in \{1,\ldots,L\}\},$$

$$\mathcal{Z} = \max\{\sum_{j=1}^{L} w_{i,j} \times w_{i',j}; \quad i \neq i'; \quad i, i' \in \{1,\ldots,J\}\},$$

then, the lower-bound on the CPM-size of type-w, $w \geq 1$, QC-LDPC codes with girth 6 is $LB = \max\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$.

III. MAXIMUM-ACHIEVABLE GIRTH OF MULTITYPE QC-LDPC CODES

By Proposition 2.3, a necessary and sufficient condition for TG(H) to have girth at least $2(l+1)$ is as follows.

$$\sum_{k=0}^{p} \Delta_k \neq 0 \pmod{m}, \tag{5}$$

for each $2 < p < l$ and each admissible chain (i_0, j_0, t_0); (i_1, j_0, t_1); $\ldots$; (i_l, j_{l-1}, t_{2l-1}); $(i_l, j_l, t_{2l}) = (i_0, j_0, t_0)$. Note that inevitable chains are independent from the slope values, but they depend on the weight-matrix. Here, we show that the maximum-achievable girth of a given weight-matrix, can be derived by the length of the minimum inevitable chain.

Lemma 3.1: For the weight-matrix W, the maximum-achievable girth $g_{\max}(W)$ is equal to the length of the minimum inevitable chain.

Proof: Clearly, corresponding to each QC-LDPC code with girth $g_{\max}(W)$, there is no inevitable chain of length $2l$, $l < g_{\max}(W)/2$. Then, $g_{\max}(W)$ is a lower-bound on the length of minimum inevitable chains. On the other hand, if $L_{\min}$ is the length of the minimum inevitable chain, then each admissible chain of length $2l$, $2l < L_{\min}$, is not inevitable, thus Eq. 3 does not hold, i.e., there are some slopes $s^{(t_{2k})}_{i_k,j_k}$ and $s^{(t_{2k+1})}_{i_{k+1},j_k}$, such that $\sum_{k=0}^{l-1}(s^{(t_{2k})}_{i_k,j_k} - s^{(t_{2k+1})}_{i_{k+1},j_k}) \neq 0$. However, clearly, the system of such nonzero linear equations has a common solution, for enough large slopes (the union of hyperplanes can not cover the whole space). Therefore, the corresponding QC-LDPC code C constructed from such slope values has no cycle of length $2l$, $2l < L_{\min}$, then $g_{\max}(W) \geq girth(C) \geq L_{\min}$. Therefore, $L_{\min} = g_{\max}(W)$. ■

In [31], the authors have proposed some matrices P_{2i}, $i \geq 3$ which prevent QC-LDPC codes from having large girth by inducing *inevitable cycles*. For example $P_6 = (3)$, $P_8 = (2 \quad 2)$, and

$$P_{10} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix},$$

$$P_{12} = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

$$P_{14} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

For each even g, to have a weight-matrix W with maximum-achievable girth at least g, we must design W such that it does not contain submatrices P_{2i}, $i < g/2$ or their transposes. For example, in [31], the authors have presented the following patterns of the weight-matrices to avoid inevitable cycles of length less than 12.

$$W_1 = \begin{pmatrix} 2&1&1&1&1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&2&1&1&1&0&0&0 \\ 0&2&0&0&0&1&0&0&0&1&1&1 \\ 0&0&2&0&0&0&1&1&0&1&1&0 \\ 0&0&0&2&0&0&1&0&1&1&0&1 \\ 0&0&0&0&2&0&0&1&1&0&1&1 \end{pmatrix}, \quad (6)$$

$$W_2 = \begin{pmatrix} 2&0&0&0&0&1&1&0 \\ 0&2&0&1&0&0&1&0 \\ 0&0&2&0&1&0&1&0 \\ 1&0&0&2&0&0&0&1 \\ 0&1&0&0&2&0&0&1 \\ 0&0&1&0&0&2&0&1 \end{pmatrix}.$$

However, this method has a high complexity, especially when g increases. Because, on one hand, finding all P_{2i}, $i < g$, needs a computer search whereas the numbers enlarge exponentially by increasing i, on the other hand, all of the row and column permutations of P_{2i}, $i < g$, and their transposes must be checked as the submatrices of the weight-matrix.

Based on Lemma 3.1, we now give an algorithm that finds the maximum-achievable girth of a weight-matrix by inevitable chains with minimum lengths.

Algorithm 1 Finding Maximum-Achievable Girth of a Weight-Matrix

$l \leftarrow 2$.
loop
  for each admissible chain $\mathcal{A}$ associated to the weight-matrix W **do**
    if $\mathcal{M}_1(\mathcal{A}) = \mathcal{M}_2(\mathcal{A})$ **then**
      return $2l$
    end if
  end for
  $l \leftarrow l + 1$.
end loop

In fact, in Algorithm 1, a depth-first search is used to find all admissible chains satisfied in Condition I of Proposition 2.3. Then, to determine whether the constructed admissible chain $\mathcal{A}$ is inevitable, it is sufficient to check the equality between multisets $\mathcal{M}_1(\mathcal{A})$ and $\mathcal{M}_2(\mathcal{A})$. Finally, Lemma 3.1 is used to find $g_{\max}(W)$ by the length of the shortest inevitable chain.

A. COMPLEXITY OF THE ALGORITHM

For positive integers J, L, $J < L$, let $W = (w_{i,j})_{J\times L}$ be the weight-matrix with $w = \max\{w_{i,j} : 1 \leq i \leq J, 1 \leq j \leq L\}$. Clearly, all admissible chains of length $2l$ can be generated in at most $J^l L^l w^{2l}$ ways. Moreover, for each admissible chain $\mathcal{A}$ of length $2l$, the equality of multisets $\mathcal{M}_1(\mathcal{A})$ and $\mathcal{M}_1(\mathcal{A})$ will be checked in at most $o(l \log l)$. Then, the overall complexity is $\sum_{l=2}^{l_{\max}} J^l L^l w^{2l} o(l \log l)$, where $l_{\max}$ is the maximum number of iterations used to find $g_{\max}(W)$. Although the complexity is still high when $l_{\max}$ enlarges, applying software Maple on a 2.6 GHz CPU and 4 GB RAM, Table 1 shows a prominent gap between the running-time of Algorithm 1 in terms of seconds, denoted by T_1, rather than the method in [31], denoted by T_2. By the outputs of Table 1, it can be seen that the method in [31] to find the maximum-achievable girth is still superior for $g_{\max}(W) \leq 12$.

In continue, applying Algorithm 1, we give some examples of weight matrices whose corresponding QC-LDPC codes have more flexible column-weights, rates, or minimum-distance upper-bounds (given by Eq.1) than type-I and type-II QC-LDPC codes in [31].

Example 3.2: Let W_1, W_2, and W_3 be the following 6×8, 6×12, and 9×12 weight matrices, respectively.

$$W_1 = \begin{pmatrix} 0&0&0&1&1&1&1&0 \\ 0&1&1&0&0&2&0&1 \\ 2&0&0&1&0&1&1&0 \\ 1&0&1&0&2&0&0&1 \\ 0&2&0&1&1&0&1&0 \\ 1&1&1&0&0&0&0&1 \end{pmatrix},$$

TABLE 1. A comaprison between the running times of Algorithm 1 and the method in [31] to find $g_{max}(W)$.

(J, L)	$g_{\max}$	T_1 (s)	T_2 [31] (s)	The weight-matrix W
$(5, 10)$	10	10.422	.625	$\mathcal{B}$=[[3,4,5],[3,4,5],[1,2,4],[3,4,5],[1,3,5],[2,4,5],[2,4,5],[3,4,5],[3,4,5],[2,3,5]], $\mathcal{W}$=[[1,1,2],[1,2,1],[1,2,1],[1,1,1],[1,2,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]
$(6, 12)$	12	18.344	3.594	$\mathcal{B}$=[[1,2],[1,3],[1,4],[1,5],[1,6],[2,3],[2,4,5],[2,4,6],[2,5,6],[3,4,5],[3,4,6],[3,5,6], $\mathcal{W}$=[[2,1],[1,2],[1,2],[1,2],[1,2],[2,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]
$(9, 12)$	14	11.359	38.828	$\mathcal{B}$=[[1,8],[2,8],[3,8],[4,8],[5,9],[6,9],[7,9],[3,4,9],[2,4,5],[1,5,6],[2,6,7],[1,3,7]], $\mathcal{W}$ =[[2,1],[2,1],[2,1],[2,1],[2,1],[2,1],[2,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]
$(9, 15)$	14	40.625	65.141	$\mathcal{B}$=[[1,8],[2,8],[3,8],[4,8],[5,9],[6,9],[7,9],[1,2,9],[3,4,9],[5,6,8],[1,3,5],[4,5,7],[2,4,6],[1,6,7],[2,3,7]], $\mathcal{W}$=[[2,1],[2,1],[2,1],[2,1],[2,1],[2,1],[2,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]
$(28, 40)$	16	554.218	$> 10^6$	$\mathcal{B}$=[[1,3,20],[2,4,19],[1,21],[2,22],[1,12,24],[2,11,23],[1,16,25],[2,15,26],[3,6,28],[4,5,27],[3,10],[4,9],[3,13,24],[4,14,23], [18,22],[17,21],[8,19],[7,20],[5,21],[6,22],[5,16,24],[6,15,23],[7,10,27],[8,9,28],[13,26],[14,25],[7,18,23],[8,17,24],[11,20], [12,19],[9,16],[10,15],[11,13,28],[12,14,27],[11,18,25],[12,17,26],[13,16,19],[14,15,20],[15,18,28],[16,17,27]], $\mathcal{W}$=[[1,1,1],[1,1,1],[2,1],[2,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[2,1],[2,1],[1,1,1],[1,1,1],[2,1],[2,1],[2,1],[2,1], [2,1],[2,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[2,1],[2,1],[1,1,1],[1,1,1],[2,1],[2,1],[1,2],[1,2],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1], [1,1,1],[1,1,1],[1,1,1]]
$(12, 15)$	16	84.546	1130.172	$\mathcal{B}$=[[3,5,6],[3,8,9],[2,9,11],[6,7,12],[3,11,12],[3,10,11],[7,8,9],[5,8,11],[1,5,6],[1,3,4],[5,9,10],[5,10,12],[1,8,11],[1,4,10], [2,4,6]], $\mathcal{W}$ =[[1,1,1],[1,1,1],[2,1,1],[1,2,1],[1,1,1],[1,1,1],[1,1,1], [1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]
$(7, 8)$	18	1.908	2507.891	$\mathcal{B}$=[[1,2],[3,4],[1,4],[5,6],[1,6],[1,7],[2,3],[2,5]], $\mathcal{W}$ =[[2,1],[2,1],[1,1],[2,1],[1,1],[1,1],[1,1],[1,1]]
$(5, 8)$	20	.563	129.508	$\mathcal{B}$= [[4,5],[3,5],[3,4],[2,4],[1,5],[1,5],[2,5],[2,3]], $\mathcal{W}$ =[[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1]]
$(7, 9)$	20	1.219	918.281	$\mathcal{B}$=[[1,3],[1,4],[1,5],[2,4],[6,7],[1,7],[2,4],[2,5],[2,6]], $\mathcal{W}$ =[[2,1],[1,1],[1,1],[1,1],[2,1],[1,1],[1,1], [1,1],[1,1]]
$(7, 10)$	22	1.656	11943.078	$\mathcal{B}$ =[[3,6],[2,4],[3,7],[2,7],[5,6],[2,4],[1,5],[1,5],[4,6],[1,2]], $\mathcal{W}$=[[2,1],[1,1],[1,1],[1,1],[1,1],[1,1], [1,1],[1,1],[1,1],[1,1]]
$(8, 10)$	28	2.256	$> 10^6$	$\mathcal{B}$=[[5,7],[1,2],[6,8],[7,8],[1,3],[3,6],[5,8],[4,6],[1,7],[1,2]], $\mathcal{W}$ =[[1,1],[1,1],[1,1],[1,1],[1,1],[1,1], [1,1],[2,1],[1,1],[1,1]]

$$W_2 = \begin{pmatrix} 1&2&0&1&0&0&0&0&0&0&0&0 \\ 2&0&0&0&1&1&1&1&1&1&1&1 \\ 1&0&2&1&0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&1&1&0&1&0&1 \\ 0&0&0&2&0&1&1&0&1&1&1&1 \\ 0&1&1&0&1&1&0&1&1&0&1&0 \end{pmatrix},$$

$$W_3 = \begin{pmatrix} 0&2&0&0&0&0&0&0&0&1&0&1 \\ 0&0&0&0&1&1&0&0&1&1&1&0 \\ 0&0&1&0&0&0&1&1&0&1&1&0 \\ 0&0&2&1&1&0&0&0&0&0&0&1 \\ 1&0&0&0&1&0&0&0&0&0&1&0 \\ 0&1&1&0&0&1&0&0&1&0&0&0 \\ 2&0&0&0&0&1&0&1&0&0&0&1 \\ 1&0&0&1&0&0&1&0&1&0&0&0 \\ 0&1&0&1&0&0&1&1&0&0&0&0 \end{pmatrix}$$

Applying Algorithm 1, it is easy to check that W_1 and W_2 have maximum-achievable girth 12 and W_3 has maximum-achievable girth 14, where $U_{d_{min}}(W_1) = 248$, $U_{d_{min}}(W_2) = 72$, and $U_{d_{min}}(W_3) = 446$. On the other hand, for the weight matrices with the same size and the same maximum-achievable girth in [31], the authors have obtained bounds 174, 68, and 416 which are less than the bounds reported for W_1, W_2, and W_3, respectively. It is noticed that the column-weight of the constructed codes varied from 3 to 4, which is not smaller than the column-weight 3 of the codes in [31]. Besides, increasing the column-weight often leads to decreasing the girth which did not happen for the constructed weight-matrices.

