www.combinatorics.ir

Transactions on Combinatorics
ISSN (print): 2251-8657, ISSN (on-line): 2251-8665
Vol. 5 No. 2 (2016), pp. 11-22.

www.ui.ac.ir

RECURSIVE CONSTRUCTION OF (J, L) QC LDPC CODES WITH GIRTH 6

MOHAMMAD GHOLAMI* AND ZAHRA RAHIMI

Communicated by Dianhua Wu

Abstract. In this paper, a recursive algorithm is presented to generate some exponent matrices which correspond to Tanner graphs with girth at least 6. For a $J \times L$ exponent matrix E, the lower bound $Q(E)$ is obtained explicitly such that (J, L) QC LDPC codes with girth at least 6 exist for any circulant permutation matrix (CPM) size $m \geq Q(E)$. The results show that the exponent matrices constructed with our recursive algorithm have smaller lower-bound than the ones proposed recently with girth 6.

1. Introduction

Low density parity-check (LDPC) codes are a class of linear block codes with sparse parity-check matrix (PCM), which were first proposed by Gallager [1] in 1960. These codes have excellent error correcting capacity, being used in many modern applications. In 1981, Tanner [2] introduced a graphical representation of each LDPC code with PCM H, now called Tanner graph TG(H). A (J, L) regular LDPC code is defined by a parity-check matrix H with constant column-weight J and constant row-weight L. Quasi cyclic (QC) LDPC codes are a class of LDPC codes whose parity-check matrices consist of circulant permutation matrices (CPMs) or zero matrices of the same size, namely, $m \times m$. LDPC codes are partitioned into two main classes from the manufacturing method perspective: random LDPC codes and structured LDPC codes. Although, long random-like LDPC codes in general perform closer to the Shannon limit [3] than their equivalent structured LDPC codes; however, for practical lengths, well designed structured LDPC codes show better error correcting performance than the random-like ones.

An important parameter affecting the performance of a given LDPC code is the girth. The girth of an LDPC code with PCM H, denoted by $g(H)$, is defined as the length of the shortest cycle in

MSC(2010): Primary: 11H71; Secondary: 11T71, 68P30.

Keywords: QC LDPC codes, Tanner graph, exponent matrix.

Received: 27 May 2014, Accepted: 26 February 2015.

*Corresponding author.

TG(H). The presence of short cycles in the Tanner graph of an LDPC code can decrease the decoder's performance [3]. Thus, a given LDPC code without short cycles, especially 4-cycles, can perform well with iterative decoding. If the parity-check matrix H of an LDPC code has this constraint on the rows and columns that no two rows (or two columns) have more than one common position where they have nonzero components, then H is referred to as the row-column (RC)-constraint. When PCM H satisfies in RC-constraint, then it's Tanner graph is free of 4-cycles, so the girth is at least 6. If the RC-constraint holds for a PCM of a (J, L) QC LDPC code then the code has minimum distance at least $J+1$. Some known methods to construct LDPC codes satisfying in RC-constraint are as follows.

In [7] the authors have introduced some QC-LDPC codes with no 4-cycles based on a special type of combinatoric designs, known as balance incomplete block designs. Lan et.al. [8] has introduced some QC-LDPC codes on finite fields and showed that the constructed codes perform very well on AWGN channel. In [9] a method has been proposed for constructing QC-LDPC codes based on CPMs via a simple quadratic congruential equation. Type-II QC LDPC codes are a class of QC LDPC codes constructed from combinations of weight-0, weight-1 and weight-2 circulant matrices. One explicit construction of Type-II QC-LDPC codes with girth at least six has been provided in [6]. In [4], one explicit construction of LDPC codes with girth at least 6 based on some known bipartite graphs have been introduced. O'Sullivan [5] has constructed some sparse matrices with an algebraic method whose connected bipartite graphs have large girth.

In this paper, a recursive algorithm for construction of some exponent matrices which correspond to Tanner graphs with girth at least six, say briefly exponent matrices with girth at least 6, is presented. Next, corresponding to each exponent matrix E with girth at least 6, the lower-bound $Q(E)$ is presented such that for any CPM size $m \geq Q(E)$, the associated codes with exponent matrix E have girth at least 6. Interestingly, the lower-bound of the constructed exponent matrices with girth at least 6 is better than the one of the best well known exponent matrices with girth at least 6 [10] (with the same row and column-weight distributions).

The outline of this paper is as follows. First, we give some preliminaries and requirements in section II, then, a recursive algorithm is proposed to generate some exponent matrices of (J, L) QC LDPC codes with girth at lest 6. In section III, corresponding to each exponent matrix E with girth at least 6, the lower-bound $Q(E)$ is presented in which states that QC LDPC codes with exponent matrix E have girth at least 6 for any CPM size $m \geq Q(E)$. Finally, some tables of the lower-bound of constructed exponent matrices in comparison with the other works are provided in the last section.

2. **Preliminaries and Recursive construction**

Let m, s be two integers such that $0 \leq s < m$. Denote the $m \times m$ identity matrix by I_m. The CPM I_m^s is obtained from I_m when each row is shifted by s positions to the right, i.e. $I_m^s = \begin{pmatrix} 0 & I_s \\ I_{m-s} & 0 \end{pmatrix}$. For simplicity, if m is known, I_m^s is shown by I^s.

