IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 13, 2018, accepted March 16, 2018, date of publication March 27, 2018, date of current version May 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2819981

An Approach to Evaluating the Number of Closed Paths in an All-One Base Matrix

SHENG JIANG AND FRANCIS C. M. LAU, (Senior Member, IEEE)
[1]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Francis C. M. Lau (francis-cm.lau@polyu.edu.hk)

This work was supported by a grant from RGC of Hong Kong, China, under Project PolyU 152088/15E.

ABSTRACT Given an all-one base matrix of size $M \times N$, a closed path of different lengths can be formed by starting at an arbitrary element and moving horizontally and vertically alternatively before terminating at the same "starting" element. When the closed-path length is small, say 4 or 6, the total number of combinations can be evaluated easily. When the length increases, the computation becomes non-trivial. In this paper, a novel method is proposed to evaluate the number of closed paths of different lengths in an all-one base matrix. Theoretical results up to closed paths of length 10 have been derived and are verified by the exhaustive search method. Based on the theoretical work, results for closed paths of length larger than 10 can be further derived. Note that each of such closed paths may give rise to one or more cycles in a low-density parity-check (LDPC) code when the LDPC code is constructed by replacing each "1" in the base matrix with a circulant permutation matrix or a random permutation matrix. Since LPDC codes with short cycles are known to give unsatisfactory error correction capability, the results in this paper can be used to estimate the amount of effort required to evaluate the number of potential cycles of an LDPC code or to optimize the code.

INDEX TERMS All-one base matrix, closed path, cycles, low-density parity-check code.

I. INTRODUCTION

Much research has been conducted to construct high performance LDPC codes [1]–[6] where the girth — minimum cycle length — plays an important role. A cycle is a closed path in the Tanner graph which starts and ends in the same node. The path alternates between check and variable nodes [3], [4], [7]. The cycle can also be easily analyzed in the corresponding parity-check matrix because each check node in the Tanner graph corresponds to a row in the parity-check matrix, and each variable node corresponds to a column. If a variable node is connected to a check node, the corresponding element in the parity-check matrix will be "1". Similarly, a "0" in the matrix means the corresponding nodes are not connected. A cycle can thus be visualized as a path moving horizontally and vertically in an alternate manner along the "1"s in the matrix. Moreover, the path must be closed, that is to say, it starts and ends at the same "1" in the matrix. Obviously the length of a cycle must be an even number because the horizontal and vertical moves always appear in pairs. It is also easy to see that the minimum cycle length is 4.

One way to form an LDPC code is to construct a parity-check matrix of a particular size by assigning "1"s randomly with a certain probability. Such a construction method is not so desirable because the code is unstructured, making the encoding and decoding processes rather complicated to implement in terms of hardware. Another approach is to form a small-size base matrix first and then to replace each non-zero element in the base matrix with a circulant permutation matrix (CPM) or a random permutation matrix (RPM) or a sum/mix of both types of matrices [8], [9].

In this paper, we consider an all-one base matrix B of size $M \times N$. Moreover, we denote the "1" in the (i, j)-th position of the base matrix by $P_{i,j}$ $(1 \le i \le M, 1 \le j \le N)$, i.e.,

$$B = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \cdots & P_{1,N} \\ P_{2,1} & P_{2,2} & P_{2,3} & \cdots & P_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{M,1} & P_{M,2} & P_{M,3} & \cdots & P_{M,N} \end{bmatrix}. \quad (1)$$

A closed path of length $2l$ can therefore be described as $P_{i_1,j_1} \rightarrow P_{i_2,j_1} \rightarrow P_{i_2,j_2} \rightarrow P_{i_3,j_2} \rightarrow \cdots \rightarrow P_{i_l,j_l} \rightarrow P_{i_1,j_l} \rightarrow P_{i_1,j_1}$ where $i_1 \neq i_2 \neq i_3 \neq \cdots \neq i_{l-1} \neq i_l \neq i_1$ and $j_1 \neq j_2 \neq j_3 \neq \cdots \neq j_{l-1} \neq j_l \neq j_1$. In other words, the closed path starts from the element P_{i_1,j_1}, then moves vertically to P_{i_2,j_1}, then moves horizontally to P_{i_2,j_2}, . . ., and

finally goes back to the original element P_{i_1,j_1}.[1] Note that each of such closed paths may give rise to one or more cycles in the corresponding LDPC code when each "1" in the base matrix is replaced with a circulant permutation matrix (CPM) or a random permutation matrix (RPM). However in practice, with careful selection of the CPMs or RPMs by code designers, many of such cycles in the LDPC code can be avoided.

One simple way to determine the number of closed paths in the all-one based matrix is to use the "tree method" [7]. The main idea of the tree method is to construct trees whose roots are variable nodes. The tree is extended based on the Tanner graph of the codes. We start from one certain variable node (i.e., column), and put the check nodes (i.e., rows) that connect to the root variable node (column) in the next layer. After that, we add one more variable-node layer where all variable nodes are connected to the check nodes in the previous layer. The procedure of adding a layer is actually a move in the Tanner graph. Once the tree is expanded further enough, closed cycles of different lengths that start and end at the same root node can be found. The main drawback of the tree method is that the trees are always too large to manipulate. It works well for short closed paths and small base matrices. However, it takes a huge amount of space to store the tree and it costs a lot of time to perform the exhaustive searching for when the base matrix becomes large. The searching results also contain duplicates, which can only be eliminated by exhaustive comparisons.

In this paper, we propose a novel method to evaluate the total number of closed paths of different lengths in an all-one base matrix. Since LPDC codes with short cycles are known to give unsatisfactory error correction capability, the results in this paper can be used to estimate the amount of effort required to evaluate the number of potential cycles of an LDPC code or to optimize the code [10], [11]. The results are also verified by those found by the tree method.

II. TWO PRELIMINARY FUNCTIONS

Theorem 1: Suppose we have to assign v different digits to u consecutive slots where $u \geq v$ and $u, v \in \mathbb{Z}^+$. Moreover, consecutive slots must contain different digits and all v digits should be used. Then the total number of combinations equals

$$G(u, v) = v! \sum_{\Omega} \prod_{j=2}^{v} (j-1)^{\alpha_j} \tag{2}$$

where

$$\Omega = \{\{\alpha_j \in \mathbb{N} : j = 2, 3, \ldots, v\} : \sum_{j=2}^{v} \alpha_j = u - v\}. \tag{3}$$

Proof: Suppose we fill the slots one-by-one. Considering the first slot, i.e, slot $i = 1$, we arbitrarily pick one digit out of the v digits. As consecutive slots must contain different digits,

[1]Note that strictly speaking, the definitions of "cycle" and "close path" in this paper do not follow the standard definitions used in graph theory. The "path" and "cycle" defined here are referred to as, respectively, "walk" and "tailless closed non-reversing walk" in graph theory.

