An Approach to Evaluating the Number of Potential Cycles in an All-one Base Matrix

Sheng Jiang and Francis C. M. Lau
Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong
Emails: sheng.jiang@connect.polyu.hk, encmlau@polyu.edu.hk

Abstract—**The "Tree Method" is usually used to identify potential cycles in low-density parity-check codes. However, with the increasing demand of high girth codes, the method becomes hard to implement because of the exponential increase of both space complexity and time complexity. In this paper, a new method is introduced to evaluate potential cycles for all-one base matrix. The method applies to large cycle length and arbitrary size base matrix. The principle of potential cycle and potential cycle duplication are studied to support the new approach. Instead of doing low efficient exhaustive search, the approach gives the number of potential cycles without duplication directly. The results of cycle numbers are given, which are verified by the "Tree Method".**

Index Terms—**base matrix, LDPC codes, potential cycles.**

I. Introduction

Much research have been conducted to achieve high performance low-density parity-check (LDPC) codes[1], [2], [3], [4], where girth—the minimum cycle length of Tanner graph, plays an important role. Researchers want to construct LDPC codes with girth as large as possible. Thus the study of potential cycles is required. In [5], the authors gives some short potential cycles for small base matrix protograph codes. 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 [1]. The cycle can also easily analyzed in matrices' manner since each check node in the Tanner graph corresponds to a row in its parity check matrix, and each variable node corresponds to a column. If a variable node is connected to a check node, the corresponding position on 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 alternatively moves horizontally and vertically along the "1"s in the matrix. The path must also be closed, that is to say, it starts and ends in the same "1" in the matrix. Obviously the length of cycles must be an even number since the horizontal and vertical moves always appear in pairs. It is also easy to find that the minimal cycle length possible is 4.

Let us consider a protograph LDPC codes with all-one base matrix. Protograph LDPC codes are constructed by replacing every "1" in the base matrix by a permutation matrix [6], [7]. For an $M \times N$ all-one base matrix, a permutation matrix denoted by $P_{i,j}$ is used to lift the slot (i, j). A typical base matrix can be written in the following form

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

For a potential cycle of length $2l$, it can always be described by a serial of permutation matrices: $P_{i_1,j_1} \rightarrow P_{i_2,j_1} \rightarrow P_{i_2,j_2} \rightarrow P_{i_3,j_2} \rightarrow \cdots \rightarrow P_{i_l,j_l} \rightarrow P_{i_1,j_l} \rightarrow P_{i_1,j_1}$. The cycle starts from permutation matrix 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 permutation matrix P_{i_1,j_1}.

The cycle is called "potential" because it does not necessarily mean that the cycle really exists in the codes. It depends on how we choose the permutation matrices used for lifting. Further searching and construction such as brute-force searching and hill climbing algorithm [8], [9] are required to achieve high-girth protograph codes based on the potential cycles. For convenience, we will use the abbreviation "PC" for potential cycle in the remaining part of the paper.

Researchers usually use the "Tree Method" to find PCs. The main idea of "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, and put the check nodes that connect to the root variable node in the next layer. After that, we add one more variable nodes 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 developed further enough, based on the cycle length requirement, researchers look into one certain variable node layer and search for the nodes that are the same as the root node. Every node that meets the criteria points to a PC.

The main drawback of the "Tree Method" is that the trees are always too large to manipulate. It works well for short PCs and small base matrices. However, it takes a huge amount of space to store the tree and it costs a lot of time to do the exhaustive searching for large base matrices. The searching results also contain duplications, which can only be eliminated by exhaustive comparisons.

To study and construct high girth protograph LDPC codes more efficiently and more practically, we develop a novel approach to evaluate PCs and give the cycle numbers directly. Furthermore, the principle of finding duplicated cycles is studied. The results are also verified with that given by the "Tree Method".

II. Proposed Method

A. Two preliminary functions

Suppose we are trying to find out all PCs of length $2l$, $l = 2, 3, 4...$ for an $M \times N$ base matrix. The following functions are helpful for calculating the number of PCs.