Let J and L be two integers with $3 \leq J < L$ and E be a $J \times L$ matrix composed of some non-negative integers, as shown below.

$$E = \begin{pmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,L} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ e_{J,1} & e_{J,2} & \cdots & e_{J,L} \end{pmatrix}$$

By replacing each entry $e_{i,j}$ by $m \times m$ matrix $I^{e_{i,j}} = I^{e_{i,j} \pmod m}$, the parity-check matrix $H_{m,E}$ of a QC LDPC code with exponent matrix E and CPM size m can be obtained as follows.

$$H_{m,E} = \begin{pmatrix} I^{e_{1,1}} & I^{e_{1,2}} & \dots & I^{e_{1,L}} \\ I^{e_{2,1}} & I^{e_{2,2}} & \dots & I^{e_{2,L}} \\ \vdots & \vdots & \ddots & \vdots \\ I^{e_{J,1}} & I^{e_{J,2}} & \dots & I^{e_{J,L}} \end{pmatrix}.$$

Example 2.1. Let $\mathbf{a} = (a_0, a_1, \dots, a_{J-1})$ be a length-J sequence of some non-negative integers and $E(\mathbf{a})$ be the following $J \times L$ matrix.

$$E(\mathbf{a}) = \begin{pmatrix} 0.a_0 & 1.a_0 & \cdots & (L-1).a_0 \\ 0.a_1 & 1.a_1 & \cdots & (L-1).a_1 \\ \vdots & \vdots & \ddots & \vdots \\ 0.a_{J-1} & 1.a_{J-1} & \cdots & (L-1).a_{J-1} \end{pmatrix}.$$

It is verified in [11] that if the elements a_i, $0 \le i \le J-1$, are distinct, then $E(\mathbf{a})$ can be considered as the exponent matrix of an LDPC code with girth at least 6, when CPM size m is large enough. Especially, for $\mathbf{a} = (0, 1, \dots, J-1)$, Karimi, et.al [10] used the following exponent matrix to generate some explicitly constructed girth-6 exponent matrices with lower-bound small as possible.

$$E = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & L-1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & J-1 & 2(J-1) & \cdots & (J-1)(L-1) \end{pmatrix}, \tag{2.1}$$

In the rest of this section, for given integers J, L, $1 \le J < L$, we propose an algorithm which finds exponent matrices $E = (e_{i,j})_{1 \le i \le J, 1 \le j \le L}$ by a recursive method, such that the corresponding Tanner graph is free of 4-cycles. In fact, the outline of the algorithm is as follows.

(1) For $v \le J$, let $\mathbf{n}_v = (n_1, n_2, \dots, n_v)$ be a sequence of some positive integers such that $\sum_{i=1}^{v} n_i = L$. Partition the exponent matrix E into v blocks as $E = \begin{pmatrix} E_1 & E_2 & \dots & E_v \end{pmatrix}$, such that the k-th block E_k, $1 \le k \le v$, is the following matrix with n_k columns.

$$E_k = \begin{pmatrix} e_{1,c_{k-1}+1} & e_{1,c_{k-1}+2} & \dots & e_{1,c_k} \\ e_{2,c_{k-1}+1} & e_{2,c_{k-1}+2} & \dots & e_{2,c_k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{J,c_{k-1}+1} & e_{J,c_{k-1}+2} & \dots & e_{J,c_k} \end{pmatrix}$$

where $c_0 = 0$ and $c_k = \sum_{i=1}^{k} n_i$, for $0 < k \leq v$.

(2) Set the entries of the first column of E_1 and the entries of the k-th row of E_k, $1 \leq k \leq v$, to be zero.

(3) Apply the following recursive formula to construct the other entries of E_k, $1 \leq k \leq v$, below the zero rows proposed in Step 2.

$$e_{i,j} = (e_{i,j-1} - e_{i-1,j-1}) + 1 + e_{i-1,j}.$$

(4) Set e_{1,n_1+1} to be the least integer such that no 4-cycle exist in the Tanner graph corresponds to E, i.e.

$$e_{1,n_1+1} = \min(\mathbb{Z}^{\geq 0} - \{e_{i,n_1+1} - e_{i,j} \ : \ 3 \leq i \leq J, \ 1 \leq j \leq n_1\}).$$

(5) The other entries of the first row of E, i.e. exception first $n_1 + 1$ entries which are equal to zero, are constructed by the following recursive formula.

$$e_{1,j} = e_{J,j} + 1 + \max\{e_{1,j-1} - e_{J,j-1}, 0\},$$

where $n_1 + 1 < j \leq L$.

(6) The other entries above the zero rows (in Step 2), i.e. exception the elements of the first row of E, are generated by the following recursive formula.

$$e_{i,j} = e_{i-1,j} + 1 + \max\{e_{i,j'} - e_{i-1,j'} \ : \ 1 \leq j' \leq j-1\}.$$

The algorithm is now given briefly as follows.