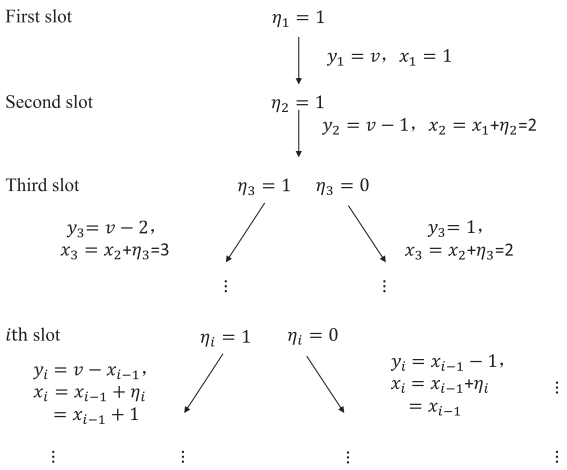

FIGURE 1. Illustration of the digit picking procedures.

we can only pick another digit out of the $v-1$ "unused" digits for the second slot, i.e, slot $i = 2$. (We call a digit unused if the digit never appears in previous slots.) For each of the remaining slots, i.e, slot $i = 3, 4, \ldots, u$, we always have two strategies: pick a digit we have used before or pick an unused digit. The only point we need to consider is to make sure that by the last slot, i.e, slot $i = u$, all v digits are used at least once. For $i = 1, 2, 3, \ldots, u$,

- let η_i represent the strategy used to select a digit for slot i: $\eta_i = 0$ means a used digit is selected and $\eta_i = 1$ indicates an unused digit is selected;
- y_i denote the number of possible digits to pick for slot i given η_i;
- x_i denote the number of distinct digits used up to and including slot i.

Note that

$$\sum_{i=1}^{u} \eta_i = v \tag{4}$$

because all the v digits must be used/selected at least once in the u slots.

Using the above notations, x_{i-1} distinct digits have been used up to slot $i-1$.

- Supposing $\eta_i = 0$ which means a used digit is to be picked for slot i, the number of possible digits to pick equals $y_i = x_{i-1} - 1$ because the digit must different be from the digit in slot $i-1$. Moreover, the number of distinct digits used remains the same and hence $x_i = x_{i-1}$.
- Supposing $\eta_i = 1$ which means an unused digit is to be picked for slot i, the number of possible digits to pick equals $y_i = v - x_{i-1}$ and the number of distinct digits used is increased by one, i.e., $x_i = x_{i-1} + 1$.

In both cases described above, we have $x_i = x_{i-1} + \eta_i$. The digit picking procedures are described with a tree shown in Fig. 1. Further, two different digit selection sequences for $u = 5$ and $v = 4$ are shown in Table 1. The difference between two selection sequences is the slot number that a used digit is selected. In sequences #1 and #2, a used digit in selected in slot 3 and slot 5, respectively.

TABLE 1. Two different digit selection sequences for $u = 5$ and $v = 4$.

(a)

i	η_i	y_i	x_i
1	1	4	1
2	1	3	2
3	0	1	2
4	1	2	3
5	1	1	4

(b)

i	η_i	y_i	x_i
1	1	4	1
2	1	3	2
3	1	2	3
4	1	1	4
5	0	3	4

Supposing $\eta_i = 0$, then $y_i = x_{i-1} - 1 = x_i - 1 = j - 1$ for some $j \in \{2, \ldots, v\}$. Moreover, there are u slots and v digits and hence reused digits must be selected $u - v$ times, i.e., $\eta_i = 0$ must occur $u - v$ times. Denote α_j $(j = 2, \ldots, v)$ as the number of occurrences of the event $\{\eta_i = 0, x_i = j\}$, i.e., $\eta_i = 0$ AND $x_i = j$, among all $i = 1, 2, \ldots, u$. Combining the above, we have

$$\sum_{j=2}^{v} \alpha_j = u - v. \tag{5}$$

Note that the solution set $\{\alpha_j : j = 2, \ldots, v\}$ for (5) is usually not unique. In the digit selection sequence #1 shown in Table 1, $\eta_i = 0$ only when $i = 3$. Hence, $x_3 = 2$ and subsequently $\alpha_2 = 1, \alpha_3 = \alpha_4 = 0$. In sequence #2, $\eta_i = 0$ only when $i = 5$. Hence, $x_5 = 4$ and subsequently $\alpha_4 = 1$, $\alpha_2 = \alpha_3 = 0$. In both cases, we have $\sum_{j=2}^{v} \alpha_j = u - v = 1$.

For a given set $\Theta = \{\alpha_j : j = 2, \ldots, v\}$ that satisfies (5), it can be readily shown that the total number of possible digit-sequence selections for the $u - v$ slots where used digits are selected equals

$$\prod_{\{i\in\{1,2,\ldots,u\}:\eta_i=0\}} y_i = \prod_{j=2}^{v} (j-1)^{\alpha_j}. \tag{6}$$

For the remaining v slots where unused digits are selected $(\eta_i = 1)$, the number of choices is decreased by one every time and hence the total number of possible digit-sequence selections equals

$$\prod_{\{i\in\{1,2,\ldots,u\}:\eta_i=1\}} y_i = v(v-1)(v-2)\ldots\times 2\times 1 = v! \tag{7}$$

Combining all the above results, the total number of combinations is hence given by

$$\begin{aligned} G(u,v) &= \sum_{\Theta\in\Omega}\prod_{i=1}^{u} y_i \\ &= \sum_{\Theta\in\Omega}\left[\left(\prod_{\{i\in\{1,2,\ldots,u\}:\eta_i=1\}} y_i\right)\left(\prod_{\{i\in\{1,2,\ldots,u\}:\eta_i=0\}} y_i\right)\right] \\ &= v!\sum_{\Theta\in\Omega}\prod_{j=2}^{v}(j-1)^{\alpha_j} \end{aligned} \tag{8}$$

where $\Omega = \{\{\alpha_j \in \mathbb{N} : j = 2, 3, v\} : \sum_{j=2}^{v} \alpha_j = u - v\}\}$ denotes the solution sets of $\{\alpha_j\}$. □

Theorem 2: If we impose an additional condition on Theorem 1 that the last slot must contain a different digit from the first one, the total number of combinations becomes

$$G'(u,v) = v!\sum_{\Omega}\frac{(v-1)^{\alpha_v+1} + (-1)^{\alpha_v}}{v}\prod_{j=2}^{v-1}(j-1)^{\alpha_j} \tag{9}$$

where

$$\Omega = \{\{\alpha_j \in \mathbb{N} : j = 2, 3, v\} : \sum_{j=2}^{v} \alpha_j = u - v\}\}. \tag{10}$$

Proof: We use the same notations as in the previous proof. Most of the proof of Theorem 2 is similar to that of Theorem 1 except when dealing with the last slot, in which the digit must now be different from the digit in the first slot.

