Construction of QC-LDPC Codes with Girth Larger Than Eight Based on GPU

Yejun He and Jie Yang
College of Information Engineering
Shenzhen University
Shenzhen, China
Email: heyejun@126.com, 17374140@qq.com, http://cie.szu.edu.cn/heyejun

***Abstract*—Based on the necessary and sufficient conditions for increasing girth, a shortened cycle elimination algorithm to construct Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) codes with large girth based on the Graphic Processing Unit (GPU) is presented. Firstly, we propose a cycle statistics algorithm based on GPU to search for the elements of base matrix that satisfies the no-cycle conditions. If the search process fails to find qualified elements, we delete the corresponding column in which this element is located in order to make the algorithm converge. And then, we apply GPU to construct QC-LDPC codes and get base matrices of QC-LDPC codes with girth larger than eight.**

Keywords- QC-LDPC; girth; Graphic Processing Unit (GPU); Compute Unified Device Architecture (CUDA).

I. Introduction

The construction methods of Low-Density Parity-Check (LDPC) codes are usually divided into two classes: random conformation and algebra structure. The performance of randomly constructed LDPC codes is better than that of algebra constructed LDPC codes, and the error floor is lower, but it is more difficult for implementation due to its higher coding and decoding complexity. The performance of algebra LDPC codes is not as good as randomly constructed LDPC codes while it is easy to implement. QC-LDPC code based on the cyclic shifted matrix is a specific kind of algebra LDPC code. QC-LDPC codes can be coded by shift register in linear time, at the same time, a few storage space are needed to store Tanner graph. The performance of QC-LDPC codes mainly depends on the girth of Tanner graph. Girth is defined as the number of edges involved in the shortest cycle in the Tanner graph. QC-LDPC codes can eliminate small trapping sets and stopping sets indirectly by increasing girth. Therefore, one way to improve the decoding performance of QC-LDPC is to eliminate short cycle. In [1], Fossorier pointed out the necessary and sufficient conditions to avoid short cycle, and proposed a computer search method that can generate QC-LDPC codes with girth at least 6. But the computer search method is time-consuming and not so efficient. Existing construction algorithms of large girth QC-LDPC codes are as follows: allow slope [2] and secondary replacement polynomial [3], balance loop [4], control equations [5], grid [6], three dimensional circulation grid [7], adjacency matrix theory [8] and hill climbing algorithm [9]. Hill climbing algorithm firstly randomly generates a base matrix, and then gradually optimizes the base matrix to meet the design requirements. But the existing algorithm has the possibility of failure.

The rest of this paper is organized as follows: Section II briefly introduces the basic theory of QC-LDPC codes with large girth, including the necessary and sufficient conditions to eliminate short cycles. A new algorithm to search QC-LDPC codes with large girth by using GPU platform is proposed in Section III. Section IV is the simulation results. Section V is the conclusions.

II. QC-LDPC codes

The parity check matrix of a (J, L) regular QC-LDPC code with code length $N = pL$ can be written as

$$H = \begin{bmatrix} I(0) & I(0) & ... & I(0) \\ I(0) & I(p_{1,1}) & ... & I(p_{1,L-1}) \\ \vdots & \vdots & \ddots & \vdots \\ I(0) & I(p_{J-1,1}) & ... & I(p_{J-1,L-1}) \end{bmatrix} \quad (1)$$

where J represents each column of H involved in J $1's$ and L represents each row of H involved in L $1's$, $1 \leq j \leq J-1$, $1 \leq l \leq L-1$. $I(p_{j,l})$ is the $p \times p$ circulation permutation matrix obtained by cyclically right shifted the $p \times p$ identity matrix $I(0)$ by $p_{j,l}$ positions. For a given QC-LDPC code, we define the corresponding base matrix as the cyclic shift times:

$$B = \begin{bmatrix} 0 & 0 & ... & 0 \\ 0 & p_{1,1} & ... & p_{1,L-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & p_{J-1,1} & ... & p_{J-1,L-1} \end{bmatrix} \quad (2)$$

The cycle of length $2i$ in the parity check matrix $H = [h_{x,y}]$ is decided by $2i$ positions $h_{x,y} = 1$, and these $2i$ positions $h_{x,y} = 1$ should meet the following requirements: (a) two consecutive positions are obtained by changing alternatively of row or column only; (b) except the first one and last one, all positions are not all the same. Thus, two consecutive positions in any cycle belong to different cyclic permutation matrices which are either in the same row or in the same column. Therefore, the cycle of length $2i$ can be described as a series of circulation permutation matrix.