Algorithm 1. *Input:* J,L and $\mathbf{n} = (n_1, \ldots, n_v)$.
Output: $J \times L$ exponent matrix E with girth at least 6.

```
for i from 1 to J do {
 e_{i,1} = 0
}
for i from 1 to v do {
  for j from sum_{t=1}^{i-1} n_t+1 to sum_{t=1}^{i} n_t do {
   e_{i,j} = 0
  }
}
for k from 1 to v do {
  for j from sum_{t=1}^{k-1} n_t+1 to sum_{t=1}^{k} n_t do {
   for i from k + 1 to J do {
    if k ≠ j then
      e_{i,j} = (e_{i,j-1} - e_{i-1,j-1}) + 1 + e_{i-1,j}
    }
   }
  }
```

}
$e_{1,n_1+1} = \min(\mathbb{Z}^{\geq 0} - \{e_{i,n_1+1} - e_{i,j} \ : \ 3 \leq i \leq J, \ 1 \leq j \leq n_1\})$
for k from 2 to v do {
  for j from $n_1 + 2$ to $\sum_{t=1}^{k+1} n_t$ do {
    $e_{1,j} = \max(e_{1,j-1} - e_{J,j-1}, 0) + 1 + e_{J,j}$
  }
}
for k from 2 to v do {
  for j from $\sum_{t=1}^{k-1} n_t$+1 to $\sum_{t=1}^{k} n_t$ do {
    for i from 2 to $k-1$ do {
      $e_{i,j} = \max\{e_{i,j'} - e_{i-1,j'} \ : \ 1 \leq j' \leq j-1\} + 1 + e_{i-1,j}$
    }
  }
}

Using Algorithm 1, all entries of exponent matrix E are generated recursively. In continue, we will prove that for CPM size m enough large, we have $g(H_{m,E}) \geq 6$. In fact, corresponding to each algorithm output E, the lower-bound $Q(E)$ is defined as the minimum integer q satisfying $g(H_{m,E}) \geq 6$, for each $m \geq q$. In the next section, we give an explicit method for evaluating $Q(E)$.

Example 2.2. Let $J = 3$, $L = 6$ and $\mathbf{n}_v = (2, 2, 2)$. Algorithm 1 can be used to generate the following 3×6 exponent matrix E with girth 6.

$$E = \begin{pmatrix} 0 & 0 & 1 & 4 & 2 & 3 \\ 0 & 1 & 0 & 0 & 4 & 6 \\ 0 & 2 & 2 & 3 & 0 & 0 \end{pmatrix} \tag{2.2}$$

In fact, the details of the algorithm used in this example are as follows.
First, partition exponent matrix E into three blocks E_1, E_2 and E_3 in which each block E_i is a 3×2 array.

$$E_1 = \begin{pmatrix} e_{1,1} & e_{1,2} \\ e_{2,1} & e_{2,2} \\ e_{3,1} & e_{3,2} \end{pmatrix}, \quad E_2 = \begin{pmatrix} e_{1,3} & e_{1,4} \\ e_{2,3} & e_{2,4} \\ e_{3,3} & e_{3,4} \end{pmatrix}, \quad E_2 = \begin{pmatrix} e_{1,5} & e_{1,6} \\ e_{2,5} & e_{2,6} \\ e_{3,5} & e_{3,6} \end{pmatrix}.$$

Next, set the first column of E and i-th row of E_i, $1 \leq i \leq 3$, to be zero, i.e.

$$e_{1,1} = e_{2,1} = e_{3,1} = e_{1,2} = e_{2,3} = e_{2,4} = e_{3,5} = e_{3,6} = 0.$$

Using Step 3 of the algorithm, the entries below the zero-rows of blocks E_i, $1 \leq i \leq 3$, can be constructed as follows.

$$e_{2,2} = (e_{2,1} - e_{1,1}) + 1 + e_{1,2} = 1, \quad e_{3,2} = (e_{3,1} - e_{2,1}) + e_{2,2} + 1 = 2,$$
$$e_{3,3} = (e_{3,2} - e_{2,2}) + 1 + e_{2,3} = 2, \quad e_{3,4} = (e_{3,3} - e_{2,3}) + 1 + e_{2,4} = 3.$$

Now, by Step 4 of the algorithm, the entry $e_{1,3}$ is constructed as follows.

$$e_{1,3} = \min(\mathbb{Z}^{\geq 0} - \{e_{i,3} - e_{i,j} \,:\, 3 \leq i \leq J,\ 1 \leq j \leq 2\}) = \min(\mathbb{Z}^{\geq 0} - \{0, 2\}) = 1.$$

because, $e_{3,3} - e_{3,1} = 2$ and $e_{3,3} - e_{3,2} = 0$.
Using Step 5 of the algorithm, the remaining entries of the first row of E can be obtained as follows.

$$\begin{aligned} e_{1,4} &= \max(e_{1,3} - e_{3,3}, 0) + 1 + e_{3,4} = \max(1 - 2, 0) + 1 + 3 = 4, \\ e_{1,5} &= \max(e_{1,4} - e_{3,4}, 0) + 1 + e_{3,5} = \max(4 - 3, 0) + 1 + 0 = 2, \\ e_{1,6} &= \max(e_{1,5} - e_{3,5}, 0) + 1 + e_{3,6} = \max(2 - 0, 0) + 1 + 0 = 3. \end{aligned}$$

Finally, the other entries of E can be generated by the last step of the algorithm, as follows.

$$\begin{aligned} e_{2,5} &= \max\{e_{2,j'} - e_{1,j'} \,:\, 1 \leq j' \leq 4\} + 1 + e_{1,5} = 1 + 1 + 2 = 4, \\ e_{2,6} &= \max\{e_{2,j'} - e_{1,j'} \,:\, 1 \leq j' \leq 5\} + 1 + e_{1,6} = 2 + 1 + 3 = 6. \end{aligned}$$