Among the u slots, there are v slots where $\eta_i = 1$ because all the v distinct digits must be used at least once. Let γ denote the position of the last slot where $\eta_i = 1$. Then, $\eta_\gamma = 1$ and for all subsequent slots, i.e., slots $i = \gamma + 1, \gamma + 2, \ldots, u$ (altogether $u - \gamma$ slots), we always have $\eta_i = 0$ and $x_i = v$. Recall that α_j $(j = 2, \ldots, v)$ denotes the number of occurrences of the event $\{\eta_i = 0, x_i = j\}$. Thus, in the case of $j = v$, we have $\alpha_v = u - \gamma$. Also, the last α_v slots all contain used digits.

Denote $f(\alpha_v)$ as the total number of choices for the last α_v consecutive slots. Thus

$$f(\alpha_v) = \prod_{\{i\in\{\gamma+1,\ldots,u\}:\eta_i=0\}} y_i = \prod_{i\in\{\gamma+1,\ldots,u\}} y_i. \tag{11}$$

(Note that in Theorem 1, $f(\alpha_v) = (v-1)^{\alpha_v}$ and equals 1 when $\alpha_v = 0$.)

- If $\alpha_v = 1$, the $(u-1)$-th slot contains an unused digit and the last (i.e., u-th) slot contains a used digit. Since the $(u-1)$th slot contains an unused digit, this digit is different from the digit in the first (i.e., $i = 1$) slot. As the digit in the last slot must be different from that in the $(u-1)$th slot and first slot, it has $v-2$ choices and hence

$$f(\alpha_v = 1) = v - 2. \tag{12}$$

- If $\alpha_v = 2$, the $(u-2)$-th slot contains an unused digit and the last two slots (i.e., $(u-1)$-th and u-th slots) contain used digits. We use similar arguments as above. Since the $(u-2)$th slot contains an unused digit, this digit is different from the digit in the first (i.e., $i = 1$) slot. We then consider the $(u-1)$-th slot. If it contains the same digit as the one in the first slot (only one choice), the u-th slot has $v-1$ choices; otherwise the digit in the $(u-1)$-th slot must be different from that in the first slot and the u-th slot ($v-2$ choices), and then the u-th slot will have $v-2$ choices. The total number of choices for the last $\alpha_v = 2$ consecutive slots therefore equals

$$\begin{aligned} f(\alpha_v = 2) &= 1\times(v-1) + (v-2)\times(v-2) \\ &= v^2 - 3v + 3. \end{aligned} \tag{13}$$

- We consider the general case where $\alpha_v \geq 3$. The γ-th (i.e, $(u-\alpha_v)$-th) is the last slot where an unused digit is selected.
  1) If the $(\gamma+1)$-th slot is the same as the first slot (1 choice), the $(\gamma+2)$-th slot must be different from the first slot ($v-1$ choices) and from the $(\gamma+3)$-th to u-th slots, the number of choices is given by $f(\alpha_v-2)$.
  2) If the $(\gamma+1)$-th slot is different from the first slot ($v-2$ choices), the number of choices from the $(\gamma+2)$-th to u-th slots is given by $f(\alpha_v-1)$.

  The total number of choices thus equals

$$\begin{aligned} f(\alpha_v) &= 1\times(v-1)\times f(\alpha_v-2)+(v-2)\times f(\alpha_v-1) \\ &= (v-1)\times f(\alpha_v-2)+(v-2)\times f(\alpha_v-1). \end{aligned} \tag{14}$$

Using (12), (13) and (14), it can be readily shown that

$$f(\alpha_v)=\frac{(v-1)^{\alpha_v+1}+(-1)^{\alpha_v}}{v} \quad \forall\, \alpha_v=1,2,\ldots \tag{15}$$

and hence (11) can be written as

$$\prod_{\{i\in\{\gamma+1,\ldots,u\}:\eta_i=0\}} y_i=\frac{(v-1)^{\alpha_v+1}+(-1)^{\alpha_v}}{v} \quad \forall\, \alpha_v=1,2,\ldots \tag{16}$$

where $\gamma=u-\alpha_v$.

For a given set $\Theta=\{\alpha_j : j=2,\ldots,v\}$ that satisfies (5) and the given conditions (particularly that the last slot must contain a different digit from the first one), the total number of possible digit-sequence selections for the $u-v$ slots where used digits are selected equals

$$\begin{aligned} &\prod_{\{i\in\{1,2,\ldots,u\}:\eta_i=0\}} y_i \\ &= \left(\prod_{\{i\in\{1,2,\ldots,\gamma\}:\eta_i=0\}} y_i\right)\times\left(\prod_{\{i\in\{\gamma+1,\ldots,u\}:\eta_i=0\}} y_i\right) \\ &= \left(\prod_{j=2}^{v-1}(j-1)^{\alpha_j}\right)\times\left(\frac{(v-1)^{\alpha_v+1}+(-1)^{\alpha_v}}{v}\right). \end{aligned} \tag{17}$$

Combining the above results with (7), the total number of combinations is hence given by

$$\begin{aligned} G'(u,v) &= \sum_{\Theta\in\Omega}\prod_{i=1}^{u} y_i \\ &= v!\sum_{\Omega}\left(\frac{(v-1)^{\alpha_v+1}+(-1)^{\alpha_v}}{v}\prod_{j=2}^{v-1}(j-1)^{\alpha_j}\right). \end{aligned} \tag{18}$$

(Note also that (18) still holds even when $\alpha_v=0$. It is because according to (15), $f(\alpha_v)=1$ when $\alpha_v=0$.) □

Table 2 and 3 show the values of $G(u,v)$ and $G'(u,v)$, respectively, for $2\leq v\leq u\leq l=7$. These values are sufficient when considering cycles with length no longer than 14. Note also that a quasi-cyclic LDPC code constructed from an all-one base matrix has a girth bounded by 12. For a fixed $v\geq 3$, we can see that both $G(u,v)$ and $G'(u,v)$ increase exponentially with u.