In Table 2, applying Algorithm 1, some weight-matrices W are constructed randomly such that the corresponding $U_{d_{min}}(W)$ is better than the bound (given by Eq. 1) on the weight-matrices in [31] with the same size $J \times L$ and maximum-achievable girth $g_{\max}$. It is noticed that the column-weight d_v of the constructed codes is at least 3, which is not smaller than the column-weight $d'_v = 3$ of codes in [31]. Moreover, Table 3 has provided some randomly-constructed $J \times L$ weight matrices such that the design-rate $\mathcal{R} = 1 - \frac{J}{L}$ of the generated type-w QC-LDPC codes is better than the design-rate $\mathcal{R}'$ of the codes in [31], whereas the maximum-achievable girth $g_{\max}$ is the same and the column-weight d_v is not smaller than d'_v [31].

IV. A DETERMINISTIC ALGORITHM TO GENERATE MULTITYPE QC-LDPC CODES

In this section, a depth-first search algorithm is presented in which for a given weight-matrix W with the corresponding maximum-achievable girth $g_{\max}(W)$ and even integer g, $2 \leq g \leq g_{\max}$, the outputs are a slope-matrix S and a CPM-size m, such that the corresponding multitype QC-LDPC code has girth at least g. In each step, the algorithm finds the minimum proper slope value such that there is no cycle smaller than g between this slope and the slopes recursively constructed in the previous steps. For this, Eq. 3 must not hold for the case that the CPM-size is infinity, i.e., all of the modular relations between the slopes are replaced with equality equations. After completing the whole elements of the slope-matrix S, the algorithm finds the minimum CPM-size m such that the corresponding QC-LDPC code has girth at least g. In fact, to find m, we collect the left-hand side of all such equations with their common divisors in P which can be considered as the set of all nonproper CPM-sizes. Now, the proper CPM-size m is chosen as the smallest nonnegative integer not belong to the set P.

To address the details, for positive integers $J, L, J < L$, let W be a $J \times L$ weight-matrix with maximum-achievable girth $g_{\max}$ and $2 \leq g \leq g_{\max}$ is even. Before, we noticed that the construction of the slope-matrix $S = (s_{i,j})$, $s_{i,j} = (s_{i,j}^{(1)}, \ldots, s_{i,j}^{(w_{i,j})})$, is based on a recursive method by constructing the nonempty elements of the matrix S (with the corresponding nonzero elements in W) from left to right in each entry, then traversing column-by-column in the matrix, i.e., for $w_{1,1} \neq 0$, finding the proper slopes in $s_{1,1}$ starting from $s_{1,1}^{(1)}$ till $s_{1,1}^{(w_{1,1})}$,

TABLE 2. $U = U_{d_{min}}(W)$ and d_v of some constructed weight-matrices of type-w QC-LDPC codes against $U' = U'_{d_{min}}(W)$ and d'_v of the patterns in [31].

(J, L)	w	$g_{\max}$	U	U'	d_v	d'_v	block design $\mathcal{B}$ and weight design $\mathcal{W}$
(6,8)	II	12	192	174	3,4	3	$\mathcal{B}$=[[1,2,3],[1,4,5],[1,3,6],[3,4,5],[2,5,6],[2,4,6],[2,4,6],[2,4,6]], $\mathcal{W}$=[[1,2,1],[2,1,1],[1,1,2],[2,1,1],[1,2,1],[1,1,1],[1,1,1],[1,1,1]]
(6,8)	I	12	248	110	4	3	$\mathcal{B}$=[[2,4,5,6],[2,3,5,6],[2,4,5,6],[3,4,5,6],[1,3,4,5],[2,3,4,6],[3,4,5,6],[1,3,5,6]], $\mathcal{W}$=[[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1]]
(6,8)	I	12	1200	110	5	3	$\mathcal{B}$=[[2,3,4,5,6],[2,3,4,5,6],[2,3,4,5,6],[1,3,4,5,6],[2,3,4,5,6],[2,3,4,5,6],[1,2,3,4,6],[1,2,4,5,6]], $\mathcal{W}$=[[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1]]
(6,12)	I	12	120	56	4	3	$\mathcal{B}$=[[3,4,5,6],[2,4,5,6],[2,4,5,6],[3,4,5,6],[3,4,5,6],[3,4,5,6],[3,4,5,6],[2,4,5,6],[1,2,3,4],[3,4,5,6],[3,4,5,6],[3,4,5,6]], $\mathcal{W}$=[[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1]]
(6,12)	I	12	192	56	4	3	$\mathcal{B}$=[[3,4,5,6],[2,4,5,6],[2,3,4,6],[3,4,5,6],[2,4,5,6],[3,4,5,6],[2,4,5,6],[1,3,4,6],[2,3,4,5],[1,4,5,6],[1,2,5,6],[3,4,5,6]], $\mathcal{W}$=[[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1]]
(9,12)	I	14	554	384	3,4	3	$\mathcal{B}$=[[4,7,8],[4,6,9],[1,3,6],[5,6,7,9],[1,7,8,9],[2,4,6,8],[1,3,9],[3,5,8,9],[1,2,5],[4,5,7],[2,3,6,7],[5,6,8]], $\mathcal{W}$=[[1,1,1],[1,1,1],[1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1],[1,1,1,1],[1,1,1],[1,1,1],[1,1,1,1],[1,1,1]]
(9,12)	I	14	4334	384	5	3	$\mathcal{B}$=[[5,6,7,8,9],[4,6,7,8,9],[1,2,5,8,9],[2,4,5,6,8],[2,4,7,8,9],[2,5,6,8,9],[2,4,6,7,8],[3,6,7,8,9],[4,5,7,8,9],[1,2,5,6,9],[3,6,7,8,9],[2,4,5,6,7]], $\mathcal{W}$=[[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1],[1,1,1,1,1]]

TABLE 3. Some weight matrices with larger rate and column-weight rather than the patterns in [31].

$\mathcal{R}$	$\mathcal{R}'$	w	$g_{\max}$	d_v	d'_v	block design $\mathcal{B}$ and weight design $\mathcal{W}$
0.375	0.25	II	12	3,4	3	$\mathcal{B}$=[[2,3,5],[1,4,5],[3,4,5],[3,4,5],[3,4,5],[3,4,5],[3,4,5],[3,4,5]], $\mathcal{W}$=[[2,1,1],[2,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]
0.286	0.25	II	12	3,4	3	$\mathcal{B}$=[[1,4,5],[2,4,5],[1,2,3],[1,2,3],[1,2,3],[1,2,3],[1,2,3]], $\mathcal{W}$=[[2,1,1],[2,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]
0.58	0.5	I	12	4	3	$\mathcal{B}$=[[1,2,4,5],[2,3,4,5],[1,2,3,5],[2,3,4,5],[2,3,4,5],[2,3,4,5],[2,3,4,5],[2,3,4,5],[1,3,4,5],[1,3,4,5],[1,2,3,5],[2,3,4,5]], $\mathcal{W}$=[[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1]]
0.58	0.5	II	12	3, 4,5	3	$\mathcal{B}$=[[2,3,4],[1,3,4,5],[2,3,4],[3,4,5],[2,3,4],[2,3,4,5],[2,3,4,5],[2,3,4,5],[2,3,4,5],[2,3,4,5],[2,3,4,5],[2,3,5]], $\mathcal{W}$=[[1,1,1],[2,1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1]]
0.66	0.5	I	12	3,4	3	$\mathcal{B}$=[[1,2,3,4],[1,2,3],[1,2,3,4],[1,2,3,4],[1,2,3],[1,2,3,4],[2,3,4],[1,2,3,4],[2,3,4],[1,2,3,4],[1,2,3,4],[1,2,3,4]], $\mathcal{W}$=[[1,1,1,1],[1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1],[1,1,1,1],[1,1,1],[1,1,1,1],[1,1,1],[1,1,1,1],[1,1,1,1],[1,1,1,1]]
0.273	0.25	II	14	3,4	3	$\mathcal{B}$=[[1,3,4],[1,2,5],[2,4,6],[1,3,6],[1,4,7],[2,4,7],[1,6,8],[1,7,8],[2,3,6],[2,3,8],[2,7,8]], $\mathcal{W}$=[[1,1,1],[1,1,2],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1],[1,1,1]]

then, for $w_{2,1} \neq 0$, the slopes in $s_{2,1}$ beginning from $s_{2,1}^{(1)}$ till $s_{2,1}^{(w_{2,1})}$, finally, for $w_{J,L} \neq 0$, ends to find the proper slopes in $s_{J,L}$, from $s_{J,L}^{(1)}$ to $s_{J,L}^{(w_{J,L})}$, sequentially. The part of S which are constructed till step k, are denoted by $S^{(k)}$. For example, $S^{(1)} = (s_{t_1,1}^{(1)})$, where t_1 is the first position in which $w_{t_1,1} \neq 0$. Then, $S^{(2)} = (s_{t_1,1}^{(1)}, s_{t_1,1}^{(2)})$, if $w_{t_1,1} > 1$, otherwise $S^{(2)} = (s_{t_1,1}^{(1)}, s_{t_2,1}^{(1)})$, where t_2 is the second position greater than t_1 in which $w_{t_2,1} \neq 0$. In step $k+1$, when $S^{(k)}$ is constructed inductively, $S^{(k+1)}$ is an expansion of $S^{(k)}$ together with the new slope $s_{p_{k+1},q_{k+1}}^{(r_{k+1})}$, where q_{k+1} is the greatest integer satisfied in $A = \sum_{j=1}^{q_{k+1}-1} \sum_{i=1}^{J} w_{i,j} \leq k+1$, p_{k+1} is the greatest integer satisfying in $B = \sum_{i=1}^{p_{k+1}-1} w_{i,q_{k+1}} \leq k+1-A$ and $r_{k+1} = k+1-A-B$, which is emphasized by the notation $S^{(k+1)} = S^{(k)} \cup \{s_{p_{k+1},q_{k+1}}^{(r_{k+1})}\}$.