Theorem 1. *Assuming $u \geq v, u, v \in \mathbb{Z}^+$, we assign v different digits into u slots. Consecutive slots must contain different digits and all v digits should be used. Then the number of choices is*

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

where

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

Theorem 2. *Assuming $u \geq v, u, v \in \mathbb{Z}^+$, we assign v different digits into u slots. Consecutive slots must contain different digits and all v digits should be used. In addition, the last slot must contain different digit from the first one. Then the number of choices is*

$$G_{cyc}(u,v) = v! \sum_{\Omega} T(\alpha_v) \prod_{j=2}^{v} (j-1)^{\alpha_j} \tag{4}$$

where

$$\Omega = \{\alpha_j \in \mathbb{N}, \sum_{j=2}^{v} \alpha_j = u - v\}$$
$$T(\alpha_v) = \frac{(v-1)^{\alpha_v+1} + (-1)^{\alpha_v}}{v(v-1)^{\alpha_v}} \tag{5}$$

In channel coding theory, we usually consider short cycles with length no longer than 14 since girth larger than that is usually difficult to obtain. More specifically, for quasi-cyclic LDPC codes with all-one base matrix, the girth is bounded by 12. Since the number of PCs may correlate with those two functions, Table I and II show the values of $G(u,v)$ and $G_{cyc}(u,v)$ respectively for $v \leq u \leq l = 7$.

TABLE I
G FUNCTION TABLE

$G, u \setminus v$	1	2	3	4	5	6	7
1	1						
2	0	2					
3	0	2	6				
4	0	2	18	24			
5	0	2	42	144	120		
6	0	2	90	600	1200	720	
7	0	2	186	2160	7800	10800	5040

TABLE II
G_{cyc} FUNCTION TABLE

$G_{cyc}, u \setminus v$	1	2	3	4	5	6	7
1	1						
2	0	2					
3	0	0	6				
4	0	2	12	24			
5	0	0	30	120	120		
6	0	2	60	480	1080	720	
7	0	0	126	1680	6720	10080	5040

B. Calculation of PC configurations

PCs can be regarded as closed patterns with various shapes. We define a PC as (m,n)-PC if it consists of m distinct rows and n distinct columns. Suppose the base matrix is an all-one matrix with size $M \times N$. Once we have found an (m,n)-PC, we can easily find out all the other PCs in the matrix with the same shape. We simply pick m rows and n columns from the whole $M \times N$ base matrix. Every combination of m rows and n columns forms a PC of that same shape. Altogether we have $C_M^m C_N^n = \frac{M!N!}{(M-m)!(N-n)!m!n!}$ PCs in the base matrix with such a shape.

In order to find out all PCs of cycle length $2l$ in the given base matrix, we only need to find out (m,n)-PCs with all possible shapes for $m \leq l, n \leq l$. Those PCs are called PC configurations because all the other PCs can be found according to them.

Let us consider (m,n)-PC configurations with length $2l$. As we mentioned before, every combination of m rows and n columns picked from the $M \times N$ base matrix forms a PC of that same shape. So we simply extract all $m \times n$ "sub matrices" from the $M \times N$ base matrix. We only need to study one of them and all the others follow the same principles. The extraction and normalization of $m \times n$ "sub matrix" is shown in Figure 1. To avoid confusion, we use notation Q for the permutation matrices in the sub matrix instead of P, which is used in the base matrix.

For a PC: $Q_{i_1,j_1} \rightarrow Q_{i_2,j_1} \rightarrow Q_{i_2,j_2} \rightarrow Q_{i_3,j_2} \rightarrow ... \rightarrow Q_{i_l,j_l} \rightarrow Q_{i_1,j_l} \rightarrow Q_{i_1,j_1}$, the indexes of the permutation matrices in both rows and columns form two l-tuple vectors: $I = (i_1, i_2, ..., i_{l-1}, i_l)$ and $J = (j_1, j_2, ..., j_{l-1}, j_l)$. Since two different nodes form a path, the consecutive permutation matrices along the cycle must differ from each other. So $i_1 \neq i_2 \neq i_3 \neq ... \neq i_{l-1} \neq i_l \neq i_1$, as well as $j_1 \neq j_2 \neq j_3 \neq ... \neq j_{l-1} \neq j_l \neq j_1$.

Since the (m,n)-PC consists of m rows and n columns, there are exact m distinct values in row vector I and n distinct values in column vector J. On the other hand, all PC configurations can be constructed by assigning m different values to the row vector I and n different values to the column vector J.