TABLE 2. $G(u,v)$ function.

G	$v=2$	3	4	5	6	7
$u=2$	2					
3	2	6				
4	2	18	24			
5	2	42	144	120		
6	2	90	600	1,200	720	
7	2	186	2,160	7,800	10,800	5,040

TABLE 3. G' function.

G'	$v=2$	3	4	5	6	7
$u=2$	2					
3	0	6				
4	2	12	24			
5	0	30	120	120		
6	2	60	480	1,080	720	
7	0	126	1,680	6,720	10,080	5,040

III. CONFIGURATIONS OF CLOSED PATHS

Referring to Fig. 2, we extract an $m\times n$ sub-matrix from the $M\times N$ all-one base matrix and represent the (i,j)-th element in the sub-matrix by $Q_{i,j}$ $(1\leq i\leq m,\ 1\leq j\leq n)$. We define a closed path (CP) as an (m,n)-CP if it passes m distinct rows and n distinct columns of an all-one matrix. Denoting an (m,n)-CP by $Q_{i_1,j_1}\to Q_{i_2,j_1}\to Q_{i_2,j_2}\to Q_{i_3,j_2}\to\cdots\to Q_{i_l,j_l}\to Q_{i_1,j_l}\to Q_{i_1,j_1}$ where $i_1\neq i_2\neq i_3\neq\cdots\neq i_{l-1}\neq i_l\neq i_1$ and $j_1\neq j_2\neq j_3\neq\cdots\neq j_{l-1}\neq j_l\neq j_1$, we represent the l-tuple row-index vector and column-index vector of the (m,n)-CP by, respectively, $I=(i_1,i_2,\ldots,i_{l-1},i_l)$ and $J=(j_1,j_2,\ldots,j_{l-1},j_l)$. Since an (m,n)-CP passes m rows and n columns, there are exactly m distinct values in the row-index vector I and n distinct values in the column-index vector J.

For a given $M\times N$ all-one base matrix and a given cycle length $2l$, the values of m and n are constrained by

$$2\leq m\leq \min(M,l) \tag{19}$$

$$2\leq n\leq \min(N,l). \tag{20}$$

For every (m,n) satisfying (19) and (20), the numbers of row-index and column-index combinations can be calculated by using the G' function in (9). Denoting $R(l,m)$ as the number of row-index combinations and $C(l,n)$ as the number of column-index combinations for an (m,n)-CP configuration with length $2l$, we have

$$R(l,m)=G'(l,m) \tag{21}$$

$$C(l,n)=G'(l,n). \tag{22}$$

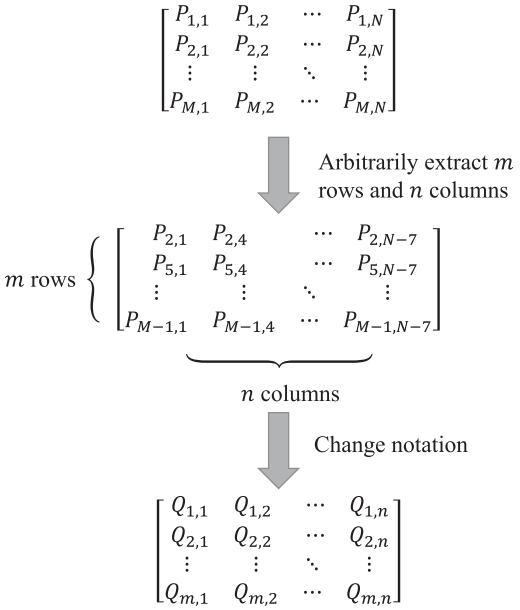

FIGURE 2. Extracting an $m \times n$ sub-matrix from an all-one $M \times N$ base matrix.

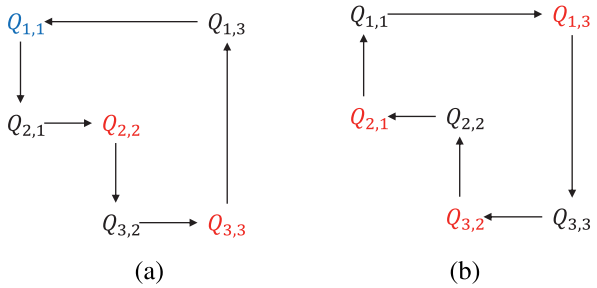

FIGURE 3. Illustration of duplicated closed-paths of length 6. CPs starting from $Q_{2,2}$, $Q_{3,3}$, $Q_{1,3}$, $Q_{3,2}$ and $Q_{2,1}$ are duplicates of that starting from $Q_{1,1}$.

The total number of (m, n)-CP configurations with length $2l$ therefore equals $R(l, m)C(l, n) = G'(l, m)G'(l, n)$ when duplicated CPs such as those shown in Fig. 3 are not eliminated.

IV. DUPLICATED CLOSED PATHS

Given two different row-index/column-index vector pairs (I, J) and (I', J'), they may represent the same CP. Fig. 3 shows an example where the same CP of length 6 is formed with different starting element and hence different row-index/column-index vector pairs. In the following, we will remove such duplicates in our computation of the number of CPs.

A. ELIMINATE CLOSED PATHS THAT DO NOT START FROM THE FIRST ROW

First, we eliminate CPs that do not start from the first row (e.g., eliminate CPs that start with elements $Q_{2,2}$, $Q_{3,3}$, $Q_{3,2}$ or $Q_{2,1}$ in Fig. 3). We therefore add a constraint that every CP must start from the first row of the

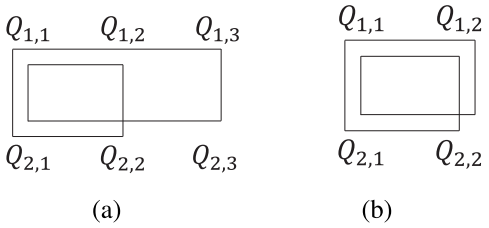

FIGURE 4. Two different CP configurations each of length 8. (a) No transform-exact pair exists, and (b) transform-exact pairs exist.

$m \times n$ sub-matrix. That is to say, we require $i_1 = 1$ for every CP configuration.