In Algorithm 2, it is noticed that to find $S^{(k+1)} = S^{(k)} \cup \{s_{p_{k+1},q_{k+1}}^{(r_{k+1})}\}$, Proposition 2.3 is used (where Eq. 3 is applied for $m = \infty$, i.e., the equality is used instead of a modular relation) to investigate all nonproper slopes $s_{p_{k+1},q_{k+1}}^{'(r_{k+1})}$ such that for $\Delta_{k'}$ defined in Proposition 2.3, we have $\sum_{k'=0}^{l-1} \Delta_{k'} = 0$ for an $l < g/2$, in which the slopes are limited in $S^{(k)} \cup \{s_{p_{k+1},q_{k+1}}^{'(r_{k+1})}\}$. After finding the slope-matrix S, we need to find proper CPM-size m, such that the constructed code with slope-matrix S and CPM-size m has girth at least g. To do this, we need to generate P as the set of all nonproper candidates for CPM-sizes. In fact, it can be seen easily that such nonproper values are all of the integers $\sum_{k=0}^{l-1} \Delta_k$, $2 \leq l \leq g/2$, together with their common divisors, because Eq. 3 holds in modulus of such integers.

Now, to justify Algorithm 2, an example is given to express the details more clear.

Example 4.1: Applying Algorithm 2 on the weight-matrix $W = W(2, 3, 2)$, we have the following slope-matrix with CPM-size 16 corresponding to a type-II QC-LDPC code with girth 6.

$$S = \begin{pmatrix} (0,1) & (0,2) & (0,3) \\ (0,2) & (5,6) & (10,14) \end{pmatrix}$$

For example, in Step 8 of the algorithm, we have constructed $(s_{1,1}^{(1)}, s_{1,1}^{(2)}, s_{2,1}^{(1)}, s_{2,1}^{(2)}, s_{1,2}^{(1)}, s_{1,2}^{(2)}, s_{2,2}^{(1)}) = (0, 1, 0, 2, 0, 2, 5)$ so far. Now, to find a proper $s_{2,2}^{(2)}$, it is sufficient to construct S_8 as all nonproper slopes $s_{2,2}^{(2)}$ such that $s_{2,2}^{(2)} - s_{i_1,2}^{(t_1)} + s_{i_1,j_1}^{(t_2)} - s_{2,j_1}^{(t_3)} = 0$, in which $(i_1, 2, t_1), (i_1, j_1, t_2), (2, j_1, t_3) \in \{(1,1,1), (1,1,2), (2,1,1), (2,1,2), (1,2,1), (1,2,2), (2,2,1), (2,2,2)\}$, and $t_1 \neq 2$ and $t_2 \neq t_3$ if $i_1 = 2$, and $t_1 \neq t_2$ and $t_3 \neq 2$, if $j_1 = 2$. Then, nonproper slopes are as follows.

Algorithm 2

Input: weight-matrix $W = (w_{i,j})_{J\times L}$ and desired girth g, $g \leq g_{\max}(W)$.

Output: slope-matrix S and CPM-size m.

$k \leftarrow 1$, $A \leftarrow \mathbb{Z}^{\geq 0}$, $S_1 \leftarrow \emptyset$, select $s^{(r_1)}_{p_1,q_1} \in A = A \setminus S_1$ arbitrary and set $S^{(1)} = \{s^{(r_1)}_{p_1,q_1}\}$.

for j from 1 to L **do**

  for i from 1 to J **do**

    for t from 1 to $w_{i,j}$ **do**

      Let S_{k+1} be the set of all slopes $s'^{(r_{k+1})}_{p_{k+1},q_{k+1}} \in A$ such that for $\Delta_{k'}$ defined in Proposition 2.3, we have $\sum_{k'=0}^{l-1} \Delta_{k'} = 0$ for an $l < g/2$, in which the slopes are limited in $S^{(k)} \cup \{s'^{(r_{k+1})}_{p_{k+1},q_{k+1}}\}$.

      Select $s^{(r_{k+1})}_{p_{k+1},q_{k+1}} \in A \setminus S_{k+1}$ arbitrary and set $S^{(k+1)} \leftarrow S^{(k)} \cup \{s^{(r_{k+1})}_{p_{k+1},q_{k+1}}\}$, $k \leftarrow k+1$.

    end for

  end for

end for

Let P be the set of all $\sum_{k=0}^{l-1} \Delta_k$, $2 \leq l \leq g/2$, together with their commom divisors and $m \notin P$ be the smallest.

return The slope-matrix S and CPM-size m.

$S_8 = \{s^{(1)}_{1,2} - s^{(1)}_{1,1} + s^{(1)}_{2,1}, s^{(1)}_{1,2} - s^{(1)}_{1,1} + s^{(2)}_{2,1}, s^{(1)}_{1,2} - s^{(2)}_{1,1} + s^{(1)}_{2,1}, s^{(1)}_{1,2} - s^{(2)}_{1,1} + s^{(2)}_{2,1}, s^{(1)}_{1,2} - s^{(2)}_{1,2} + s^{(1)}_{2,2}, s^{(2)}_{1,2} - s^{(1)}_{1,1} + s^{(1)}_{2,1}, s^{(2)}_{1,2} - s^{(1)}_{1,1} + s^{(2)}_{2,1}, s^{(2)}_{1,2} - s^{(2)}_{1,1} + s^{(1)}_{2,1}, s^{(2)}_{1,2} - s^{(2)}_{1,1} + s^{(2)}_{2,1}, s^{(2)}_{1,2} - s^{(1)}_{1,2} + s^{(1)}_{2,2}, s^{(1)}_{2,2} - s^{(1)}_{2,1} + s^{(2)}_{2,1}, s^{(1)}_{2,2} - s^{(2)}_{2,1} + s^{(1)}_{2,1}, s^{(1)}_{2,2}\} = \{-1, 0, 1, 2, 3, 4, 5, 7\}$.

Therefore, a proper value for $s^{(2)}_{2,2}$ is 6. On the other hand, to find CPM-size m, we notice that the set of values $\sum_{k=0}^{1} \Delta_k = s^{(t_1)}_{i_0,j_0} - s^{(t_2)}_{i_1,j_0} + s^{(t_3)}_{i_1,j_1} - s^{(t_4)}_{i_0,j_1}$, for all admissible chains (i_0, j_0, t_1), (i_1, j_0, t_2), (i_1, j_1, t_3), (i_0, j_1, t_4) is $Q = \{\pm 1, \pm 2, \ldots, \pm 15\}$. Then, P, the set of common divisors of elements of Q, is equal to Q and so the smallest CPM-size is $m = \min(\mathbb{Z}^{\geq 0} \setminus P) = 16$.

A. THE COMPLEXITY OF THE ALGORITHM

In Algorithm 2, we accept the notions $w^{(k+1)}_{\max}$, $J^{(k+1)}_{\max}$ and $r^{(k+1)}_{\max}$ to be the maximum values of entries in the weight-matrix, maximum column-weight and maximum row-weight of W, respectively. Now, in step k, when $S^{(k)}$ is constructed, to construct $S^{(k+1)}$, we must generate all of the chains (i_0, j_0, t_0); (i_1, j_0, t_1); (i_1, j_1, t_2); (i_2, j_1, t_3); $\ldots$; $(i_{l-1}, j_{l-1}, t_{2l-2})$; (i_l, j_{l-1}, t_{2l-1}); $(i_l, j_l, t_{2l}) = (i_0, j_0, t_0)$ which are listed as the indices of the slopes in $S^{(k)} \cup \{s^{(r_{k+1})}_{p_{k+1},q_{k+1}}\}$, starting from $(i_0, j_0, t_0) = (p_{k+1}, q_{k+1}, r_{k+1})$. However, after assumption $(i_0, j_0, t_0) = (p_{k+1}, q_{k+1}, r_{k+1})$, to generate all possible (i_1, j_0, t_1), two cases can be considered. If $i_1 = i_0$, then $t_1 \neq t_0$, which is enumerated at most $w^{(k+1)}_{\max} - 1$, and if $i_1 \neq i_0$, enumerated at most $J^{(k+1)}_{\max} - 1$, then the possible values for t_1 is at most $w^{(k+1)}_{\max}$. These two cases are enumerated in at most $(w^{(k+1)}_{\max} - 1) + (J^{(k+1)}_{\max} - 1) w^{(k+1)}_{\max} = J^{(k+1)}_{\max} w^{(k+1)}_{\max} - 1$. After selection (i_1, j_0, t_1), to generate all possible (i_1, j_1, t_2), two cases can be considered. If $j_1 = j_0$, then $t_2 \neq t_1$, which is enumerated at most $(w^{(k+1)}_{\max} - 1)$, and if $j_1 \neq j_0$, selected at most $r^{(k+1)}_{\max} - 1$, then the possible values for t_1 is at most $w^{(k+1)}_{\max}$. These two cases are combined in at most $(w^{(k+1)}_{\max} - 1) + (r^{(k+1)}_{\max} - 1) w^{(k+1)}_{\max} = r^{(k+1)}_{\max} w^{(k+1)}_{\max} - 1$.

This process repeats l times successively to traverse all admissible chains in at most $l w^{(k+1)}_{\max}(r^{(k+1)}_{\max} + J^{(k+1)}_{\max}) o(2l)$, because the complexity to find nonproper slopes $s^{(r_{k+1})}_{p_{k+1},q_{k+1}}$ in Eq. 3 is at most $o(2l)$. Then, the overall complexity to accept a slope-matrix as the solution is at most $\sum_{l=2}^{g-2} \sum_{k=1}^{n} l w^{(k)}_{\max}(r^{(k)}_{\max} + J^{(k)}_{\max}) o(2l)$, where $n = \sum_{i,j} w_{i,j}$, because $2 \leq l \leq g-2$ and the required steps to find the solution is n. But, this is at most $w^{(n)}_{\max}(r^{(n)}_{\max} + J^{(n)}_{\max}) o(ng^2)$ which shows a polynomial complexity, i.e., the algorithm is efficient. It is noticed that, if one uses a depth-first search to find a proper slope-matrix as the solution when the CPM-size m is given, the complexity is at most $w^{(n)}_{\max}(r^{(n)}_{\max} + J^{(n)}_{\max}) o(ng^2 m^{n+2})$ which enlarges exponentially, when n enlarges.

It is noticed that the approach of Algorithm 2 is not a backtracking search, because otherwise if there is not any candidate for $s^{(r_{k+1})}_{p_{k+1},q_{k+1}}$ in Step $k+1$, i.e., $S_{k+1} = \mathbb{Z}$, then we have an inevitable chain in $S^{(k)} \cup \{s^{(r_{k+1})}_{p_{k+1},q_{k+1}}\}$ which is a contradiction, because there are no inevitable chains of length $2l$, $l \geq g/2$. Moreover, although the approach to find slope values or the CPM-size is to find the smallest nonnegative integers not belonging to the sets S_{k+1}, $k \leq 0$, and P, respectively, we can select the values arbitrarily in the complement of the sets which may have some advantages to achieve smaller CPM-sizes. Applying this random approach by Algorithm 2, Tables 4-12 provide some proper $J \times L$ slope-matrices S such that the corresponding multitype QC-LDPC codes with CPM-size m have girth g, $g \leq 16$. In tables, each weight-matrix W is denoted by the block-design $\mathcal{B}$ and the weigh-design $\mathcal{W}$. Moreover, because of the space limitation, the desired slope-matrix $S = (s_{i,j})$ is denoted by the following *slope-vector*.

$$S = (s^{(2)}_{b_{1,1},1}, \ldots, s^{(w_{b_{1,1},1})}_{b_{1,1},1}, s^{(1)}_{b_{1,2},1}, \ldots, s^{(w_{b_{1,2},1})}_{b_{1,2},1}, \ldots, s^{(1)}_{b_{1,m_1},1}, \ldots, s^{(w_{b_{1,m_1},1})}_{b_{1,m_1},1}, \ldots, s^{(1)}_{b_{L,m_L},L}, \ldots, s^{(w_{b_{L,m_L},L})}_{b_{L,m_L},L}),$$

in which $s^{(1)}_{b_{i,1},i} = 0$, $1 \leq i \leq L$, which are not appended to S.

Example 4.2: Applying Algorithm 2 for the weight-matrix W, a type-II QC-LDPC code with girth $g = 8$ and CPM-size $m = 24$ can be derived with PCM H.

$$W = \begin{pmatrix} 2 & 1 & 1 & 1 & 0 \\ 2 & 1 & 1 & 0 & 1 \\ 0 & 2 & 0 & 1 & 2 \\ 0 & 0 & 2 & 2 & 1 \end{pmatrix},$$

TABLE 4. Some constructed type-II QC-LDPC code with girth $g = 6, 8$ against type-I QC-LDPC code in [18].