For a given $M \times N$ base matrix and a given cycle length $2l$, the possible values of m and n are

$$2 \leq m \leq \min(M,l), \; 2 \leq n \leq \min(N,l). \tag{6}$$

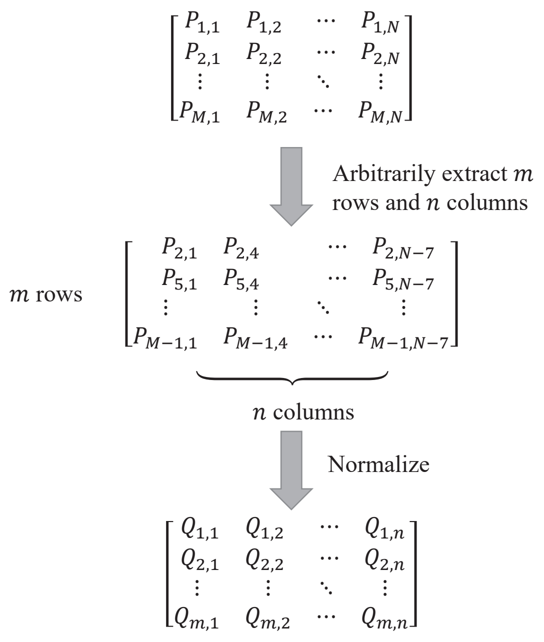

Fig. 1. The extraction and normalization of $m \times n$ "sub matrix".

For every m and n given by (6), the row and column combinations are calculated by using G_{cyc} function. Denote $R(l,m)$ as the number of the row combinations and $C(l,n)$ as the number of the column combinations for (m,n)-PC configuration with length $2l$. According to Theorem 2, we have

$$R(l,m) = G_{cyc}(l,m) \tag{7}$$

$$C(l,n) = G_{cyc}(l,n) \tag{8}$$

Without considering duplication, the number of (m,n)-PC configurations is $R(l,m)C(l,n) = G_{cyc}(l,m)G_{cyc}(l,n)$.

C. Duplication elimination

Now we have assigned values to vectors I and J, which are actually the indexes of permutation matrices that form PC configurations of length $2l$. However, duplication will occur when different (I,J) pairs lead to identical cycle. In such situation, we call one PC the duplication of another.

An example of PC duplication of cycle 6 is given in Figure 2, where there are 6 permutations matrices that can act as the starting permutation matrices. We will first eliminate those duplicated PCs that do not start from the first row.

In order to eliminate PC duplication that do not start from the first row (PCs start from $Q_{2,2}, Q_{3,3}, Q_{3,2}$ and $Q_{2,1}$ in Figure 2), we add a restriction that every PC must start from the first row of the $m \times n$ sub matrix. That is to say, we have $i_1 = 1$ for every PC configuration.

Because of the new restriction, the row combinations for (m,n)-PC with length $2l$ have to be re-calculated. Denote $R'(l,m,t)$ as the number of the row combinations under new restriction where t is the number of paths in the first row of the PC configuration. t also represents the number of ones in

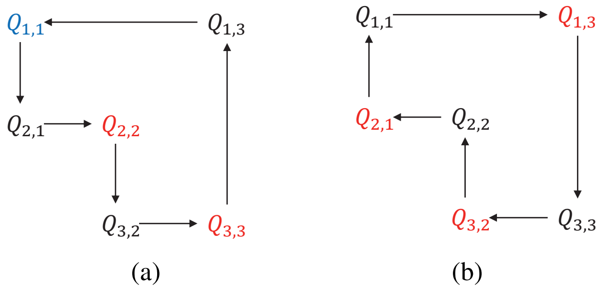

Fig. 2. Illustration of duplication for cycle 6. PCs start from $Q_{2,2}, Q_{3,3}, Q_{1,3}, Q_{3,2}$ and $Q_{2,1}$ are duplication of that starts from $Q_{1,1}$.

vector I. As consecutive elements in I must differ, there are at most $\lfloor \frac{l}{2} \rfloor$ ones in I. So we have $1 \le t \le \lfloor \frac{l}{2} \rfloor$. On the other hand, t is also restricted by $t \le l - m + 1$. To sum up, $t \le \min(\lfloor \frac{l}{2} \rfloor, l - m + 1)$. For the example in Figure 2, $t = 1$. An example of $t = 2$ is also shown in Figure 3. We will calculate $R'(l,m,t)$ for different t. In this paper we only consider PCs with length between 4 to 8, so $2 \le l \le 4$, $1 \le t \le 2$. For larger t and l, the principles are the same, and we can compute the results in the same manner.