Denote $R'(l, m, t)$ as the number of row-index combinations with the new constraint where t represents the number of paths along the first row of the CP configuration. In fact, t also represents the number of "1"s in the row-index vector I. Since consecutive elements in I must differ, there are at most $\lfloor \frac{l}{2} \rfloor$ "1"s in I and thus $1 \le t \le \lfloor \frac{l}{2} \rfloor$. Moreover, t is restricted by $t \le l - m + 1$. To summarize, we have

$$t \le \min\left(\left\lfloor \frac{l}{2} \right\rfloor, l - m + 1\right). \tag{23}$$

Figs. 3 and 4 illustrate, respectively, the cases where $t = 1$ and $t = 2$. In this paper we only consider CPs with length $2l$ ranging from 4 to 10. Thus $2 \le l \le 5$ and $1 \le t \le 2$. For larger values of l and t, the guiding principles are the same and the results can be derived in a similar manner.

Since $i_1 = 1$, we only need to assign values to $i_2, i_3, \ldots, i_l$.

1) When $t = 1$, we assign all $m-1$ digits to the $l-1$ slots: $i_2, i_3, \ldots, i_l$ and make sure $i_2 \ne i_3 \ne i_4 \ne \cdots \ne i_l$. According to Theorem 1, we have

$$R'(l, m, 1) = G(l - 1, m - 1). \tag{24}$$

2) When $t = 2$, one and only one element of $i_3, \ldots, i_{l-1}$ equals 1. Assuming $i_\xi = 1$, the row-index vector becomes $I = (1, i_2, i_3, \ldots, i_{\xi-1}, 1, i_{\xi+1}, \ldots, i_l)$.
   - If $i_{\xi-1} \ne i_{\xi+1}$, it is equivalent to assigning $m - 1$ different digits to $i_2, i_3, \ldots, i_{\xi-1}, i_{\xi+1}, i_{\xi+2}, \ldots, i_l$ with different consecutive elements. According to Theorem 1, there are $G(l-2, m-1)$ combinations.
   - If $i_{\xi-1} = i_{\xi+1}$ then it is equivalent to assign $m - 1$ different digits to $i_2, i_3, \ldots, i_{\xi-1}, i_{\xi+2}, i_{\xi+3}, \ldots, i_l$ with different consecutive elements. According to Theorem 1, there are $G(l-3, m-1)$ combinations. In such a situation, $m < l - 1$.

   Combining the above two scenarios, we have

$$R'(l, m, 2) = \begin{cases} 0 & \text{if } m > l-1 \\ G(l-2, m-1) & \text{if } m = l-1 \\ G(l-2, m-1) + G(l-3, m-1) & \text{if } m < l-1. \end{cases} \tag{25}$$

B. ELIMINATE DUPLICATED CLOSED PATHS THAT START FROM THE FIRST ROW

After the new constraint in the previous section has been applied, most of the duplicated CPs are eliminated. Some duplicates still remain due to the fact that more than one elements in the first row can act as the starting element. For example in Fig. 3(b), the CP $Q_{1,3} \to Q_{3,3} \to Q_{3,2} \to Q_{2,2} \to Q_{2,1} \to Q_{1,1}$ is a duplicated version of $Q_{1,1} \to Q_{2,1} \to Q_{2,2} \to Q_{3,2} \to Q_{3,3} \to Q_{1,3}$ while both start from the first row.

Assume that we have an (I, J) pair denoted by

$$\begin{cases} I = (i_1, i_2, \ldots, i_{l-1}, i_l) \\ J = (j_1, j_2, \ldots, j_{l-1}, j_l). \end{cases} \tag{26}$$

Then the "reversed" version of the (I, J) pair will always correspond to the same CP because it is equivalent to indexing the same CP from the last element to the first one. Denoting the reversed pair as (I', J'), we have

$$\begin{cases} I' = (i_1, i_l, i_{l-1}, \ldots, i_3, i_2) \\ J' = (j_l, j_{l-1}, \ldots, j_2, j_1). \end{cases} \tag{27}$$

Moreover, an (m, n)-CP of length $2l$ can start from any of the $2l$ elements. If I and J are cyclically shifted with same number of positions, the CP that the new pair represents will be identical to the original one. In the following, we define an operator that performs transformation on the (I, J) pair.

Definition 1: For any (I, J) pair with the form of (26), we define $F(I, J, k)$ as the transformation of the (I, J) pair such as all elements in I and J are cyclically shifted to the left by k positions ($k = 1, 2, \ldots, l-1$). Let $(I_k, J_k) = F(I, J, k)$. Then

$$\begin{cases} I_k = (i_{k+1}, i_{k+2}, \ldots, i_l, i_1, i_2, \ldots, i_k) \\ J_k = (j_{k+1}, j_{k+2}, \ldots, j_l, j_1, j_2, \ldots, j_k). \end{cases} \tag{28}$$

It can be easily shown that all the transformed pairs represent the same CP as the original one. Similarly, we have $(I'_k, J'_k) = F(I', J', k)$ for $k = 1, 2, \ldots, l-1$. All these l pairs also represent the same CP as the (I, J) pair. In summary, for a given (I, J) pair, there are another $2l-1$ (I'', J'') transformed pairs that give rise to the same CP. In the previous section, we have already eliminated duplicates that do not start from the first row. So we only need to consider transformed pairs whose CPs start from the first row. To analyze such cases, we make use of the following theorem.

Theorem 3: For an (I, J) pair with t "1"s in I, there are $2t-1$ transformed pairs whose CPs start from the first row.

Proof: Obviously (I', J') is a transformed pair that starts from the first row. Assume that the t "1"s are located at the 1st, n_2-th, n_3-th, ..., n_t-th positions in I. Then (I_{n_k-1}, J_{n_k-1}) and $(I'_{l-n_k+1}, J'_{l-n_k+1})$ for $k = 2, 3, \ldots, t$ are all transformed pairs that starts from the first row. So altogether there are $2t-1$ such transformed pairs. □

When all the transformed pairs are different from the original pair, we call the original pair (I, J) "transform-equivalent". According to Theorem 3, there are exact $2t-1$ duplicates for each transform-equivalent pair. In some occasions, some of the $2t-1$ transformed pairs are exactly the same as the original (I, J) pair. In such cases, we call the original (I, J) "transform-exact". To find out the duplicates, we have to consider each transform-exact pair separately. Fig. 4 illustrates the CP configurations with and without transform-exact pair.