L	g	m	m'	$d_{\min}$	$d'_{\min}$	slope-vector	block design $\mathcal{B}$ and weight design $\mathcal{W}$
4	6	23	23	24	22	[13,8,11,18,5,9,19,1]	$\mathcal{B}$=[[1,2],[2,3],[1,3],[1,2,3]], $\mathcal{W}$=[[2,1],[2,1],[1,2],[1,1,1]]
4	6	21	23	22	22	[6,0,13,8,2,17,5,18]	$\mathcal{B}$=[[1,2],[2,3],[1,3],[1,2,3]], $\mathcal{W}$=[[2,1],[2,1],[1,2],[1,1,1]]
4	8	9	9	8	6	[1,0,2,0,1,5,4,3]	$\mathcal{B}$=[[1,2],[2,3],[1,3],[1,2,3]], $\mathcal{W}$=[[2,1],[2,1],[1,2],[1,1,1]]
4	8	26	29	24	24	[8,25,11,17,20,22,20,10]	$\mathcal{B}$=[[1,2],[2,3],[1,3],[1,2,3]], $\mathcal{W}$=[[2,1],[2,1],[1,2],[1,1,1]]

$$H = \begin{pmatrix} I^0+I^1 & I^0 & I^0 & I^0 & 0 \\ I^0+I^3 & I^7 & I^{13} & 0 & I^0 \\ 0 & I^0+I^2 & 0 & I^5 & I^4+I^{10} \\ 0 & 0 & I^0+I^5 & I^9+I^{22} & I^{18} \end{pmatrix}.$$

In this case, the corresponding slope-vector is

$$S = [1, 0, 3, 7, 0, 2, 13, 0, 5, 5, 9, 22, 4, 10, 18].$$

B. SOME RESULTS AND ALGORITHM OUTPUTS

Related to the design of multitype QC-LDPC codes, improving some of the parameters, such as the CPM-size, rate, minimum-distance, or cycle distribution, is desirable. In fact, in the class of QC-LDPC codes with the same girths and column-weight distributions, finding codes with smaller CPM-sizes, larger rates or minimum-distances, or better cycle distributions is of favor. In continue, applying Algorithm 2, we give some examples of multitype QC-LDPC codes having better parameters than the codes in [7], [8], [16], [18], [20], [23]–[25], [29], [31]. Moreover, some 4-cycle free multitype QC-LDPC codes are constructed whose CPM-sizes are close to the lower-bound proposed recently in [5]. To find the minimum-distance of the constructed codes, Magma software [37] is used. As Algorithm 1 and Algorithm 2 produce a variety class of the weight and slope-matrices, the outputs have the potential to attain codes with flexibility parameters, such as length, rate, and minimum-distance.

1) TYPE I,II QC-LDPC CODES

Example 4.3: Applying Algorithm 2 on the following weight-matrix W, matrices H_1 and H_2 are constructed as the PCM of two girth-6 (3,4)-regular type-II QC-LDPC codes with CPM-size 23 and CPM-size 21, respectively.

$$W = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 2 & 1 \end{pmatrix},$$

$$H_1 = \begin{pmatrix} \mathcal{I}+\mathcal{I}^{13} & 0 & \mathcal{I} & \mathcal{I} \\ \mathcal{I}^8 & \mathcal{I}+\mathcal{I}^{11} & 0 & \mathcal{I}^{19} \\ 0 & \mathcal{I}^{18} & \mathcal{I}^5+\mathcal{I}^9 & \mathcal{I}^1 \end{pmatrix},$$

$$H_2 = \begin{pmatrix} \mathcal{I}+\mathcal{I}^{6} & 0 & \mathcal{I} & \mathcal{I} \\ \mathcal{I} & \mathcal{I}+\mathcal{I}^{13} & 0 & \mathcal{I}^{5} \\ 0 & \mathcal{I}^{8} & \mathcal{I}^2+\mathcal{I}^{17} & \mathcal{I}^{18} \end{pmatrix}.$$

The corresponding code with PCM H_1 has minimum-distance 24, whereas the (3,4)-regular type-I QC-LDPC code in [18] with the same CPM-size, column-weight, and girth has minimum-distance 22. Moreover, the code with PCM H_2 has CPM-size 21 which is shorter than the (3,4)-regular type-I QC-LDPC code in [18] with the same minimum-distance, column-weight, and girth having CPM-size 23.

In addition to the codes in Example 4.3 (shown as the first two rows of Table 4), Table 4 provides some other comparisons between type-II QC-LDPC codes constructed by Algorithm 2 and type-I QC-LDPC codes presented in [18] with the same column-weight, rate, and girth. As the table shows, the constructed codes have smaller CPM-size m or larger minimum-distance $d_{\min}$ rather than those of codes in [18], denoted by m' and $d'_{\min}$, respectively.

Example 4.4: In [16], a [184, 47] type-II QC-LDPC code with rate 0.2554 and CPM-size 46 is constructed which attains the optimal minimum-distance 32. Against, Algorithm 2 can be used to generate a girth-6 [184, 49, 32] type-II QC-LDPC code with the following PCM.

$$H = \begin{pmatrix} \mathcal{I}+\mathcal{I}^{2} & \mathcal{I}+\mathcal{I}^{30} & 0 & 0 \\ \mathcal{I}^{16} & \mathcal{I}^{37} & \mathcal{I} & \mathcal{I} \\ 0 & 0 & \mathcal{I}^{37}+\mathcal{I}^{34} & \mathcal{I}^{17}+\mathcal{I}^{44} \end{pmatrix},$$

The constructed code has rate 0.2663 and dimension 49 which are better than those parameters of the code in [16] with the same CPM-size and minimum-distance.

Example 4.5: The structure of the code's Tanner graph and especially the distribution of short cycles, are very effective factors in the code performance. Applying Algorithm 2, the following matrix is generated as the PCM of a type-II QC-LDPC code C with CPM-size 46 and girth 8.

$$H = \begin{pmatrix} \mathcal{I}+\mathcal{I}^{31} & 0 & \mathcal{I} & \mathcal{I} \\ \mathcal{I}^{34} & \mathcal{I} & \mathcal{I}^{17}+\mathcal{I}^{29} & 0 \\ 0 & \mathcal{I}^{28}+\mathcal{I}^{3} & 0 & \mathcal{I}^{4}+\mathcal{I}^{39} \end{pmatrix},$$

TABLE 5. Cycle multiplicities of the constructed QC-LDPC code of length n against the code in [16].

Cycle Length	8		10	
n	C	[16]	C	[16]
184	92	138	736	828

Table 5 provides the (6, 8)−cycle multiplicities of code C in comparison with a type-II QC-LDPC code with the same counterparts in [16]. As the table shows, the number of short cycles of code C is less than the corresponding code cycle enumerations in [16].

Tables 6 and 7 provide a CPM-size comparison between the constructed type-II QC-LDPC codes with weight-matrix $W(J, L; 2)$, and some type-II QC-LDPC codes with the same weight-matrix in [23], [24] and [25]. As the outputs show, not only the constructed codes have smaller CPM-sizes than the

TABLE 6. CPM-size *m* of the constructed codes against CPM-sizes *m′* and *m″* of the codes in [23] and [24], respectively and a lower-bound *LB* on CPM-size in [5].

(J, L)	m	m' [23]	m'' [24]	LB [5]	slope-vector
(2,3)	12	21	23	12	[1,0,2,3,6,10,4,8,9]
(2,4)	16	36	23	16	[1,0,2,2,7,11,4,8,14,6,3,12]
(2,5)	20	55	23	20	[1,0,2,2,5,8,3,10,14,4,16,17,6,4,15]
(2,6)	24	78	35	24	[1,0,2,2,5,6,5,13,19,3,12,20,4,11,22,6,16,21]
(2,7)	28	105	57	28	[1,0,2,2,5,6,3,10,14,5,13,21,4,23,26,6,15,24,8,20,25]
(2,8)	32	136	71	32	[1,0,2,2,5,6,3,10,14,4,13,19,6,18,27,5,25,28,7,24,29,10,8,26]
(3,3)	13	35	35	12	[1,0,3,0,4,3,4,8,10,11,4,10,11,6,9]
(3,4)	20	63	57	16	[1,0,2,0,3,2,5,6,10,14,3,10,14,1,16,4,12,17,9,11]
(3,5)	25	99	57	20	[1,0,2,0,3,2,5,6,10,14,3,11,16,1,18,4,14,21,9,20,7,19,22,11,13]
(3,6)	30	143	57	24	[1,0,2,0,3,2,5,6,10,14,3,10,15,4,25,4,13,21,9,27,6,14,24,21,26,5,25,28,11,18]
(3,7)	35	195	57	28	[1,0,2,0,3,2,5,6,10,14,3,10,14,1,7,4,12,17,23,31,5,23,29,11,18,6,21,32,15,28,8,27,30,5,29]

TABLE 7. CPM-size *m* and rate *R* of the constructed codes against CPM-size *m′* and rate *R′* of codes in [25].

L	m	m' [25]	R	R' [25]	slope-vector
3	12	133	0.38	0.34	[1,0,2,3,6,10,4,8,9]
4	16	273	0.53	0.50	[1,0,2,2,7,11,4,8,14,6,3,12]
5	20	381	0.62	0.60	[1,0,2,2,5,8,3,10,14,4,16,17,6,4,15]
6	24	553	0.68	0.67	[1,0,2,2,5,6,5,13,19,3,12,20,4,11,22,6,16,21]

TABLE 8. CPM-size *m* of the constructed type-I QC-LDPC codes against CPM-size *m′* of the codes in [20].

(J,L)	girth	m	m'	slope-vector
(6,12)	6	2	4	[1,0,1,1,0,0,0,0,0,1,1,0,0,1,1,0,0,1]
	8	6	8	[1,0,1,5,3,1,4,3,2,5,5,4,0,5,3,4,3,4]
	10	19	24	[10,11,17,17,2,5,4,12,2,2,9,8,8,6,16,7,1,8]
	12	58	66	[0,0,0,0,1,0,1,3,2,10,5,10,18,26,30,38,33,41]
	14	267	289	[0,0,0,0,1,0,2,6,5,19,10,26,38,74,60,122,49,154]
(6,24)	6	6	13	[0,0,0,0,0,0,0,0,1,0,1,0,1,2,1,2,2,1,0,1,3,4,1,3,1,3,2,5,3,2,2,1,3,1,3,4,4,3,2,2,2,2]
	8	24	45	[0,0,0,0,1,0,1,1,1,3,2,4,3,5,4,6,4,11,2,7,6,15,8,5,10,3,12,18,9,21,17,13,14,8,14,22,16,15,17,13,10,12]
(6,54)	6	13	27	[9,6,3,0,3,2,10,11,4,4,7,4,3,4,6,2,7,8,11,6,2,7,8,0,12,7,1,6,0,2,9,3,11,7,11,10,8,10,11,12,6,10,3,1,11,11,8,9,5,5,9,9,10,5,7,0,11,9, 2,1,1,10,9,5,1,0,3,9,1,8,10,4,0,1,8,2,6,12,5,3,6,0,2,4,12,3,8,10,3,6,7,2,5,9,0,5,1,4,8]

TABLE 9. Some type-II QC-LDPC codes with girth 8 achieving the upper-bound $U = U_{d_{min}}(H)$ on the minimum-distance.