Example 2.3. For $J = 7$, $L = 18$ and $\mathbf{n}_v = (3, 4, 5, 6)$, Algorithm 1 can be used to generate the following 7×18 exponent matrix with girth 6.

$$\begin{pmatrix} 0 & 0 & 0 & 4 & 21 & 27 & 33 & 32 & 37 & 42 & 47 & 52 & 45 & 49 & 53 & 57 & 61 & 65 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 35 & 41 & 47 & 53 & 59 & 53 & 58 & 63 & 68 & 73 & 78 \\ 0 & 2 & 4 & 3 & 4 & 5 & 6 & 0 & 0 & 0 & 0 & 0 & 60 & 66 & 72 & 78 & 84 & 90 \\ 0 & 3 & 6 & 6 & 8 & 10 & 12 & 7 & 8 & 9 & 10 & 11 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 8 & 9 & 12 & 15 & 18 & 14 & 16 & 18 & 20 & 22 & 12 & 13 & 14 & 15 & 16 & 17 \\ 0 & 5 & 10 & 12 & 16 & 20 & 24 & 21 & 24 & 27 & 30 & 33 & 24 & 26 & 28 & 30 & 32 & 34 \\ 0 & 6 & 12 & 15 & 20 & 25 & 30 & 28 & 32 & 36 & 40 & 44 & 36 & 39 & 42 & 45 & 48 & 51 \end{pmatrix}$$

3. Minimum CPM Size of Girth 6 QC LDPC Codes

For some integers J, L, $1 < J < L$, let E be a $J \times L$ exponent matrix corresponds to Tanner graph of $H_{m,E}$ with girth 6, when m is large enough. Here, we want to find the lower-bound $Q(E)$ such that for each $m \geq Q(E)$, we have $g(H_{m,E}) \geq 6$. First, define $Q(E)$ as follows.

$$Q(E) = 1 + \max\{(e_{i_1,j_1} - e_{i_2,j_1}) + (e_{i_2,j_2} - e_{i_1,j_2}) \,:\, 1 \leq i_1 \neq i_2 \leq J,\ 1 \leq j_1 \neq j_2 \leq L\}.$$

Now, the following lemma shows $Q(E)$ is the requested lower-bound. For simplicity, hereinafter, $Q(E)$ is represented by Q.

Lemma 3.1. If E corresponds to a Tanner graph with girth 6, when the CPM size m tends to infinity, then E for each $m \geq Q$ also is.

Proof. Suppose that there is a 4-cycle for a fixed m ($m \geq Q$), so there are some distinct indices $1 \leq i_1, i_2 \leq J$ and $1 \leq j_1, j_2 \leq L$ such that:

$$(e_{i_1,j_1} - e_{i_2,j_1}) + (e_{i_2,j_2} - e_{i_1,j_2}) \equiv 0 \pmod{m}. \tag{3.1}$$

However, E corresponds to a Tanner graph with girth 6 for $m = +\infty$, so the left hand side equation (3.1) (LHS) cannot be zero. On the other hand, by definition of Q, it is obvious that $LHS < Q$, so we have $0 < |LHS| < Q$, which is a contradiction, because $m \geq Q$, and this completes the proof. □

To find Q more simplify, we use the notation $D_{r,s}$, $1 \leq r, s \leq J$, to denote the difference set of the r-th and s-th rows of E, i.e.

$$D_{r,s} = \{e_{r,1} - e_{s,1}, e_{r,2} - e_{s,2}, \ldots, e_{r,L} - e_{s,L}\}.$$

It is clear that $D_{s,r} = -D_{r,s}$. Next, set $T_{r,s} = \max(D_{r,s}) + \max(D_{s,r})$, then it can be seen easily that

$$Q = \max\{T_{r,s}\}_{1 \leq r < s \leq J} + 1.$$

Corollary 3.2. Let E be the exponent matrix given by Equation (2.1). Then $g(H_{m,E}) \geq 6$ for any CPM size $m \geq (J-1)(L-1)+1$. Hereinafter, by Q', we mean the lower-bound $(J-1)(L-1)+1$.

Example 3.3. The lower bound Q for the exponent matrix (2.2) is as follows.

$$\begin{array}{lll} D_{1,2} = \{0,-1,1,4,-2,-3\}, & D_{2,1} = \{0,1,-1,-4,2,3\} & \rightarrow T_{1,2} = 4+3 = 7, \\ D_{1,3} = \{0,-2,-1,1,2,3\}, & D_{3,1} = \{0,2,1,-1,-2,-3\} & \rightarrow T_{1,3} = 3+2 = 5, \\ D_{2,3} = \{0,-1,-2,-3,4,6\}, & D_{3,2} = \{0,1,2,3,-4,-6\} & \rightarrow T_{2,3} = 6+3 = 9. \end{array}$$

So, we have $Q = \max\{T_{r,s}\}_{1 \leq r < s \leq 3} + 1 = 9 + 1 = 10$.