Considering the CP configuration shown in Fig. 4(a), we denote

$$\begin{cases} I = (1, 2, 1, 2) \\ J = (1, 3, 1, 2). \end{cases} \tag{29}$$

Hence, the 3 transformed pairs are given by

$$\begin{cases} I' = (1, 2, 1, 2) \\ J' = (2, 1, 3, 1) \end{cases} \quad \begin{cases} I_2 = (1, 2, 1, 2) \\ J_2 = (1, 2, 1, 3) \end{cases} \quad \begin{cases} I'_2 = (1, 2, 1, 2) \\ J'_2 = (3, 1, 2, 1). \end{cases} \tag{30}$$

Since all 4 column-index vectors J, J', J_2, J'_2 are different, (I, J) is a transform-equivalent pair and has 3 duplicates.

For the CP configuration shown in Fig. 4(b), we denote

$$\begin{cases} I = (1, 2, 1, 2) \\ J = (1, 2, 1, 2) \end{cases} \tag{31}$$

The 3 transformed pairs are therefore given by

$$\begin{cases} I' = (1, 2, 1, 2) \\ J' = (2, 1, 2, 1) \end{cases} \quad \begin{cases} I_2 = (1, 2, 1, 2) \\ J_2 = (1, 2, 1, 2) \end{cases} \quad \begin{cases} I'_2 = (1, 2, 1, 2) \\ J'_2 = (2, 1, 2, 1) \end{cases} \tag{32}$$

Since $(I, J) = (I_2, J_2)$ and $(I', J') = (I'_2, J'_2)$, (I, J) is a transform-exact pair and has only 1 duplicate.

We denote $D_{ex}(l, m, n, t)$ and $D_{eq}(l, m, n, t)$, respectively, as the number of transform-exact and transform-equivalent pairs for an (m, n)-CP with length $2l$. We also denote $D_{sum}(l, m, n, t)$ as the total number of distinct (i.e., no duplicates) (m, n)-CP with length $2l$ and t "1"s in I. Then for $t = 1$ and 2, we have the following.

1) T = 1

We denote the index vectors as

$$\begin{cases} I = (1, i_2, \ldots, i_{l-1}, i_l) \\ J = (j_1, j_2, \ldots, j_{l-1}, j_l). \end{cases} \tag{33}$$

According to Theorem 3, there is 1 ($= 2t-1$) transformed pair that may be duplicated. Obviously, the transformed pair is given by

$$\begin{cases} I' = (1, i_l, i_{l-1}, \ldots, i_3, i_2) \\ J' = (j_l, j_{l-1}, j_{l-2} \ldots, j_2, j_1). \end{cases} \tag{34}$$

If $l \equiv 0 \pmod 2$, then

$$J = (j_1, j_2, \ldots, j_{l/2}, j_{(l/2)+1}, \ldots, j_{l-1}, j_l) \tag{35}$$

$$J' = (j_l, j_{l-1}, \ldots, j_{(l/2)+1}, j_{l/2}, \ldots, j_2, j_1). \tag{36}$$

The $(l/2)$-th elements of J and J' are $j_{l/2}$ and $j_{(l/2)+1}$ respectively. Since $j_{l/2} \neq j_{(l/2)+1}$, $J \neq J'$ and $(I, J) \neq (I', J')$.

If $l \equiv 1 \pmod 2$, it can proved in a similar way that $I \neq I'$ and $(I, J) \neq (I', J')$.

To sum up, when $t = 1$, all (I, J) pairs are transform-equivalent and there is 1 $(= 2t - 1)$ duplicate for each (I, J) pair. As a result,

$$D_{eq}(l, m, n, 1) = R'(l, m, 1)C(l, n) \tag{37}$$

$$D_{ex}(l, m, n, 1) = 0 \tag{38}$$

$$D_{sum}(l, m, n, 1) = \frac{D_{eq}(l, m, n, 1)}{2}. \tag{39}$$

2) T = 2

Besides the first element in I, another element among $i_3, \ldots, i_{l-1}$ equals 1. Assuming $i_\xi = 1$, we denote the index vectors by

$$\begin{cases} I = (1, i_2, i_3, \ldots, i_{\xi-1}, 1, i_{\xi+1}, \ldots, i_l) \\ J = (j_1, j_2, \ldots, j_l). \end{cases} \tag{40}$$

According to Theorem 3, there are 3 $(= 2t-1)$ transformed pairs that may be duplicated. These pairs are given by

$$\begin{cases} I' = (1, i_l, i_{l-1}, \ldots, i_{\xi+1}, 1, i_{\xi-1}, \ldots, i_3, i_2) \\ J' = (j_l, j_{l-1}, \ldots, j_2, j_1) \end{cases} \tag{41}$$

$$\begin{cases} I_{\xi-1} = (1, i_{\xi+1}, \ldots, i_l, 1, i_2, i_3, \ldots, i_{\xi-1}) \\ J_{\xi-1} = (j_\xi, j_{\xi+1}, \ldots, j_l, j_1, j_2, j_3 \ldots, j_{\xi-1}) \end{cases} \tag{42}$$

$$\begin{cases} I'_{\xi-1} = (1, i_{\xi-1}, i_{\xi-2}, \ldots, i_3, i_2, 1, i_l, i_{l-1}, \ldots, i_{\xi+2}, i_{\xi+1}) \\ J'_{\xi-1} = (j_{\xi-1}, j_{\xi-2}, \ldots, j_2, j_1, j_l, j_{l-1}, \ldots, j_{\xi+1}, j_\xi). \end{cases} \tag{43}$$

1) Using a similar proof as in the case $t = 1$, it can be shown that $(I, J) \neq (I', J')$.
2) Next we consider (I, J) and $(I'_{\xi-1}, J'_{\xi-1})$. If we assume $I = I'_{\xi-1}$, then $(i_2, i_3, \ldots, i_{\xi-2}, i_{\xi-1}) = (i_{\xi-1}, i_{\xi-2}, \ldots, i_3, i_2)$. Since consecutive elements in the index vectors must be different, the number of elements in $(i_2, i_3, \ldots, i_{\xi-2}, i_{\xi-1})$ must be odd. Thus $\xi - 2$ is an odd number and so is ξ. If we further assume $J = J'_{\xi-1}$, then $(j_1, j_2, \ldots, j_{\xi-2}, j_{\xi-1}) = (j_{\xi-1}, j_{\xi-2}, \ldots, j_2, j_1)$. As a result, $\xi - 1$ should be an odd number and ξ should be even. This contradicts with the above requirement that ξ should be odd. Therefore, $I = I'_{\xi-1}$ and $J = J'_{\xi-1}$ cannot be true simultaneously and $(I, J) \neq (I'_{\xi-1}, J'_{\xi-1})$.
3) Finally we consider the pairs (I, J) and $(I_{\xi-1}, J_{\xi-1})$. If $(I, J) = (I_{\xi-1}, J_{\xi-1})$, it can be easily seen that $(I', J') = (I'_{\xi-1}, J'_{\xi-1})$. Under this circumstance, there can only be one duplicate.