(J, L)	m	girth	block design $\mathcal{B}$	weight design $\mathcal{W}$	slope-vector	U	$d_{\min}$
(2,3)	53	8	$\mathcal{B}$=[[1,2],[1,2],[1,2]],	$\mathcal{W}$=[[2,2],[2,2],[2,2]]	[3,0,2,4,11,24,16,35,41]	24	24
(3,4)	46	8	$\mathcal{B}$=[[1,2],[2,3],[1,2],[1,3]],	$\mathcal{W}$=[[2,1],[1,2],[1,2],[1,2]]	[10,26,18,13,4,43,16,14]	32	32
(3,4)	45	8	$\mathcal{B}$=[[1,2],[2,3],[1,2],[1,3]],	$\mathcal{W}$=[[2,1],[1,2],[1,2],[1,2]]	[31,19,40,33,1,9,22,20]	32	32
(3,4)	45	8	$\mathcal{B}$=[[1,2],[2,3],[1,3],[1,2,3]],	$\mathcal{W}$=[[2,1],[2,1],[1,2],[1,1,1]]	[3,8,20,33,13,34,43,42]	30	30
(3,4)	44	8	$\mathcal{B}$=[[1,2],[2,3],[1,3],[1,2,3]],	$\mathcal{W}$=[[2,1],[2,1],[1,2],[1,1,1]]	[10,7,38,34,2,10,30,29]	30	30
(3,4)	30	8	$\mathcal{B}$=[[1,2,3],[1,2,3],[1,2,3],[1,2,3]],	$\mathcal{W}$=[[1,1,1],[1,1,1],[1,1,1],[1,1,1]]	[0,15,7,20,12,26,25,24]	24	24

codes in [23]–[25], but also the rate of our codes in Table 7 is better.

Moreover, for type-I QC-LDPC codes with the following block-designs $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ associated to the base matrices in [20], a comparison is provided in Table 8 between the CPM-size of the constructed codes and the CPM-size of the codes in [20]. As the table shows, the constructed codes have a smaller CPM-size.

$\mathcal{B}_1 =$ [[1, 3, 5], [2, 4, 6], [1, 3, 6], [2, 4, 5], [1, 4, 5], [2, 3, 6], [1, 2], [1, 2], [3, 4], [3, 4], [5, 6], [5, 6]],

$\mathcal{B}_2 =$ [[1, 2, 4], [1, 3, 5], [4, 5, 6], [2, 3, 6], [1, 4, 6], [2, 3, 5], [1, 2, 4], [3, 5, 6], [1, 2, 5], [3, 4, 6], [1, 2, 5], [3, 4, 6], [1, 3, 5], [2, 4, 6], [1, 4, 5], [2, 3, 6], [1, 3, 6], [2, 4, 5], [1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [1, 6]],

$\mathcal{B}_3 =$ [[1, 3, 5], [2, 4, 6], [1, 3, 5], [2, 4, 6], [1, 2, 4], [3, 5, 6], [1, 2, 5], [3, 4, 6], [1, 2, 4], [3, 5, 6], [1, 2, 3], [4, 5, 6], [1, 4, 6], [2, 3, 5], [1, 2, 4], [3, 5, 6], [1, 4, 5], [2, 3, 6], [1, 2, 3], [4, 5, 6], [1, 4, 5], [2, 3, 6], [1, 5, 6], [2, 3, 4], [1, 4, 5], [2, 3, 6], [1, 2, 6], [3, 4, 5], [1, 2, 4], [3, 5, 6], [1, 3, 4], [2, 5, 6], [1, 2, 4], [3, 5, 6], [1, 4, 5], [2, 3, 6], [1, 3, 5], [2, 4, 6], [1, 3, 5], [2, 4, 6], [1, 2, 6], [3, 4, 5], [1, 2, 4], [3, 5, 6], [1, 5, 6], [1, 4], [2, 5], [3, 6], [1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [1, 6]].

According to the minimum-distance of multitype QC-LDPC codes, Eq. 1 and Eq. 2 present two upper-bounds on the minimum-distance in [16]. By the way, the experiments show that these upper-bounds are very close to the minimum-distance when the CPM-size is large enough. In fact, by increasing the boundary of the minimum-distance, we hope to gain a larger minimum-distance. For example, according to the PCM H of some type-II QC-LDPC codes with girth 8, Table 9 provides some codes whose minimum-distances achieving the bound $U_{dmin}(H)$.

TABLE 10. Some irregular type-II QC-LDPC codes with girths g, column-weights d_v, and minimum-distance upper-bound $U = U_{d_{\min}}(H)$.

(J, L)	m	g	d_v	block design $\mathcal{B}$ and weight design $\mathcal{W}$	slope-vector	U
(5,13)	10830	10	5,6	$\mathcal{B}$=[1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], [1, 2, 3, 4, 5], $\mathcal{W}$=[1, 2, 1, 1, 1], [2, 1, 1, 1, 1], [1, 1, 2, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 2, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 2], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1]	[0, 1, 0, 0, 0, 3, 11, 23, 28, 37, 43, 88, 136, 177, 216, 52, 203, 281, 423, 83, 366, 476, 702, 912, 154, 528, 989, 1260, 358, 910, 1632, 2205, 498, 1170, 2216, 2833, 578, 1539, 2696, 3630, 4825, 648, 2258, 3858, 5303, 980, 2389, 4572, 6358, 1027, 2970, 5483, 8058, 1353, 3760, 6920, 9688]	720
(6,12)	1710	10	4,5	$\mathcal{B}$=[2, 4, 5, 6], [1, 4, 5, 6], [1, 3, 5, 6], [2, 3, 4, 6], [3, 4, 5, 6], [1, 4, 5, 6], [2, 3, 5, 6], [1, 4, 5, 6], [1, 2, 5, 6], [1, 3, 4, 5], [2, 4, 5, 6], [2, 4, 5, 6] $\mathcal{W}$=[1, 1, 1, 1], [1, 2, 1, 1], [2, 1, 1, 1], [2, 1, 1, 1], [2, 1, 1, 1], [1, 1, 2, 1], [1, 1, 1, 1], [1, 1, 1, 2], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]	[0, 0, 0, 0, 1, 4, 9, 6, 0, 20, 38, 7, 4, 44, 27, 19, 69, 103, 153, 52, 143, 211, 309, 39, 261, 403, 66, 246, 497, 693, 42, 328, 591, 84, 389, 807, 151, 623, 1239, 175, 887, 1543]	306
(6,13)	6338	10	5,6	$\mathcal{B}$=[1, 2, 3, 5, 6], [1, 2, 4, 5, 6], [1, 3, 4, 5, 6], [1, 2, 3, 4, 5], [1, 3, 4, 5, 6], [2, 3, 4, 5, 6], [2, 3, 4, 5, 6], [1, 2, 4, 5, 6], [1, 2, 4, 5, 6], [2, 3, 4, 5, 6], [2, 3, 4, 5, 6], [1, 3, 4, 5, 6], [1, 3, 4, 5, 6], $\mathcal{W}$=[2, 1, 1, 1, 1], [1, 2, 1, 1, 1], [1, 1, 1, 2, 1], [1, 1, 2, 1, 1], [1, 1, 1, 1, 2], [1, 1, 2, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1], [1, 1, 1, 1, 1]	[1, 0, 0, 0, 0, 3, 9, 0, 17, 28, 12, 16, 48, 75, 112, 43, 20, 90, 139, 227, 97, 205, 355, 486, 605, 52, 284, 487, 660, 1003, 102, 513, 911, 1501, 196, 746, 1231, 1993, 252, 900, 1711, 2568, 120, 1145, 2213, 3338, 181, 1295, 2716, 4038, 314, 1660, 3529, 4932, 410, 2002, 3944, 5877]	1632
(4,8)	800	12	3,4	$\mathcal{B}$=[1, 2, 3], [2, 3, 4], [2, 3, 4], [2, 3, 4], [2, 3, 4], [2, 3, 4], [2, 3, 4], [2, 3, 4] $\mathcal{W}$=[2, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1]	[1, 0, 0, 4, 0, 10, 13, 22, 54, 49, 118, 120, 254, 190, 474, 359, 651]	48
(9,13)	2406	12	4,5	$\mathcal{B}$=[6, 7, 8, 9], [1, 3, 4, 6], [2, 5, 8, 9], [1, 7, 8, 9], [2, 3, 7, 9], [2, 4, 7, 9], [1, 2, 5, 6], [2, 4, 7, 8], [1, 2, 5, 6], [1, 3, 5, 6], [4, 5, 7, 9], [2, 5, 7, 9], [1, 3, 5, 6] $\mathcal{W}$=[2, 1, 1, 1], [1, 1, 2, 1], [1, 1, 1, 1], [2, 1, 1, 1], [1, 2, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]	[2, 0, 0, 0, 0, 0, 3, 0, 0, 0, 7, 8, 20, 33, 59, 1, 18, 65, 142, 22, 93, 242, 56, 80, 248, 152, 383, 623, 122, 322, 574, 194, 371, 695, 35, 610, 1051, 220, 914, 1669, 302, 857, 1768]	2824
(8,14)	4861	12	4,5	$\mathcal{B}$=[1, 4, 6, 7], [2, 5, 6, 8], [1, 2, 3, 8], [3, 4, 5, 7], [2, 3, 6, 8], [2, 3, 6, 8], [2, 3, 6, 8], [1, 4, 5, 7], [1, 4, 5, 7], [2, 3, 6, 8], [1, 4, 5, 7], [2, 3, 6, 8], [2, 4, 7, 8], [1, 4, 5, 7] $\mathcal{W}$=[1, 1, 2, 1], [1, 2, 1, 1], [2, 1, 1, 1], [2, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]	[0, 0, 2, 0, 0, 3, 0, 0, 7, 20, 0, 28, 12, 1, 60, 76, 22, 113, 185, 38, 238, 409, 60, 485, 703, 85, 316, 279, 130, 522, 701, 424, 1148, 1895, 338, 967, 1401, 732, 2003, 3190, 841, 1653, 3508, 924, 2248, 3320]	480
(11,24)	4178	14	3,4	$\mathcal{B}$=[7, 9, 11], [8, 10, 11], [6, 7, 10], [3, 8, 9], [3, 4, 10], [1, 5, 8], [1, 5, 10], [1, 2, 6], [1, 2, 3], [3, 5, 11], [5, 6, 9], [1, 3, 7], [4, 5, 7], [2, 5, 10], [3, 6, 11], [2, 5, 11], [1, 4, 10], [3, 4, 5], [2, 4, 7], [1, 4, 11], [2, 3, 10], [4, 6, 8], [2, 7, 8], [2, 4, 9] $\mathcal{W}$=[1, 2, 1], [2, 1, 1], [2, 1, 1], [1, 1, 1]	[0, 4, 0, 1, 1, 0, 5, 0, 17, 0, 14, 0, 24, 0, 0, 6, 39, 0, 35, 9, 58, 35, 96, 99, 165, 99, 242, 16, 237, 94, 360, 291, 466, 339, 796, 121, 660, 273, 743, 449, 1242, 526, 1582, 868, 2022, 1294, 2014, 1531, 2822, 1648, 3423]	106
(14,20)	1000	14	3,4	$\mathcal{B}$=[6, 7, 10], [1, 5, 7], [6, 12, 14], [8, 10, 12], [9, 13, 14], [6, 11, 13], [2, 11, 12], [10, 11, 14], [7, 12, 13], [5, 10, 13], [2, 4, 7], [1, 7, 11], [2, 5, 6], [3, 8, 13], [2, 9, 10], [1, 8, 10], [8, 9, 12], [8, 9, 11], [5, 8, 14], [3, 4, 5] $\mathcal{W}$=[2, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 2], [1, 1, 1], [1, 2, 1], [1, 1, 1], [1, 1, 1], [2, 1, 1], [1, 2, 1], [1, 1, 1], [1, 1, 1], [2, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1]	[475, 711, 392, 106, 402, 930, 750, 430, 869, 430, 288, 497, 554, 587, 359, 187, 799, 144, 43, 611, 553, 248, 30, 65, 676, 282, 621, 800, 840, 778, 220, 544, 805, 514, 591, 272, 477, 668, 772, 535, 852, 557, 103, 973, 257, 178]	1512
(12,15)	900	16	3,4	$\mathcal{B}$=[3, 5, 6], [3, 8, 9], [2, 9, 11], [6, 7, 12], [3, 11, 12], [3, 10, 11], [7, 8, 9], [5, 8, 11], [1, 5, 6], [1, 3, 4], [5, 9, 10], [5, 10, 12], [1, 8, 11], [1, 4, 10], [2, 4, 6], $\mathcal{W}$=[1, 1, 1], [1, 1, 1], [2, 1, 1], [1, 2, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1], [1, 1, 1]	[33, 815, 391, 30, 766, 773, 181, 116, 471, 501, 356, 814, 154, 421, 717, 343, 431, 860, 31, 832, 504, 874, 313, 278, 591, 439, 526, 743, 567, 162, 309, 3]	458

In [31], some column-weight three type-II QC-LDPC codes with girth at most 14 are constructed based on some regular block designs, such as *balanced ternary designs*, *pair-wise balanced design*, and *difference triangle sets*. In general, increasing the column-weights often leads to decreasing the maximum-achievable girth. Table 10 has provided some irregular type-II QC-LDPC codes with girths at most 16 and column-weights $d_v \geqslant 3$.