Theorem 3.4. The exponent matrix E generated by Algorithm 1 corresponds to Tanner graph with girth 6 for any $m \geq Q$, where $Q = e_{v-1,L} + e_{v,L-n_v} + 1$ if $v > 2$ or ($v = 2$ and $n_v = n_2 > 1$) and $Q = e_{J,L} + 1$ if $v = 2$ and $n_v = n_2 = 1$.

Proof. First, let $m = +\infty$. By contradiction, let there is a 4-cycle in $\mathrm{TG}(H_{m,E})$, then there exist some $1 \leq i_1 < i_2 \leq J$ and $1 \leq j_1 < j_2 \leq L$ such that

$$(e_{i_2,j_1} - e_{i_1,j_1}) + (e_{i_1,j_2} - e_{i_2,j_2}) = 0 \tag{3.2}$$

Let $i_2 = i_1 + \alpha$. Consider the following cases.

(1) All four elements of equation (3.2) are under the zero rows (constructed in Step 1 of the algorithm). Thus, by step 2 of the algorithm, we have:

$$\begin{aligned} e_{i_2,j_2} - e_{i_1,j_2} &= (e_{i_2,j_2} - e_{i_2-1,j_2}) + (e_{i_2-1,j_2} - e_{i_2-2,j_2}) + \cdots + (e_{i_2-\alpha+1} - e_{i_2-\alpha,j_2}) \\ &> (e_{i_2,j_2-1} - e_{i_2-1,j_2-1}) + (e_{i_2-1,j_2-1} - e_{i_2-2,j_2-1}) + \cdots + (e_{i_2-\alpha+1,j_2-1}) \\ &> \cdots \\ &> (e_{i_2,j_1} - e_{i_2-1,j_1}) + (e_{i_2-1,j_1} - e_{i_2-2,j_1}) + \cdots + (e_{i_1+1,j_1} - e_{i_1,j_1}) \\ &= e_{i_2,j_1} - e_{i_1,j_1}, \end{aligned}$$

which is impossible, thus, such 4-cycle can not exist.

(2) All four elements of equation (3.2) are above the zero rows, then, according to Step 5 of the algorithm, we have:

$$\begin{aligned} e_{i_2,j_2} - e_{i_1,j_2} &= (e_{i_2,j_2} - e_{i_2-1,j_2}) + (e_{i_2-1,j_2} - e_{i_2-2,j_2}) + \cdots + (e_{i_2-\alpha+1,j_2} - e_{i_2-\alpha,j_2}) \\ &> \max\{e_{i_2,j'} - e_{i_2-1,j'}, 0\}_{1 \leq j' \leq j_2-1} + \max\{e_{i_2-1,j'} - e_{i_2-2,j'}, 0\}_{1 \leq j' \leq j_2-1} \\ &+ \cdots + \max\{e_{i_1+1,j'} - e_{i_1,j'}, 0\}_{1 \leq j' \leq j_2-1} \\ &\geq (e_{i_2,j_1} - e_{i_2-1,j_1}) + (e_{i_2-1,j_1} - e_{i_2-2,j_1}) + \cdots + (e_{i_1+1,j_1} - e_{i_1,j_1}) \\ &= e_{i_2,j_1} - e_{i_1,j_1}. \end{aligned}$$

which is impossible.

(3) Some of the elements of equation (3.2) are under the zero rows and some of them are above the zero rows. Consider the following cases.

(a) Let e_{i_1,j_1} and e_{i_2,j_1} are under the zero rows, and e_{i_1,j_2} and e_{i_2,j_2} are above the zero rows, then similar to the part (2), it can be seen easily that

$$e_{i_2,j_2} - e_{i_1,j_2} > e_{i_2,j_1} - e_{i_1,j_1},$$

which is impossible.

(b) Let $j_2 > n_1 + 1$, and only e_{i_1,j_2} be above the zero rows, thus we have:

$$\begin{aligned} e_{i_1,j_2} - e_{i_2,j_2} &= (e_{i_1,j_2} - e_{i_1-1,j_2}) + (e_{i_1-1,j_2} - e_{i_1-2,j_2}) + \cdots + (e_{2,j_2} - e_{1,j_2}) + (e_{1,j_2} - e_{J,j_2}) \\ &+ \cdots + (e_{i_2+1,j_2} - e_{i_2,j_2}) \\ &> \max\{e_{i_1,j'} - e_{i_1-1,j'}\}_{1\le j'\le j_2-1} + \max\{e_{i_1-1,j'} - e_{i_1-2,j'}\}_{1\le j'\le j_2-1} \\ &+ \cdots + \max\{e_{2,j'} - e_{1,j'}\}_{1\le j'\le j_2-1} + \max\{e_{1,j_2-1} - e_{J,j_2-1}, 0\} \\ &+ \cdots + (e_{i_2+1,j_2-1} - e_{i_2,j_2-1}) \\ &\ge (e_{i_1,j_1} - e_{i_1-1,j_1}) + (e_{i_1-1,j_1} - e_{i_1-2,j_1}) + \cdots + (e_{2,j_1} - e_{1,j_1}) + (e_{1,j_1} - e_{J,j_1}) \\ &+ \cdots + (e_{i_2+1,j_1} - e_{i_2,j_1}) \\ &= e_{i_1,j_1} - e_{i_2,j_1}. \end{aligned}$$

(c) Only, e_{i_2,j_1} is under the zero rows, then similar to part (2), it can be seen easily that

$$e_{i_2,j_2} - e_{i_1,j_2} > e_{i_2,j_1} - e_{i_1,j_1},$$

which is impossible.