Since $i_1 = 1$, we only need to assign values to $i_2, i_3, ..., i_l$. When $t = 1$, we assign all $m - 1$ digits to the $l - 1$ slots: $i_2, i_3, ..., i_l$ and make sure $i_2 \ne i_3 \ne i_4 \ne ... \ne i_l$. According to Theorem 1, we have

$$R'(l,m,1) = G(l-1, m-1) \tag{9}$$

When $t = 2$, one and only one element of $i_3, ..., i_{l-1}$ equals to 1. Assume $i_\xi = 1$, then we have vector $I = (1, i_2, i_3, ..., i_{\xi-1}, 1, i_{\xi+1}, ..., i_l)$.

If $i_{\xi-1} \ne i_{\xi+1}$ then it is equivalent to assign $m-1$ different digits to $i_2, i_3, ..., i_{\xi-1}, i_{\xi+1}, i_{\xi+2}, ..., i_l$ with consecutive elements different. According to Theorem 1, we have $G(l-2, m-1)$ combinations.

If $i_{\xi-1} = i_{\xi+1}$ then it is equivalent to assign $m-1$ different digits to $i_2, i_3, ..., i_{\xi-1}, i_{\xi+2}, i_{\xi+3}, ..., i_l$ with consecutive elements different. According to Theorem 1, we have $G(l-3, m-1)$ combinations. In such situation $m < l - 1$.

So we have

$$R'(l,m,2) = \begin{cases} 0 & \text{if } m > l-1 \\ G(l-2,m-1) & \text{if } m = l-1 \\ G(l-2,m-1) + G(l-3,m-1) & \text{if } m < l-1 \end{cases} \tag{10}$$

After the new constraint is applied, most of the duplications are eliminated. However, there are still some duplication existing since there may be more than one permutation matrix in the first row that can act as the starting point. For example, in Figure 2(b), the PC: $Q_{1,3} \to Q_{3,3} \to Q_{3,2} \to Q_{2,2} \to Q_{2,1} \to Q_{1,1}$ is a duplicated version of $Q_{1,1} \to Q_{2,1} \to Q_{2,2} \to Q_{3,2} \to Q_{3,3} \to Q_{1,3}$, which also starts from the first row. After further eliminating duplicated PCs that start from the first row, the total number of PCs without duplications equals

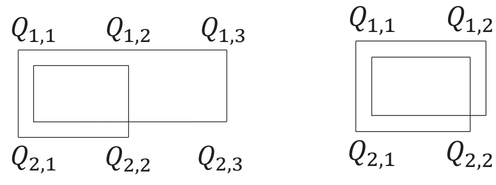

Fig. 3. Two length 8 PC configurations. (a) without Transform-Exact pair (b) with Transform-Exact pair.

- when $t = 1$,

$$D(l, m, n, t) = \frac{R'(l, m, 1)C(l, n)}{2}; \tag{11}$$

- when $t = 2$,

$$\begin{aligned} & D(l, m, n, t) \\ = \; & \frac{1}{4}\left(R'(l, m, 2)C(l, n) - G\left(\frac{l-2}{2}, m-1\right)G_{cyc}\left(\frac{l}{2}, n\right)\right) \\ & + \frac{1}{2}G\left(\frac{l-2}{2}, m-1\right)G_{cyc}\left(\frac{l}{2}, n\right). \end{aligned} \tag{12}$$

III. Results of Potential Cycle numbers

In this section, we will give the number of PCs with length from 4 to 8. For an $M \times N$ all-one base matrix and cycle length of $2l$, denote $S_{dup}(l, M, N)$ as the number of PCs with duplication, $S(l, M, N)$ as the number of PCs with duplication removed. Then we have

$$S_{dup}(l, M, N) = \sum_{m=2}^{\min(M,l)} \sum_{n=2}^{\min(N,l)} C(l, n)R(l, m)C_M^m C_N^n \tag{13}$$

and

$$S(l, M, N) = \sum_{m=2}^{\min(M,l)} \sum_{n=2}^{\min(N,l)} \sum_{t=1}^{\min(\lfloor \frac{l}{2} \rfloor, l-m+1)} D(l, m, n, t)C_M^m C_N^n. \tag{14}$$

A. Cycle 4

For cycle 4, $l = 2$, $m = n = 2$, $t = 1$. So for $N \geq M \geq 2$,

$$S_{dup}(2, M, N) = 4C_M^2 C_N^2 \tag{15}$$
$$S(2, M, N) = C_M^2 C_N^2 \tag{16}$$

B. Cycle 6

For cycle-6, $l = 3$, $2 \leq m, n \leq 3$, $t = 1$. So for $N \geq M \geq 3$,

$$S_{dup}(3, M, N) = 36C_M^3 C_N^3 \tag{17}$$
$$S(3, M, N) = 6C_M^3 C_N^3 \tag{18}$$