In continue, we give some examples in which the constructed codes have smaller CPM-size rather than the codes in [31] with the same weight matrices.

Example 4.6: For the 6×12 and 6×8 weight-matrices W_1 and W_2 given by (6), Algorithm 2 can be used to generate the slope-vectors $S_1 = [1,0,0,4,0,6,0,9,0,11,13,33,44,67,75,72,103,148,85,171,109,272,196,62]$ and $S_2 = [46,0,2,32,12,52,47,52,47,15,44,29,38,40,21,36]$, corresponding to some type-II QC-LDPC codes with girth 12 having CPM-sizes 322 and 55, respectively. Against, two type-II QC-LDPC codes in [31] with the same weight matrices have constructed with CPM-sizes 600 and 100, respectively, which are larger than the constructed codes.

Example 4.7: Consider the following 9×12 and 9×15 weight matrices W_3 and W_4, respectively, reported in [31].

$$W_3 = \begin{pmatrix} 2&0&0&0&0&0&0&0&0&1&0&1 \\ 0&2&0&0&0&0&0&0&1&0&1&0 \\ 0&0&2&0&0&0&0&1&0&0&0&1 \\ 0&0&0&2&0&0&0&1&1&0&0&0 \\ 0&0&0&0&2&0&0&0&1&1&0&0 \\ 0&0&0&0&0&2&0&0&0&1&1&0 \\ 0&0&0&0&0&0&2&0&0&0&1&1 \\ 1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&0&0&0&0 \end{pmatrix}, \tag{7}$$

$$W_4 = \begin{pmatrix} 2&0&0&0&0&0&0&1&0&0&1&0&0&1&0 \\ 0&2&0&0&0&0&0&1&0&0&0&0&1&0&1 \\ 0&0&2&0&0&0&0&0&1&0&1&0&0&0&1 \\ 0&0&0&2&0&0&0&0&1&0&0&1&1&0&0 \\ 0&0&0&0&2&0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&2&0&0&0&1&0&0&1&1&0 \\ 0&0&0&0&0&0&2&0&0&0&0&1&0&1&1 \\ 1&1&1&1&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&1&0&0&0&0&0&0 \end{pmatrix}$$

Applying Algorithm 2, two type-II QC-LDPC codes C_3 and C_4 with girth 14 are constructed with the slope-vectors S_3= [92, 81, 82, 104, 158, 58, 63, 60, 67, 98, 113, 85, 95, 107,

TABLE 11. The rate of some 4-cycle free type-w, $w = 3, 4$, QC-LDPC codes with CPM-size m and lower-bound LB of CPM-sizes in [5].

		Type-III QC-LDPC codes			Type-IV QC-LDPC codes		
L	rate	m	LB [5]	slope-vector	m	LB [5]	slope-vector
2	0.5	13	12	[4,3,6,8]	26	24	[22,12,7,6,9,8]
3	0.66	19	18	[14,8,9,2,3,18]	37	36	[4,15,9,16,14,13,17,25,7]
4	0.75	25	24	[22,23,8,13,4,18,9,19]	49	48	[7,4,23,31,36,2,43,32,8,21,12,22]
5	0.8	31	30	[1,3,4,11,5,15,6,18,8,17]	61	60	[55,24,16,49,11,1,5,9,34,47,19,26,3,20,18]
6	0.83	37	36	[28,35,14,17,29,24,19,15,26,1,16,10]	73	72	[58,17,70,29,55,68,16,23,59,52,48,42,40,2,51,8,9,54]
7	0.86	43	42	[1,3,39,23,32,25,6,34,38,12,29,19,30,22]	85	84	[68,79,54,80,38,22,61,9,65,32,13,39,3,1,51,8,23,44,45,73,55]
8	0.875	49	48	[34,41,23,32,46,18,1,13,22,20,11,16,10,45,19, 43]	97	96	[16,73,15,75,48,67,85,11,29,55,62,65,50,45,36,54,28,91,31,33,77,25,21, 38]
9	0.88	55	54	[45,19,18,22,43,34,7,32,27,13,20,5,2,54,24,16, 17,6]	111	108	[81,10,57,48,14,83,103,80,58,96,98,99,39,90,85,84,43,52,25,61,6,38,67, 49,33,16,107]
10	0.9	61	60	[16,27,51,32,9,53,33,36,37,60,41,54,15,55,22, 4,12,14,26,31]	121	120	[67,86,14,16,93,97,99,69,66,65,34,76,7,111,36,37,78,60,2,27,59,15,20, 115,1,9,48,70,58,108]

140, 88, 148, 154, 44, 69, 107, 29, 76, 33] and S_4=[366, 587, 332, 555, 481, 617, 205, 563, 256, 358, 388, 617, 250, 604, 390, 541, 15, 154, 430, 442, 66, 623, 535, 248, 124, 305, 52, 328, 209, 233], with CPM-sizes 165 and 650, respectively, whereas the CPM-sizes of the corresponding codes in [31] with the same weight-matrix and the same girth are 300 and 1000, respectively, which are larger than CPM-size of C_3 and C_4.

Example 4.8: In [7], the authors have constructed some symmetric type-I QC-LDPC codes with girth 8, for example a (4, 5)-regular code with weight-matrix $W = W(4, 5, 1)$, CPM-size 23 and minimum-distance 24. Against, using Algorithm 2 for weight-matrix $W = \begin{pmatrix} 1&2&1&1&0 \\ 2&1&1&0&1 \\ 1&0&2&1&1 \\ 0&1&0&2&2 \end{pmatrix}$, the slope-vector $S = [7, 13, 17, 13, 8, 17, 11, 2, 1, 10, 1, 19, 1, 2, 21]$ is generated in which the corresponding code with CPM-size 23 has minimum-distance 33. Then, the constructed code has better minimum-distance than the code in [7], where length, column/row weight, and girth are the same.

Example 4.9: Applying Algorithm 2 on the 3×4 weight-matrix $W = \begin{pmatrix} 2&0&1&1 \\ 0&1&2&1 \\ 1&2&0&1 \end{pmatrix}$, three slope-vectors [1, 1, 1, 2, 0, 1, 2, 2], [8, 5, 7, 5, 2, 6, 4, 1] and [18, 25, 14, 15, 7, 18, 5, 0] are generated such that the corresponding type-II QC-LDPC codes with girths 6, 8 and 10 and CPM-sizes 3, 9 and 28 have minimum-distances 6, 8 and 24, respectively. This is whereas the minimum CPM-sizes and minimum-distances of the corresponding type-I codes in [8] with the same column/row weights and girths are 3, 9, 37 and 6, 6, 14, respectively. Moreover, for the 3×6 weight-matrix $W = \begin{pmatrix} 1&1&1&1&1&1 \\ 1&1&1&0&2&1 \\ 1&1&1&2&0&1 \end{pmatrix}$, the slope-vector [4, 14, 5, 6, 10, 4, 8, 9, 1, 9, 6, 12] is generated by Algorithm 2 with CPM-size 18 and girth 8, for which the constructed code has minimum-distance 12, whereas the corresponding code in [8] with the same column/row weight, girth, and length has minimum-distance 10. Then, the constructed codes have better minimum-distance, whereas the CPM-sizes of the constructed codes are not greater.

2) TYPE-III, IV QC-LDPC CODES

First, we know that the existence of element 3 in the weight-matrix induces inevitable 6-cycles in the Tanner graph of the corresponding code [31], therefore, in this subsection, we consider type-III, IV QC-LDPC codes with girth 6. Interestingly, increasing the type of the constructed QC-LDPC codes usually cause to increase upper-bounds on the minimum-distance and for some proper CPM-sizes, we can find QC-LDPC codes with larger minimum-distances. In fact, applying Algorithm 2 on the weight-matrix $W(1, L, T)$, $T = 3, 4$, $2 \leq L \leq 10$, some slope-vectors corresponding to 4-cycle free type-III and type-IV QC-LDPC codes with different rates are provided in Table 11, in which much effort has been made to bring CPM-size m of the constructed codes as close enough to the lower-bound LB reported in [5]. As Table 11 shows, the rate of the constructed codes tends to 1 when L enlarges, while the construction of type-I QC-LDPC codes with the same column-weights and lengths usually leads to lower rates.

Example 4.10: For a given weight-matrix, although Algorithm 2 can be used to generate some slope vectors randomly to achieve a given girth, there are some explicit methods to generate proper slope-matrices. The main drawback is that the CPM-size of the explicitly constructed codes is not small, although they may have good cycle multiplicities. For example, for positive integer L, let C_L be the type-III QC-LDPC code with weight-matrix $W = W(1, L, 3)$ and slope-matrix $S = (s_{1,j})_{1 \leq j \leq L}$, $s_{1,j} = (0, j, (L + 2)j)$. By Proposition 2.3, C_L has girth 6, for enough large CPM-size. Table 12 provides a (6,8)-cycle multiplicities comparison between C_L, $4 \leq L \leq 8$, of girth 6 and some 4-cycle free type-III QC-LDPC codes in [29] with the same weight matrices and lengths. As the table shows, the constructed codes have better (6,8)-cycle multiplicities than codes in [29].

V. SIMULATION RESULTS

In continue, we provide some bit-error-rate (BER) performance comparisons between the proposed multitype QC-LDPC codes with different girths, on one hand, and random-like QC-LDPC codes [3], progressive edge growth (PEG) LDPC codes [33] and some codes in [16], [19], [25], [31], [34]–[36], on the other hand. The simulated codes have been decoded by the sum-product algorithm [32] with

TABLE 12. (6,8)-cycle multiplicities of the constructed type-III QC-LDPC codes of length *n* against the codes in [29].

Cycle Length	6		8		Length
L	C_L	[29]	C_L	[29]	n
4	2077	2139	31248	30969	124
5	4085	4128	80625	80625	215
6	7182	8341	173052	180519	342
7	11680	14746	328719	364343	511
8	17927	19474	572208	593957	728

different maximum iterations after transmitting over binary phase-shift keying-modulated additive white Gaussian noise channel. Simulations show that the constructed multitype QC-LDPC codes outperform the mentioned codes, in some cases they are as well as the constructed codes.In the figures, constructed type-I and type-II QC-LDPC codes lifted from a weight matrix of size $J \times L$ with girth b and CPM-size m are denoted by $C_I(J \times L; gb)$ and $C_{II}(J \times L; gb)$, respectively. Moreover, a PEG LDPC code of size $J \times L$ with target girth b is denoted by $PEG(J \times L; tgb)$ or $PEG, Tg = b$ if the size is known.

Example 5.1: In [31] four codes with good performances are presented two of which have girth 12 and sizes 6×12 and 6×8 with weight-matrices W_1 and W_2 in (6), respectively, and the other two have girth 14 and sizes 9×12 and 9×15 with weight-matrices W_3 and W_4 in (7), respectively. Applying Algorithm 2 over weight matrices $W_1, \ldots, W_4$, the following slope-vectors $S_1, \ldots, S_4$ with CPM-sizes 600, 100, 300, and 1000, can be obtained to construct some codes, denoted C_1 and C_2 with girth 12 and C_3 and C_4 with girth 14, respectively.