(d) If e_{i_1,j_1}, e_{i_1,j_2} are above the zero rows, and e_{i_2,j_1}, e_{i_2,j_2} are under the zero rows, then

$$\begin{aligned} e_{i_1,j_2} - e_{i_2,j_2} &= (e_{i_1,j_2} - e_{i_1-1,j_2}) + (e_{i_1-1,j_2} - e_{i_1-1,j_2}) + \cdots + (e_{2,j_2} - e_{1,j_2}) + (e_{1,j_2} - e_{J,j_2}) \\ &+ (e_{J,j_2} - e_{J-1,j_2}) + \cdots + (e_{i_2+1,j_2} - e_{i_2,j_2}) \\ &> \max\{e_{i_1,j'} - e_{i_1-1,j'}\}_{1\le j'\le j_2-1} + \max\{e_{i_1-1,j'} - e_{i_1-2,j'}\}_{1\le j'\le j_2-1} + \cdots \\ &+ \max\{e_{2,j'} - e_{1,j'}\}_{1\le j'\le j_2-1} + (e_{1,j_2-1} - e_{J,j_2-1}) + (e_{J,j_2-1} - e_{J-1,j_2-1}) \\ &+ \cdots + (e_{i_2+1,j_2-1} - e_{i_2,j_2-1}) \end{aligned}$$

which implies:

$$\begin{aligned} RHS &> (e_{i_1,j_1} - e_{i_1-1,j_1}) + (e_{i_1-1,j_1} - e_{i_1-2,j_1}) + \cdots + (e_{2,j_1} - e_{1,j_1}) + (e_{1,j_1} - e_{J,j_1}) \\ &+ (e_{J,j_1} - e_{J-1,j_1}) + \cdots + (e_{i_2+1,j_1} - e_{i_2,j_1}) = e_{i_1,j_1} - e_{i_2,j_1} \end{aligned}$$

and this is contradiction.

(e) If $i_1 = 1$, $i_2 \ge 2$, $j_1 < n_1+1$ and $j_2 = n_1+1$, then based on Step 3 of the algorithm, e_{1,n_1+1} is constructed in such a way that no 4-cycles can't exist with the previous entries, i.e. $e_{i,n_1+1} - e_{1,n_1+1} \neq e_{i,j} - e_{1,j}$. Since $e_{1,j} = 0$ for $j < n_1+1$, we have $e_{1,n_1+1} \neq e_{i,n_1+1} - e_{i,j}$. Therefore, there is no 4-cycle between e_{1,n_1+1} and the other entries.

(f) If $i_1 = 1$, $j_1 = n_1 + 1$, $i_2 \ge 2$ and $j_2 > n_1 + 1$, then, without loss of generality, let $j_2 = n_1 + k$ and $i_2 = i$. Now, we need to show that $e_{1,n_1+1} - e_{i,n_1+1} \neq e_{1,n_1+k} - e_{i,n_1+k}$. Two following cases can be considered.

(i) The entry e_{i,n_1+k} is under the zero rows. Then, we have

$$\begin{aligned} e_{i,n_1+k} - e_{1,n_1+k} &= (e_{i,n_1+k} - e_{i-1,n_1+k}) + (e_{i-1,n_1+k} - e_{i-2,n_1+k}) + \cdots \\ &+ (e_{2,n_1+k} - e_{1,n_1+k}) \end{aligned}$$

However, according to Step (5) of the algorithm, we have:

$$\begin{aligned} RHS &> (e_{i,n_1+1} - e_{i-1,n_1+1}) + (e_{i-1,n_1+1} - e_{i-2,n_1+1}) + \cdots + (e_{2,n_1+1} - e_{1,n_1+1}) \\ &= e_{i,n_1+1} - e_{1,n_1+1} \end{aligned}$$

which is impossible.

(ii) The entry e_{i,n_1+k} is above the zero rows. Then, we have

$$\begin{aligned} e_{1,n_1+k} - e_{i,n_1+k} &= (e_{1,n_1+k} - e_{J,n_1+k}) + (e_{J,n_1+k} - e_{J-1,n_1+k}) + \cdots \\ &+ (e_{i+1,n_1+k} - e_{i,n_1+k}). \end{aligned}$$

However, based on Step (2) of the algorithm,

$$\begin{aligned} RHS &> (e_{1,n_1+1} - e_{J,n_1+1}) + (e_{J,n_1+1} - e_{J-1,n_1+1}) + \cdots + (e_{i+1,n_1+1} - e_{i,n_1+1}) \\ &= e_{1,n_1+1} - e_{i,n_1+1} \end{aligned}$$

which is impossible.

(g) If at least one of the elements of Equation (3.2) is zero, then the following cases can be considered.

(i) $e_{i_1,j_1} = e_{i_1,j_2} = 0$. In this case, all four elements of Equation (3.2) are in a block, such as block E_k for some $1 \leq k \leq v$, and e_{i_2,j_1} and e_{i_2,j_2} are under zero rows. Thus, according to part (1), we have:

$$e_{i_2,j_2} - e_{i_1,j_2} > e_{i_2,j_1} - e_{i_1,j_1},$$

which is impossible.