TABLE III
Evaluation of PC numbers

PC length	base matrix size $M \times N$	No. of PC obtained by proposed method with(without) duplication removed	No. of PC obtained by the "Tree Method"
4	2×3	3 (12)	3
6	3×4	24 (144)	24
8	4×5	2760 (21840)	2760

C. Cycle 8

For cycle-8, $l = 4$, $2 \leq m, n \leq 4$, $t = 1, 2$. So for $N \geq M \geq 4$,

$$\begin{aligned} S_{dup}(4, M, N) = & \; C_M^2(4C_N^2 + 24C_N^3 + 48C_N^4) \\ & + C_M^3(24C_N^2 + 144C_N^3 + 288C_N^4) \\ & + C_M^4(48C_N^2 + 288C_N^3 + 576C_N^4) \end{aligned} \tag{19}$$

$$\begin{aligned} S(4, M, N) = & \; C_M^2(C_N^2 + 3C_N^3 + 6C_N^4) \\ & + C_M^3(3C_N^2 + 18C_N^3 + 36C_N^4) \\ & + C_M^4(6C_N^2 + 36C_N^3 + 72C_N^4) \end{aligned} \tag{20}$$

The combination notation C_n^r are used in above equations. For $r \leq n$, $C_n^r = \frac{n!}{(n-r)!r!}$. We set $C_n^r = 0$ for $r > n$ which happens in the situation when the base matrix is not sufficient large to obtain all PCs. Our results are verified by those produced by the "Tree method". All results are shown in Table III.

IV. Conclusion

In this paper, we present a new method of evaluating potential cycles in an all-one base matrix. Although we only give results up to cycle of length 8, which corresponds to codes with girth 10, longer cycles share similar principles. Compared with traditional "Tree Method" which uses exhaustive searching, our method reveals the principle of cycles and cycle duplications and gives precise expressions for the number of cycles, which will be useful when constructing high-girth protograph LDPC codes.

Acknowledgment

The work described in this paper was supported by a grant from the RGC of the Hong Kong SAR, China (Project No. PolyU 152088/15E).

References

[1] Y. Wang, S. C. Draper, and J. S. Yedidia, "Hierarchical and high-girth qc ldpc codes," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4553–4583, July 2013.
[2] D. Mitchell, R. Smarandache, and D. Costello, "Quasi-cyclic ldpc codes based on pre-lifted protographs," in *Information Theory Workshop (ITW), 2011 IEEE*, Oct 2011, pp. 350–354.
[3] 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," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, no. 7, pp. 1857–1869, July 2013.

[4] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 1, pp. 134–145, Jan 2016.

[5] R. Smarandache, D. Mitchell, and J. Costello, D.J., “Partially quasi-cyclic protograph-based ldpc codes,” in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–5.

[6] 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,” *Communications Letters, IEEE*, vol. 14, no. 2, pp. 169–171, 2010.

[7] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, “A survey on protograph ldpc codes and their applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1989–2016, Fourthquarter 2015.

[8] Y. Wang and J. S. Yedidia, “Construction of high-girth qc-ldpc codes,” in *Turbo Codes and Related Topics, 2008 5th International Symposium on*, Sept 2008, pp. 180–185.

[9] F. C. M. Lau and W. M. Tam, “A fast searching method for the construction of qc-ldpc codes with large girth,” in *Computers and Communications (ISCC), 2012 IEEE Symposium on*, July 2012, pp. 125–128.

Sheng Jiang received the Bachelor of Engineering Degree in Microelectronics from Shanghai Jiaotong University, China, and the Master of Engineering Degree in Electronic Engineering from Hong Kong University of Science and Technology, Hong Kong. He is currently pursuing his PhD degree at The Hong Kong Polytechnic University, Hong Kong.

Francis C. M. Lau received the BEng (Hons) degree in electrical and electronic engineering and the PhD degree from King’s College London, University of London, UK. He is a Professor and Associate Head at the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. He is also a Fellow of IET and a Senior Member of IEEE.

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 US 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 Technical Committee on Nonlinear Circuits and Systems, IEEE Circuits and Systems Society in 2012-13. He served as an associate editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II in 2004-2005 and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I in 2006-2007, and IEEE CIRCUITS AND SYSTEMS MAGAZINE in 2012-2015. He has been a guest associate editor of INTERNATIONAL JOURNAL AND BIFURCATION AND CHAOS since 2010 and an associate editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II since 2016.