In the following, we evaluate the exact numbers of CP-configurations that belong to the transform-exact category and transform-equivalent category, respectively.

Transform-Exact Category: Suppose $I = I_{\xi-1}$, we have $i_2 = i_{\xi+1}$, $i_3 = i_{\xi+2}, \cdots$, and $i_{\xi-1} = i_l$. Obviously $\xi - 1 - 2 = l - (\xi + 1)$ and hence $\xi = \frac{l+2}{2}$, which means the positions of "1"s in the row-index vector I are fixed. Since there are m distinct digits in I, there are $m - 1$ distinct digits in the vector $(i_2, i_3, \ldots, i_{\xi-1})$. The number of elements in the vector equals $\xi - 2 = \frac{l-2}{2}$. According to Theorem 1, we have $G(\frac{l-2}{2}, m-1)$ such row-index vectors. If $J = J_{\xi-1}$, we further have $j_1 = j_\xi$, $j_2 = j_{\xi+1}, \cdots, j_{\xi-1} = j_l \neq j_1$. We need to assign n distinct digits to the column-index vector $(j_1, j_2, \ldots, j_{\xi-1})$ which has $\xi - 1 = \frac{l}{2}$ elements. We also need to make sure $j_1 \neq j_{\xi-1}$. According to Theorem 2, we have $G'(\frac{l}{2}, n)$ such column-index vectors. Hence,

$$D_{ex}(l, m, n, 2) = G\left(\frac{l-2}{2}, m-1\right) G'\left(\frac{l}{2}, n\right). \tag{44}$$

For the transform-exact pairs with the above row-index vectors and column-index vectors, each pair has only one duplicate. Therefore, the number of CPs without duplicates equals $\frac{D_{ex}(l,m,n,2)}{2}$.

Transform-Equivalent Category: Since the total number of transform-exact pairs and transform-equivalent pairs equals $R'(l, m, 2)C(l, n)$, the number of transform-equivalent pairs equals

$$D_{eq}(l, m, n, 2) = R'(l, m, 2)C(l, n) - D_{ex}(l, m, n, 2). \tag{45}$$

For the transform-equivalent pairs with the above row-index vectors and column-index vectors, each pair has 3 duplicates. Therefore, the number of such CPs without duplicates equals $\frac{D_{eq}(l,m,n,2)}{4}$.

To summarize, when $t = 2$, the total number of distinct CPs equals

$$D_{sum}(l, m, n, 2) = \frac{D_{eq}(l, m, n, 2)}{4} + \frac{D_{ex}(l, m, n, 2)}{2}. \tag{46}$$

C. OVERALL RESULTS

When $t = 1, 2$, the total number of distinct (m, n)-CP with length $2l$ is given by

$$\begin{aligned} D_{sum}(l, m, n, t) &= \frac{D_{eq}(l, m, n, t)}{2t} + \frac{D_{ex}(l, m, n, t)}{t} \\ &= \frac{R'(l, m, t)C(l, n) + D_{ex}(l, m, n, t)}{2t} \end{aligned} \tag{47}$$

where

$$D_{ex}(l, m, n, 1) = 0 \tag{48}$$

$$D_{ex}(l, m, n, 2) = \begin{cases} 0 & \text{if } l \equiv 1 \pmod 2 \\ G(\frac{l-2}{2}, m-1)G'(\frac{l}{2}, n) & \text{if } l \equiv 0 \pmod 2 \end{cases} \tag{49}$$

and other expressions are defined in (2), (9), (22), (24) and (25).

V. NUMBER OF CLOSED CYCLES WITH DIFFERENT LENGTHS

In this section, we will evaluate the number of CPs with length ranging from 4 to 10. For an $M \times N$ all-one base matrix, we denote $S_{dup}(l, M, N)$ as the number of length-$2l$

TABLE 4. Number of CPs of different lengths under different base-matrix sizes.

CP length	Base matrix size ($M \times N$)	No. of CPs obtained by the proposed method with (without) duplicates removed	No. of CPs obtained by the "Tree Method"
4	2×3	3 (12)	3
6	3×4	24 (144)	24
8	4×5	2,760 (21,840)	2,760
10	5×5	104,040 (1,040,400)	104,040
10	4×24	154,471,680 (1,544,716,800)	NA
10	5×20	252,560,160 (2,525,601,600)	NA

CPs with duplicates and $S(l, M, N)$ as the number of distinct length-$2l$ CPs. Then we have

$$S_{dup}(l, M, N) = \sum_{m=2}^{\min(M,l)} \sum_{n=2}^{\min(N,l)} C(l, n) R(l, m) C_M^m C_N^n \quad (50)$$

$$S(l, M, N) = \sum_{m=2}^{\min(M,l)} \sum_{n=2}^{\min(N,l)} \sum_{t=1}^{\min(\lfloor \frac{l}{2} \rfloor, l-m+1)} D_{sum}(l, m, n, t) C_M^m C_N^n \quad (51)$$

where $C_K^k = \frac{K!}{(K-k)!k!}$.

A. LENGTH-4 CPS

For CPs with length 4, we have $l = 2$, $m = n = 2$, and $t = 1$. Thus for $N \geq M \geq 2$,

$$S_{dup}(2, M, N) = 4C_M^2 C_N^2 \quad (52)$$

$$S(2, M, N) = C_M^2 C_N^2. \quad (53)$$

B. LENGTH-6 CPS

For CPs with length 6, we have $l = 3$, $2 \leq m, n \leq 3$, and $t = 1$. Thus for $N \geq M \geq 3$,

$$S_{dup}(3, M, N) = 36C_M^3 C_N^3 \quad (54)$$

$$S(3, M, N) = 6C_M^3 C_N^3. \quad (55)$$

C. LENGTH-8 CPS

For CPs with length 8, we have $l = 4, 2 \leq m, n \leq 4, t = 1, 2$. Thus for $N \geq M \geq 4$,

$$\begin{aligned} S_{dup}(4, M, N) = {} & C_M^2 (4C_N^2 + 24C_N^3 + 48C_N^4) \\ & + C_M^3 (24C_N^2 + 144C_N^3 + 288C_N^4) \\ & + C_M^4 (48C_N^2 + 288C_N^3 + 576C_N^4) \end{aligned} \quad (56)$$

$$\begin{aligned} S(4, M, N) = {} & C_M^2 (C_N^2 + 3C_N^3 + 6C_N^4) \\ & + C_M^3 (3C_N^2 + 18C_N^3 + 36C_N^4) \\ & + C_M^4 (6C_N^2 + 36C_N^3 + 72C_N^4). \end{aligned} \quad (57)$$