(ii) $e_{i_2,j_1} = e_{i_2,j_2} = 0$. So, all four elements of Equation (3.2), are in a block, such as block E_k for some $1 \leq k \leq v$, and e_{i_1,j_1} and e_{i_1,j_2} are above the zero rows. In this case, similar to part (b) in (3), we have:

$$e_{i_1,j_2} - e_{i_2,j_2} > e_{i_1,j_1} - e_{i_2,j_1},$$

which is impossible.

(iii) $e_{i_1,j_1} = e_{i_2,j_2} = 0$. In this case, we have:

$$e_{i_2,j_1} - e_{i_1,j_1} = e_{i_2,j_1} - 0 > 0, \quad e_{i_2,j_2} - e_{i_1,j_2} = 0 - e_{i_1,j_2} < 0$$

Thus, Equation (3.2) is not established.

(iv) $j_1 = 1$. Then, $e_{i_1,j_1} = e_{i_2,j_1} = 0$. It is sufficient to show that $e_{i_2,j_2} - e_{i_1,j_2} \neq 0$. The following cases can be considered.

(A) $e_{i_2,j_2}, e_{i_1,j_2} \neq 0$. Similar to part (2), it can be seen that

$$e_{i_2,j_2} - e_{i_1,j_2} > e_{i_2,j_1} - e_{i_1,j_1}.$$

(B) $e_{i_1,j_2} = 0$. Then, $e_{i_2,j_2} - e_{i_1,j_2} = e_{i_2,j_2} - 0 > 0$.

(C) $e_{i_2,j_2}=0$. Then $e_{i_2,j_2}-e_{i_1,j_2}=0-e_{i_1,j_2}<0$.

(v) Only e_{i_1,j_1} is zero. Certainly, e_{i_2,j_1} is under the zero rows and e_{i_1,j_2} is above the zero rows. Now, two following cases can be considered.

(A) e_{i_2,j_2} is under zero rows. Then, similar to part (b) of (3), it can be seen that

$$e_{i_1,j_2}-e_{i_2,j_2}>e_{i_1,j_1}-e_{i_2,j_1}.$$

(B) e_{i_2,j_2} is above the zero rows. Similar to part (2), we have

$$e_{i_2,j_2}-e_{i_1,j_2}>e_{i_2,j_1}-e_{i_1,j_1}.$$

(vi) Only e_{i_2,j_1} is zero. Therefore, the other elements of Equation (3.2) are above the zero rows. Like part (2), it can be seen that

$$e_{i_2,j_2}-e_{i_1,j_2}>e_{i_2,j_1}-e_{i_1,j_1}.$$

(vii) Only e_{i_1,j_2} is zero. Thus, the other elements of Equation (3.2) are under the zero rows. Similar to part (1), we have

$$e_{i_2,j_2}-e_{i_1,j_2}>e_{i_2,j_1}-e_{i_1,j_1}.$$

(viii) Only e_{i_2,j_2} is zero. Then, e_{i_1,j_1} and e_{i_1,j_2} are above the zero rows and e_{i_2,j_1} is under the zero rows. According to part (b) of (3), it can be seen that

$$e_{i_1,j_2}-e_{i_2,j_2}>e_{i_1,j_1}-e_{i_2,j_1}.$$

Hence, the existence of 4-cycle in each case is impossible and the first part of the proof is now completed. On the other hand, for the second part, Lemma 3.1 can be used to find the lower-bound Q easily as $Q=e_{v-1,L}+e_{v,L-n_v}+1$, if $v>2$ or ($v=2$ and $n_v>1$) and $Q=e_{J,L}+1$, if $v=2$ and $n_v=1$. Now, the proof is completed. □

4. **Numerical Results**

Tables 1-3 have provided a comparison between the lower-bound of the constructed exponent matrices with girth at least 6 and the ones recently proposed by Karimi [10], for different column-weights 3, 4 and 5, respectively. In these tables, Q and Q' are used to represent the lower bounds of our structure and the structure given in [10], respectively. Moreover, e_{1,n_1+1} is evaluated by Step (3) of the algorithm, where the vector $\mathbf{n}_v=(n_1,\ldots,n_v)$ is chosen randomly as the input of the algorithm. According to outputs of these tables, for different $\mathbf{n}_v$, the lower-bound Q is smaller than Q'. Especially, it seems that if $\mathbf{n}_v=(L-1,1)$, then the least lower-bound is obtained. An important point that should be mentioned is that the value of the entry e_{1,n_1+1} plays a significant role in the size of the lower-bound and also in $g(H_{m,E})$. That means, if this entry is changed, then the value of Q and in some cases the girth will be changed. By Algorithm 1, the least possible value of e_{1,n_1+1} is evaluated such that the least lower-bound is obtained.

5. **Conclusion**

In this paper, a recursive algorithm is presented for constructing $J \times L$ exponent matrices with girth at least 6. Then, corresponding to each exponent matrix E with girth at least 6, the lower-bound $Q = Q(E)$ is obtained such that the (J, L) QC LDPC codes with exponent matrix E have girth at least 6, for any $m \geq Q$. Tables 1-3 show that the constructed exponent matrices have smaller lower-bound rather than ones recently proposed by Karimi, et.al, [10], especially when $\mathbf{n}$, as one of the algorithm inputs, is equal to $(L-1, 1)$.