$$I(p_{j_0,l_0}),I(p_{j_1,l_0}),I(p_{j_1,l_1}),\ldots,I(p_{j_i-1,l_i-1}),I(p_{j_0,l_i-1}),I(p_{j_0,l_0}) \tag{3}$$

where $1 \leq k \leq i$, $j_k \neq j_k - 1$ and $l_k \neq l_k - 1$. Equation (3) can be expressed as the cyclic shift number:

$$(p_{j_0,l_0}),(p_{j_1,l_0}),(p_{j_1,l_1}),\ldots,(p_{j_i-1,l_i-1}),(p_{j_0,l_i-1}),(p_{j_0,l_0}) \tag{4}$$

The sufficient and necessary conditions for QC-LDPC codes to have a girth at least $2i+1$ [1] is:

$$\sum_{k=0}^{m-1} \Delta_{j_k,j_k+1}(l_k) \neq 0 \qquad mod \quad p \tag{5}$$

where $\Delta_{j_x,j_y}(l) = p_{j_x,l} - p_{j_y,l}$, $2 \leq m \leq i$, $0 \leq j_k \leq J-1$, $0 \leq j_{k+1} \leq J-1$, $0 \leq l_k \leq L-1$ and $j_0 = j_m$, $j_k \neq j_{k+1}$, $l_k \neq l_{k+1}$.

We know a method to design QC-LDPC codes with large girth from Equation (5). So the construction of QC-LDPC codes can be simplified as that of the base matrix.

Graphics Processing Units (GPUs) are traditionally applied to image Processing, but with the large advancement in the parallel processing performance of GPU and the rapid development of the programmable ability in recent years, the use of GPU to accelerate algorithm has become a hot topic. CUDA (Compute Unified Device Architecture) is a general parallel computing architecture launched by NVIDIA. This structure created the programming model and storage type of GPU, which can apply parallel computing of GPU to solve complex computing problems. This structure contains a CUDA Instruction Set framework (ISA) and GPU internal parallel computing engine. Developers can use C language for CUDA architecture program which runs on the processor supported CUDA with high performance. CUDA has been used in molecular dynamics, bioinformatics, earth physics.

GPU consists of a series of Stream Multiprocessor (SM) as shown in Fig. 1. Each SM contains Stream Processors (SP), registers, constant memory and texture memory. In the CUDA programming model and CPU is the host, GPU is the equipment. CPU and GPU work collaboratively, as shown in Fig. 2. Hollow lines with arrow a, b, c and d denote data bus. Thin lines with arrow e and f denote controlling bus. CPU is responsible for the logical affairs and serial calculations, while GPU is responsible for highly threaded parallel computing tasks. Both CPU and GPU have independent memory address space. The parallel computation function running on the GPU is called kernel functions. A complete CUDA program consists of a series of kernel functions on CUDA and serial program on hosts. Kernel is composed of thread grid; each thread grid is made up of multiple thread blocks; and every block is made up of multiple threads.

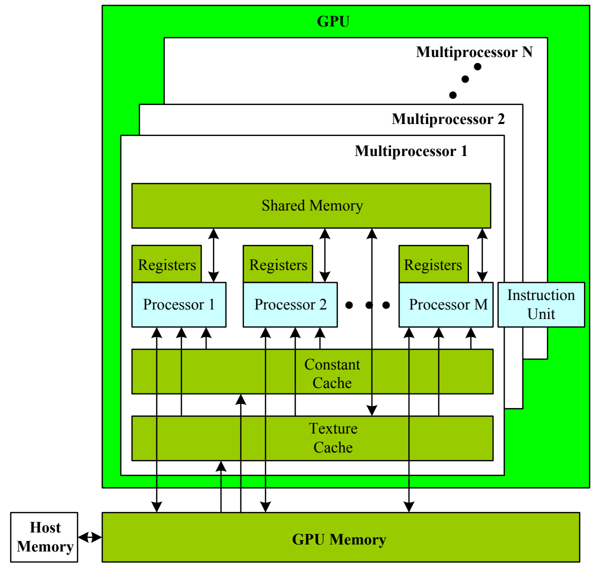


Fig. 1. Structure of GPU

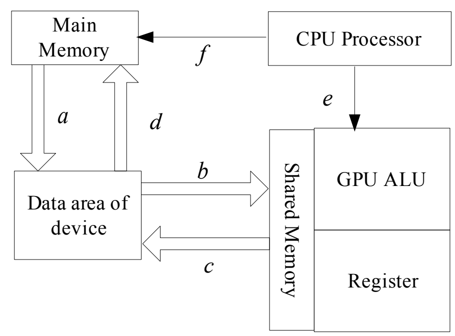


Fig. 2. Program execution flow of CUDA

III. A SHORTENED CYCLE ELIMINATION ALGORITHM

The key to design base matrix is to choose the elements in the base matrix to eliminate short cycle. There are two kinds of methods to construct QC-LDPC codes without short cycles. One method is random search, and the other is structured construction. Random search method randomly generates base matrix, then checks whether the base matrix meets the girth conditions. The efficiency of this method is not good. Structured construction generates the base matrix according to certain rules in order to make the base matrix meet girth condition, such as array LDPC codes. But the parameters of QC-LDPC codes that generated by structured construction are not flexible, and it is difficult to generate QC-LDPC codes with girth larger than 8.