D. LENGTH-10 CPS

For CPs with length 10, we have $l = 5$, $2 \leq m, n \leq 5$, $t = 1, 2$. Thus for $N \geq M \geq 5$,

$$\begin{aligned} S_{dup}(5, M, N) = {} & C_M^3 (900C_N^3 + 3600C_N^4 + 3600C_N^5) \\ & + C_M^4 (3600C_N^3 + 14400C_N^4 + 14400C_N^5) \\ & + C_M^5 (3600C_N^3 + 14400C_N^4 + 14400C_N^5) \end{aligned} \quad (58)$$

$$\begin{aligned} S(5, M, N) = {} & C_M^3 (90C_N^3 + 360C_N^4 + 360C_N^5) \\ & + C_M^4 (360C_N^3 + 1440C_N^4 + 1440C_N^5) \\ & + C_M^5 (360C_N^3 + 1440C_N^4 + 1440C_N^5). \end{aligned} \quad (59)$$

In the above equations, we set $C_K^k = 0$ for $k > K$ which occurs when the base matrix is not large enough to generate the CPs. The above equations also provide the number of different CP configurations of a given length under a specific sub-matrix size. For example in (59), the term 360 $C_M^4 C_N^3$ implies that for a sub-matrix with size 4×3, there are 360 different CP configurations with length 10. Similarly, the term 1440 $C_M^5 C_N^4$ indicates that for a sub-matrix with size 5×4, there are 1440 different CP configurations with length 10.

Table 4 shows the number of CPs evaluated based on our proposed method for different path lengths under different base-matrix sizes. For example, with a base matrix of size 5×5, the number of CPs of length 10 calculated using our method equals 104, 040 (= $S(5, 5, 5)$ in (59)). The results are compared with those obtained by the tree method. It can be seen that our method produces the exact numbers of CPs as the tree method.

VI. CONCLUSION

In this paper, we present a new method of evaluating the number of closed paths in a base matrix. Although we only give results up to length 10, results for longer paths can be readily derived and computed using similar principles. Compared with the traditional "tree method" which uses exhaustive searching, our method reveals the principle of closed paths and their duplicates and derives expressions for computing the number of closed paths. The results are useful when estimating the time resources required in optimizing and constructing LDPC codes.

REFERENCES

[1] Y. Wang, S. C. Draper, and J. S. Yedidia, "Hierarchical and high-girth QC LDPC codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4553–4583, Jul. 2013.
[2] D. G. M. Mitchell, R. Smarandache, and D. J. Costello, "Quasi-cyclic LDPC codes based on pre-lifted protographs," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2011, pp. 350–354.
[3] G. Zhang, "Type-II quasi-cyclic low-density parity-check codes from Sidon sequences," *Electron. Lett.*, vol. 52, no. 5, pp. 367–369, 2016. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/el.2015.2634
[4] M. Gholami and M. Alinia, "High-performance binary and non-binary low-density parity-check codes based on affine permutation matrices," *IET Commun.*, vol. 9, no. 17, pp. 2114–2123, Nov. 2015.
[5] C.-W. Sham, X. Chen, F. C. M. Lau, Y. Zhao, and W. M. Tam, "A 2.0 Gb/s throughput decoder for QC-LDPC convolutional codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 7, pp. 1857–1869, Jul. 2013.
[6] Q. Lu, J. Fan, C.-W. Sham, W. M. Tam, and F. C. M. Lau, "A 3.0 Gb/s throughput hardware-efficient decoder for cyclically-coupled QC-LDPC codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 1, pp. 134–145, Jan. 2016.
[7] X. Zheng, F. C. M. Lau, and T. K. Chi, "Constructing short-length irregular LDPC codes with low error floor," *IEEE Trans. Commun.*, vol. 58, no. 10, pp. 2823–2834, Oct. 2010.
[8] W. M. Tam, F. C. M. Lau, and C. K. Tse, "A class of QC-LDPC codes with low encoding complexity and good error performance," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 169–171, Feb. 2010.
[9] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, "A survey on protograph LDPC codes and their applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1989–2016, 4th Quart., 2015.
[10] Y. Wang, J. S. Yedidia, and S. C. Draper, "Construction of high-girth QC-LDPC codes," in *Proc. 5th Int. Symp. Turbo Codes Related Topics*, Sep. 2008, pp. 180–185.
[11] F. C. M. Lau and W. M. Tam, "A fast searching method for the construction of QC-LDPC codes with large girth," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2012, pp. 125–128.

SHENG JIANG received the B.Eng. degree in microelectronics from Shanghai Jiao Tong University, China, and the M.Eng. degree in electronic engineering from The Hong Kong University of Science and Technology, Hong Kong. He is currently pursuing the Ph.D. degree with The Hong Kong Polytechnic University, Hong Kong.

FRANCIS C. M. LAU (M'93–SM'03) received the B.Eng. degree (Hons.) in electrical and electronic engineering and the Ph.D. degree from the King's College London, University of London, U.K. He is currently a Professor and an Associate Head of the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. He is also a fellow of IET.

He is the co-author of *Chaos-Based Digital Communication Systems* (Heidelberg: Springer-Verlag, 2003) and *Digital Communications with Chaos: Multiple Access Techniques and Performance Evaluation* (Oxford: Elsevier, 2007). He is also a co-holder of five U.S. patents. He has published over 280 papers. His main research interests include channel coding, cooperative networks, wireless sensor networks, chaos-based digital communications, applications of complex-network theories, and wireless communications. He was the Chair of the Technical Committee on Nonlinear Circuits and Systems, IEEE Circuits and Systems Society, from 2012 to 2013. He served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II from 2004 to 2005, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I from 2006 to 2007, and the IEEE *Circuits and Systems Magazine* from 2012 to 2015. He has been a Guest Associate Editor of the *International Journal and Bifurcation and Chaos* since 2010 and an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II since 2016.