S_1 = [285, 180, 372, 458, 505, 551, 187, 436, 179, 126, 491, 152, 264, 366, 532, 382, 595, 519, 304, 412, 33, 141, 27, 494],

S_2 = [96, 4, 44, 14, 35, 49, 29, 88, 84, 65, 77, 60, 0, 83, 83, 40],

S_3 = [62, 107, 278, 156, 138, 151, 96, 134, 255, 64, 273, 41, 35, 132, 190, 235, 122, 202, 44, 200, 224, 100, 268, 34],

S_4 = [659, 823, 598, 704, 738, 132, 407, 641, 616, 294, 529, 659, 453, 822, 206, 251, 268, 323, 176, 410, 750, 123, 453, 241, 875, 734, 249, 715, 184, 865]

Fig. 2, parts (a), (b), present BER/FER comparisons between $C_1, \ldots, C_4$, on one hand and type-II QC-LDPC codes in [31] and PEG LDPC codes with target girth 14 and 12, on the other hand, where weight-matrices and lengths are the same. As the figures show, for maximum iteration 20, the constructed type-II QC-LDPC codes perform better than codes in [31] and the corresponding PEG LDPC codes. Moreover, the constructed code C_2 has 5600 cycles of length 12, whereas the number of 12-cycles of the corresponding code in [31] is 6000 which is larger.

Example 5.2: Corresponding to the weight matrices $W_1 = W(2, 3, 2)$ and $W_2 = W(2, 5, 2)$, the following slope matrices S_1 and S_2 are constructed to generate two girth-6 type-II QC-LDPC codes with CPM-sizes 133 and 381, respectively.

$$S_1 = \begin{pmatrix} (0, 2) & (0, 91) & (0, 90) \\ (127, 42) & (35, 79) & (98, 30) \end{pmatrix},$$

$$S_2 = \begin{pmatrix} (0, 206) & (0, 273) & (0, 357) & (0, 104) & (0, 377) \\ (251, 355) & (129, 147) & 79, 257) & (172, 62) & (22, 177) \end{pmatrix}$$

Two comparisons between the constructed codes and corresponding type-II QC-LDPC codes in [25], PEG LDPC codes and some random-like LDPC codes are provided in Fig. 2, parts (c), (d), with maximum iteration 50. All of the constructed codes have girth 6 and regular column-weight 4. As the figures show, the constructed type-II QC-LDPC codes perform better than the codes in [25], PEG and random LDPC codes with the same lengths, rates and girths.

Example 5.3: Let W be the following $(3, 4)$ weight-matrix with slope-matrix S corresponding to a type-II QC-LDPC code with girth 8 and CPM-size 46.

$$W = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 0 & 2 & 0 & 2 \end{pmatrix}$$

$$S = \begin{pmatrix} (0, 9) & () & (0) & (0) \\ (41) & (0) & (27, 42) & () \\ () & (34, 35) & () & (19, 24) \end{pmatrix}$$

The constructed code with girth 8 is compared in Fig. 2, part (e), with a girth-8 type-II QC-LDPC code in [16], PEG code with target girth 8 and a 4-cycle free random-like LDPC code, with maximum iteration 64. All of the constructed codes have column-weight 3 and length 184. As the figure shows, the constructed type-II QC-LDPC code performs better than the code in [16], PEG and random-like LDPC codes with the same lengths and rates.

Moreover, to have an insight about the error-rate behavior of codes in different iterations, Figure 3 has provided BER and FER comparisons between the constructed type-II QC-LDPC code with the code in [16] in $dB = 5.5$. As the figure shows, the constructed code outperforms the code in [16] for iterations greater than or equal to 20.

Example 5.4: Applying Algorithm 2 on the base matrices of four IEEE 802.11n/ac standard codes with rates 2/3, 3/4 and 5/6 presented in [34], some of the proper slopes are presented in Table 13 with CPM-size 27. Fig. 2, parts (f), (g) show that the constructed type-I QC-LDPC codes of various rates R perform BER/FER as well as the IEEE 802.11n/ac standard codes and PEG codes of the length 648 when the maximum iteration is 20.

Example 5.5: Applying Algorithm 2 on the weight-matrix $W = W(2, 16, 2)$, the following slope-vector S_1 is generated to have a girth-6 type-II QC-LDPC code with CPM-size 511.

S_1 =[143, 215, 416, 133, 232, 463, 56, 479, 501, 109, 508, 510, 185, 40, 304, 390, 60, 492, 220, 183, 224, 44, 244, 407, 297, 54, 425, 396, 198, 480, 207, 278, 325, 397, 108, 414, 197, 476, 496, 412, 106, 443, 180, 359, 462, 105, 49, 131]

Then, the constructed code from S_1 is compared with a CCSDS standard code in [35] for different maximum iterations 10 and 50, whereas the weight matrix, girth and length are the same. Fig. 2, Part (h) shows that the constructed codes perform BER/FER as well as the code in [35] and PEG code with the same length and girth.

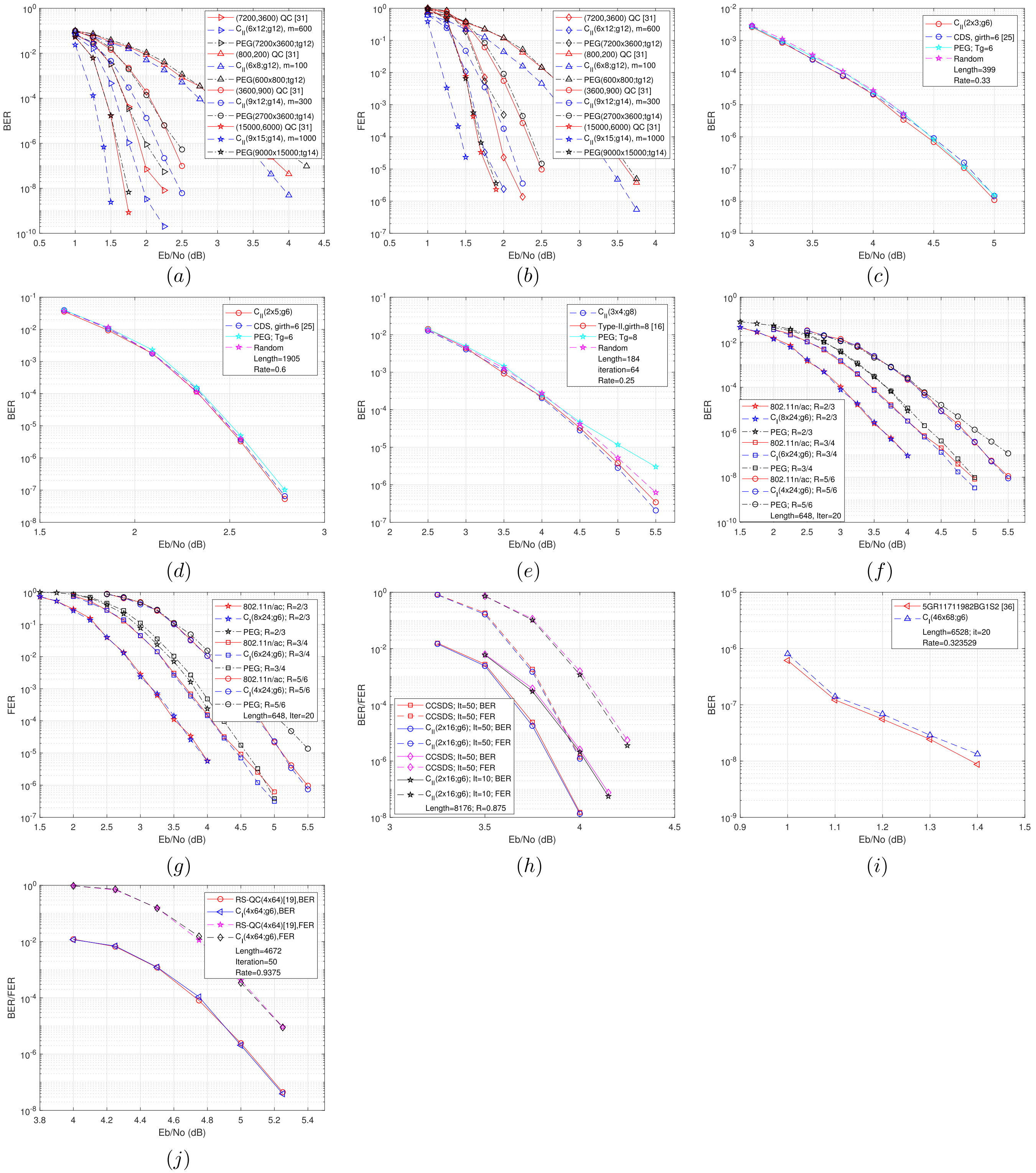

FIGURE 2. **Some BER/FER comparisons between the codes in (a),(b) Example 5.1 (c),(d) Example 5.2 (e) Example 5.3, (f),(g) Example 5.4, (h) Example 5.5 (i) Example 5.6, (j) Example 5.7.**

Example 5.6: The 3GPP agreed to consider two rate-compatible weight-matrices of sizes 46×68 and 42×52, denoted by $BG1$ and $BG2$, respectively [36]. Each weight-matrix supports all CPM-sizes $a \times 2^j$, for $a \in \{2, 3, 5, 7, 9, 11, 13, 15\}$ and $0 \leq j \leq 7$. We consider weight-matrix $BG1$ for $a = 3$ and $j = 5$ (CPM-size

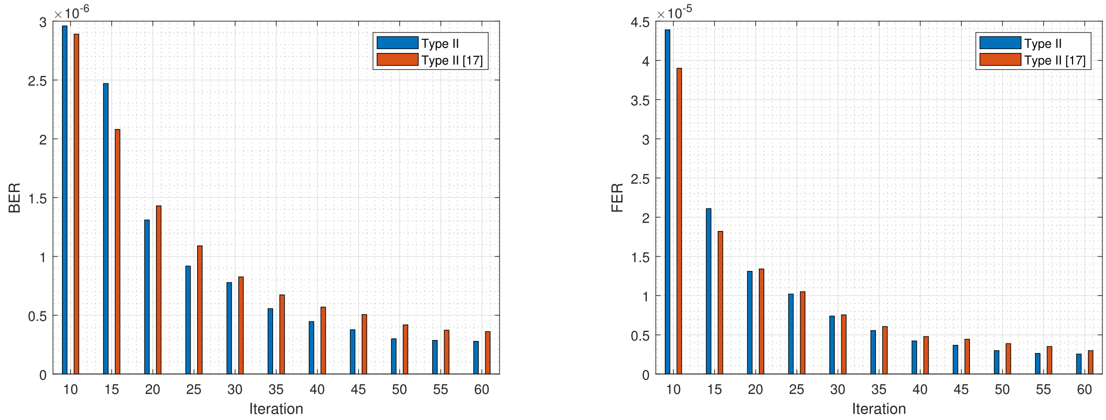

FIGURE 3. A BER/FER comparison of the codes in Example 5.3 for dB = 5.5 with different iterations.

TABLE 13. Some constructed codes with weight-matrices given by 802.11n/ac IEEE standard.

(J, L)	R	slope-vector
$(8, 24)$	$2/3$	[11,18,14,18,8,10,7,20,22,21,10,5,0,6,17,5,25,25,25,5,19,20,23,19,10,10,9,20,15,4,17,5,19,17,22,12,13,21,26,22,1,12,15,10,19,1,7,25,4,10,3,12,16,9,25,15,8, 2,0,22,9,3,11,15]
$(6, 24)$	$3/4$	[4,25,21,10,19,8,15,14,8,10,12,23,10,5,11,16,24,1,14,14,1,2,17,18,1,11,11,21,10,3,17,21,12,3,18,8,25,22,11,3,26,16,18,2,18,0,5,5,26,25,1,15,20,10,11,2,1,12, 7,23,15,6,11,1]
$(4, 24)$	$5/6$	[5,5,19,8,12,10,16,15,8,2,20,5,24,13,20,21,14,22,0,8,18,23,1,1,17,4,23,25,26,15,4,19,25,3,16,12,22,17,15,11,14,13,3,2,7,18,4,11,21,11,9,6,7,10,17,21,1,7,9,24, 6,18,22,18]

$3 \times 2^5 = 96$) to construct the PCM of a 5G QC-LDPC code, denoted by R11711982BG1S2, with rate 0.323529 and length 6528. Now, applying Algorithm 2 on $BG1$, the following slope-vector can be generated to construct a type-I QC-LDPC code with CPM-size 96.