E	3×6	3×7	3×8	3×9	3×10	3×11	3×12	3×12	3×13	3×14	3×15	3×15
$\mathbf{n}_v$	$(2,2,2)$	$(3,1,3)$	$(4,2,2)$	$(4,1,4)$	$(5,5)$	$(4,3,4)$	$(4,4,4)$	$(11,1)$	$(7,6)$	$(3,6,5)$	$(4,7,4)$	$(14,1)$
Q	10	12	14	16	18	20	22	21	24	26	28	27
Q'	11	13	15	17	19	21	23	23	25	27	29	29
e_{1,n_1+1}	1	2	1	1	2	1	1	2	2	2	1	1

TABLE 1. Lower-bound of the constructed $3 \times L$ exponent matrices in comparison with [10], for different L and $\mathbf{n}_v$.

E	4×6	4×7	4×8	4×9	4×10	4×11	4×12	4×12	4×13	4×14	4×15	4×15
$\mathbf{n}_v$	$(5,1)$	$(4,2,1)$	$(7,1)$	$(8,1)$	$(8,2)$	$(2,3,4,2)$	$(4,2,4,2)$	$(11,1)$	$(12,1)$	$(13,1)$	$(4,3,3,5)$	$(14,1)$
Q	13	18	19	22	27	30	33	31	34	37	42	40
Q'	16	19	22	25	28	31	34	34	37	40	43	43
e_{1,n_1+1}	2	1	4	3	3	3	1	2	1	4	1	3

TABLE 2. Lower-bound of the constructed $4 \times L$ exponent matrices in comparison with [10], for different L and $\mathbf{n}_v$.

E	5×6	5×7	5×8	5×9	5×10	5×11	5×12	5×12	5×13	5×14	5×15	5×15
$\mathbf{n}_v$	$(5,1)$	$(4,3)$	$(7,1)$	$(8,1)$	$(2,2,2,2,2)$	$(8,3)$	$(3,2,1,4,2)$	$(11,1)$	$(12,1)$	$(5,4,5)$	$(3,4,1,3,4)$	$(14,1)$
Q	17	24	25	29	36	40	44	41	45	52	56	53
Q'	21	25	29	33	37	41	45	45	49	53	57	57
e_{1,n_1+1}	2	1	4	3	3	3	2	2	1	2	2	3

TABLE 3. Lower-bound of the constructed $5 \times L$ exponent matrices in comparison with [10], for different L and $\mathbf{n}_v$.

Acknowledgments

We would like to thank the anonymous referees for their helpful comments. This work was supported in part by the research council of Shahrekord university. Moreover, the fist author was in part supported by a grant from IPM (No. 93050065).

References

[1] R. G. Gallager, Low density parity check codes, *IRE Trans.*, **8** (1962) 21–28.

[2] R. M. Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory*, **27** (1981) 533–547.

[3] R. J. McEliece, D. J. C. Mackay and J. F. Cheng, Turbo decoding as instance of pearl's "belief propagation" algorithm, *IEEE J. Sel. Areas Commun*, **16** (1998) 140–152.

[4] J. L. Cheng, U. N. Peled, I. Perepelitsa and V. Pless, *Explicit construction of families of LDPC codes with girth at least six*, in Proc. 40th Annu. Allerton Conf. Communication, control and computing, Monticello, IL, (2002) 1024–1031.

[5] M. E. O'Sullivan , Algebraic construction of sparse matrices with large girth, *IEEE Trans. Inform. Theoy*, **8** (2006) 1788–1793.

[6] K. Lally, Explicit construction of Type-II QC-LDPC codes with girth at least 6, *IEEE Int. Symp. on Inf. Theory*, (2007) 2371–2375.

[7] B. Ammar, B. Honary, Y. Kou, J. Xu and S. Lin, Construction of low-density parity-check codes based on balanced incomplete block designs, *IEEE Trans. Inform. Theoy*, **50** (2004) 1257–1268.

[8] L. Lan, L. Zeng, Y. Y. Tai, L. Chen, S. Lin and K. A. Ghaffar, Construction of QC-LDPC codes for AWGN and binary erasure channels: A finite field approach, *IEEE Trans. Inform. Theoy*, **53** (2007) 2429–2457.

[9] C. M. Huang, J. F. Huang and C. C. Yang, Construction of QC-LDPC codes from quadratic congruences, *IEEE Comm. Lett.*, **12** (2008) 313–315.

[10] M. Karimi and A. H. Banihashemi, On the girth of quasi cyclic protograph LDPC codes, *IEEE Trans. Inform. Theoy*, **59** (2013) 4542–4552.

[11] G. Zhang, R. Sun and X. Wang, Construction of girth-eight QC-LDPC codes from greatest common divisor, *IEEE Comm. Lett.*, **17** (2013) 369–372.

Mohammad Gholami

Department of Mathematics, University of Shahrekord, P. O. Box 115, Shahrekord, Iran and
the School of Mathematics, Institute for Research in Fundamental Sciences (IPM), P. O. Box 19395-5746, Tehran, Iran
Email: gholami-m@sci.sku.ac.ir, gholamimoh@gmail.com

Zahra Rahimi

Department of Mathematics, University of Shahrekord, P. O. Box 115, Shahrekord, Iran
Email: zrahimi.1368@yahoo.com