$S = [30, 42, 90, 13, 26, 6, 73, 25, 30, 68, 31, 64, 91, 26, 20, 39, 34, 27, 88, 19, 44, 59, 89, 33, 55, 39, 66, 58, 90, 12, 28, 70, 36, 62, 90, 74, 35, 86, 30, 10, 56, 85, 11, 69, 29, 40, 3, 69, 45, 78, 80, 6, 5, 72, 9, 25, 51, 10, 10, 87, 4, 18, 71, 83, 83, 89, 21, 82, 16, 22, 89, 89, 34, 47, 54, 20, 84, 92, 81, 32, 65, 40, 10, 50, 88, 89, 62, 33, 89, 18, 4, 67, 52, 0, 75, 39, 76, 29, 64, 49, 50, 42, 31, 5, 77, 15, 93, 92, 3, 89, 5, 95, 47, 13, 28, 43, 35, 55, 4, 8, 14, 0, 7, 70, 31, 95, 67, 7, 92, 60, 83, 28, 59, 32, 19, 57, 87, 33, 71, 50, 6, 14, 33, 72, 37, 27, 76, 20, 34, 84, 33, 26, 38, 86, 63, 28, 29, 6, 57, 40, 16, 94, 94, 47, 56, 72, 74, 77, 1, 54, 89, 80, 95, 21, 15, 48, 57, 27, 93, 90, 14, 61, 42, 41, 43, 57, 84, 33, 11, 58, 25, 20, 49, 6, 22, 70, 82, 48, 28, 23, 35, 13, 56, 20, 41, 29, 55, 69, 42, 25, 13, 76, 60, 15, 13, 49, 37, 76, 49, 9, 57, 34, 54, 33, 19, 54, 84, 91, 15, 22, 9, 23, 56, 32, 8, 47, 93, 77, 22, 38, 34, 45, 48, 55, 22, 70, 2, 60]$

Fig. 2, Part (i) shows that the constructed code, denoted by $C_I(46 \times 68; g6)$, has the BER performance close to R11711982BG1S2 for maximum iteration 20. Moreover, if $n_l(C)$ is the number of cycles of length l in the Tanner graph of code C and $n_{6,8} = n_6 + n_8$, then we have $n_{6,8}(C_I(46 \times 68; g6)) = 5, 523, 696$ and $n_{6,8}(R11711982BG1S2) = 5, 577, 648$. Then, the constructed code has fewer total (6, 8) cycle multiplicities than the code in [36]

Example 5.7: In [19], some 4-cycle free algebraic type-I QC-LDPC codes are constructed based on Reed-Solomon (RS) codes, denoted by RS-QC-LDPC codes. For weight-matrix $W = (4, 64, 1)$ and CPM-size 73, Algorithm 2 is applied to generate the following slope vector corresponding to a type-I QC-LDPC code with girth 6.

$S = [34, 26, 3, 13, 53, 67, 30, 56, 20, 27, 17, 51, 57, 57, 11, 65, 72, 44, 11, 49, 58, 19, 51, 40, 28, 47, 5, 48, 32, 35, 67, 19, 49, 33, 63, 28, 46, 14, 57, 3, 46, 22, 17, 39, 39, 58, 30, 38, 9, 69, 19, 15, 27, 31, 41, 38, 24, 18, 0, 53, 53, 3, 68, 4, 70, 16, 14, 34, 18, 32, 9, 33, 29, 58, 36, 50, 21, 6, 0, 24, 64, 36, 41, 29, 35, 36, 69, 22, 28, 30, 20, 29, 34, 69, 43, 42, 72, 12, 48, 25, 5, 23, 8, 45, 66, 60, 64, 13, 64, 1, 21, 68, 59, 65, 51, 20, 37, 59, 2, 27, 70, 68, 10, 42, 44, 0, 63, 48, 55, 2, 10, 7, 54, 15, 9, 12, 11, 12, 6, 37, 63, 49, 4, 52, 44, 25, 15, 10, 66, 4, 62, 6, 61, 39, 33, 14, 38, 16, 26, 5, 8, 43, 16, 31, 59, 40, 13, 60, 37, 23, 62, 21, 60, 72, 66, 7, 2, 45, 40, 8, 1, 62, 41, 26, 61, 32, 56, 52, 45, 7, 55, 46]$.

Fig. 2, part (j) shows that the constructed code C_1, denoted by $C_I(4 \times 64; g6)$, has BER/FER performance as well as the counterpart code C_2 in [19], denoted by RS-QC(4×64), for maximum iteration 50. Moreover, we have $n_{6,8}(C_1) = 11, 284, 778$ and $n_{6,8}(C_2) = 11, 339, 820$. Then, the constructed code has fewer total (6, 8) cycle multiplicities than the code in [19].

VI. CONCLUSION

The class of protograph LDPC codes lifted cyclically from the protographs with multiple edges are referred to as multitype QC-LDPC codes presented by two weight and slope-matrices. In this paper, the maximum-achievable girth of multitype QC-LDPC codes with a given weight-matrix is determined efficiently by some inevitable chains. Using this approach, some weight matrices with a given maximum-achievable girth can be found by a simple random search. To continue, for a given weight-matrix, some slope-matrices are found by a depth-first search algorithm such that the corresponding multitype QC-LDPC codes have some advantages over the existing codes in terms of CPM-sizes, minimum-distance upper-bounds, cycle distributions, and girths. Simulation results show that the constructed codes have a good BER/FER performance in comparison with the state-of-the-art achievements.

ACKNOWLEDGMENT

The authors would like to appreciate the use of the computational clusters of the High Performance Computing Center (Shahr-e-Kord University, Iran), to complete this work.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
[2] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.
[3] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
[4] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.
[5] M.-R. Sadeghi and F. Amirzade, "Analytical lower bound on the lifting degree of multiple-edge QC-LDPC codes with girth 6," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1528–1531, Aug. 2018.
[6] A. Tasdighi and M. R. Sadeghi, "On advisability of designing short length QC-LDPC codes using perfect difference families," 2017, *arXiv:1701.06218*. [Online]. Available: http://arxiv.org/abs/1701.06218
[7] A. Tasdighi, A. H. Banihashemi, and M. R. Sadeghi, "Symmetrical constructions for regular girth-8 QC-LDPC codes," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 14–22, Oct. 2016.
[8] A. Tasdighi, A. H. Banihashemi, and M. R. Sadeghi, "Efficient search of girth-optimal QC-LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 1552–1564, Feb. 2016.
[9] M. Esmaeili and M. Gholami, "Structured quasi-cyclic LDPC codes with girth 18 and column-weight $J \geqslant 3$," *AEU Int. J. Electron. Commun.*, vol. 64, no. 3, pp. 202–217, Mar. 2010.
[10] S. Kim, J. S. No, H. Chung, and D. J. Shin, "Quasi-cyclic low-density parity-check codes with girth larger than 12," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2885–2891, Aug. 2007.
[11] M. E. O'Sullivan, "Algebraic construction of sparse matrices with large girth," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 718–727, Feb. 2006.
[12] G. Raeisi and M. Gholami, "Edge coloring of graphs with applications in coding theory," *China Commun.*, vol. 18, no. 1, pp. 181–195, Jan. 2021.
[13] L. Dolecek, P. Lee, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright, "Predicting error floors of structured LDPC codes: Deterministic bounds and estimates," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 908–917, Jun. 2009.
[14] M. P. C. Fossorier, "Quasi-cyclic low density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.
[15] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.
[16] R. Smarandache and P. O. Vontobel, "Quasi-cyclic LDPC codes: Influence of proto- and tanner-graph structure on minimum Hamming distance upper bounds," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 585–607, Feb. 2012.
[17] M. Gholami, M. Esmaeili, and M. Samadieh, "Quasi-cyclic low-density parity-check codes based on finite set systems," *IET Commun.*, vol. 8, no. 10, pp. 1837–1849, Jul. 2014.
[18] I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, and R. V. Satyukov, "Searching for voltage graph-based LDPC tailbiting codes with large girth," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2265–2279, Apr. 2012.
[19] X. Xiao, B. Vasic, S. Lin, K. Abdel-Ghaffar, and W. E. Ryan, "Reed-solomon based quasi-cyclic LDPC codes: Designs, girth, cycle structure, and reduction of short cycles," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5275–5286, Aug. 2019.
[20] J. Huang, L. Liu, W. Zhou, and S. Zhou, "Large-girth nonbinary QC-LDPC codes of various lengths," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3436–3447, Dec. 2010.
[21] H. Xu and B. Bai, "Superposition construction of q-ary LDPC codes by jointly optimizing girth and number of shortest cycles," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1285–1288, Jul. 2016.
[22] G. Zhang, Y. Hu, D. Ren, Y. Liu, and Y. Yang, "Type-II QC-LDPC codes from multiplicative subgroup of prime field," *IEEE Access*, vol. 8, pp. 142459–142467, 2020.
[23] K. Lally, "Explicit construction of type-II QC LDPC codes with girth at least 6," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 2371–2375.
[24] G. Zhang, "Type-II quasi-cyclic low-density parity-check codes from Sidon sequences," *Electron. Lett.*, vol. 52, no. 5, pp. 367–369, Mar. 2016.
[25] L. Zhang, B. Li, and L. Cheng, "Construction of type-II QC LDPC codes based on perfect cyclic difference set," *Chin. J. Electron.*, vol. 24, no. 1, pp. 146–151, Jan. 2015.
[26] L.-X. Li, H.-B. Li, J.-B. Li, and H. Jiang, "Construction of type-II QC-LDPC codes with fast encoding based on perfect cyclic difference sets," *Optoelectron. Lett.*, vol. 13, no. 5, pp. 358–362, Sep. 2017.
[27] G. Malema and M. Nkwebi, "Construction of flexible type II and III QC-LDPC codes," *Signal Process*, vol. 3, pp. 31–34, May 2014.
[28] F. Abedi and M. Gholami, "Some explicit constructions of type-II, III, IV, V QC-LDPC codes with girth 6," *China Commun.*, vol. 17, no. 5, pp. 89–109, May 2020.
[29] H. Park, S. Hong, J.-S. No, and D.-J. Shin, "Construction of high-rate regular quasi-cyclic LDPC codes based on cyclic difference families," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3108–3113, Aug. 2013.
[30] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Prog. Rep.*, vol. 42, no. 154, pp. 142–154, 2003.
[31] 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.
[32] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[33] X. Jiang, M. H. Lee, and J. Qi, "Improved progressive edge-growth algorithm for fast encodable LDPC codes," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 178, Dec. 2012.
[34] *IEEE 802.11n Standard. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, IEEE Standard 802.11n-2009, Oct. 2009, pp. 1–502.
[35] The Consultative Committee for Space Data Systems, "Low density parity check codes for use in near-Earth and deep space applications," Orange Book, Washington, DC, USA, Sep. 2007, pp. 1–44, no. 2. [Online]. Available: https://public.ccsds.org/Pubs/131x1o2e2s.pdf
[36] *Chairman's Notes of Agenda Item 7.1.4*, document 3GPP TSG RAN WG1 Meeting AH 2, R1-1711982, Channel Coding, Ad-Hoc chair (Nokia), 2017. Accessed: May 22, 2019. [Online]. Available: https://portal.3gpp.org/ngppapp/CreateTdoc.aspx?mode=view&contributionId=805088
[37] J. Cannon, W. Bosma, C. Fieker, and A. Steel, "Handbook of magma functions version 2.19," Sydney, NSW, Australia, Apr. 2013. [Online]. Available: http://magma.maths.usyd.edu.au/magma/

FARZANEH ABEDI received the B.S. and M.S. degrees in applied mathematics from the University of Isfahan, Isfahan, Iran, in 2012 and 2014, respectively. She is currently pursuing the Ph.D. degree in mathematics (coding theory) with Shahrekord University, Shahrekord, Iran, under the supervisor of Dr. Mohammad Gholami. Her research interests include algebraic coding theory, LDPC codes, and algebraic decoding algorithms.

MOHAMMAD GHOLAMI graduated in mathematics (Hons.) from the Sharif University of Technology, Tehran, Iran, in 2003. He received the Ph.D. degree in mathematics (information and coding theory) from the Isfahan University of Technology, Isfahan, Iran, in 2009. Since September 1998, he has been with the Department of Mathematical Sciences, Shahrekord University, Shahrekord, Iran, where he was received the grade of an Associate Professor from October 2015.