Based on the necessary and sufficient conditions of increasing girth, we put forward a shortened cycle elimination algorithm by using the statistical algorithm, which searches for qualified elements one by one. If the searching process fails to find qualified elements, then we delete the whole column corresponding to this element in order to make the algorithm converge. The simulation results show that the successful probability of constructing QC-LDPC codes is significantly higher than the hill climbing algorithms, and its time complexity is much lower.

A. A cycle statistics algorithm

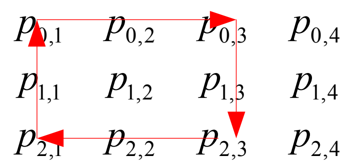
Calculate all the possible row number combinations that may form a cycle. For example, for base matrix with $J = 3$, all possible row number combinations that may form a 4-cycle are $(0, 1)$, $(0, 2)$, $(1, 2)$, and all possible row number combinations that may form a 6-cycle are $(0, 1, 2)$, all possible row number combinations that may form a 8-cycle are $(0, 1, 0, 1)$, $(0, 1, 0, 2)$, $(0, 2, 0, 2)$, $(0, 1, 2, 1)$, $(0, 2, 1, 2)$, $(1, 2, 1, 2)$.

Various possible shapes for 4-cycle, 6-cycle, 8-cycle are shown in Fig. 3. The line segment with arrow denotes an edge of each cycle. For example, a 4-cycle form between $(0, 2)$ in Fig. $3(a)$, where 0 and 2 denote the zeroth row and the second row, respectively. In Fig. $3(b)$ we give two 6-cycle examples using $(0, 1, 2)$, where 0, 1 and 2 denote the zeroth row, the first row and the second row, respectively. In Fig. $3(c)$ we give an example of six 8-cycle using $(0, 1, 0, 2)$, $(0, 2, 0, 1)$, $(0, 1, 2, 1)$, $(0, 1, 0, 1)$, $(0, 1, 2, 0)$, $(0, 2, 0, 2)$, where 0, 1 and 2 denote the zeroth row, the first row and the second row, respectively.

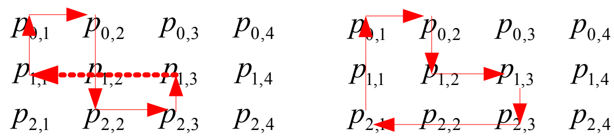
Assign a block and certain number of thread for each row combination. For example, the number of threads assigned for 4-cycle, 6-cycle, 8-cycle are L, $L \times L$, $L \times L \times L$, respectively.

Each thread processes one combination and records the corresponding row numbers and column number that may form a cycle. For example, when calculating the number of 4-cycle of the elements in base matrix, each thread processes one column of the row combination, and finds out other columns which form a 4 cycle with the current processed by thread, then records the column number that forms 4-cycle with each other. When calculating the number of 6-cycle of the elements in base matrix, each thread processes two columns of the row combination, and finds out other columns which form a 6 cycle with the two columns processed by thread, then records the column number that forms a 6-cycle. When calculating the number of 8-cycle of the elements in base matrix, each thread processes two column of the row combination, and finds out all other two columns which form a 8-cycle with the columns processed by thread, then records the column number that forms 8-cycle. Combined with row combination and corresponding column number, it is easy to calculate the number of cycle of each element in the base matrix.

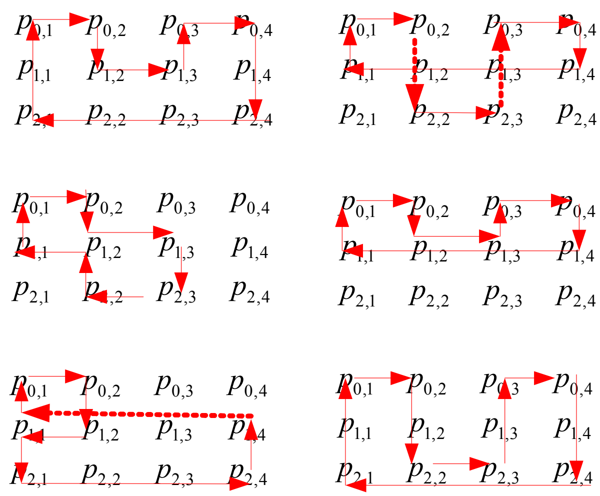
In Fig. $4(a)$, there are three row combinations for $J = 3$, so we assign three blocks. Each block processes one combination, and each block contains $L + 1$ threads. The number of thread is corresponding to the number of column. For example, in Fig. $4(a)$, the number 1 block processes combination $(0, 2)$, then the number 1 thread processes the first column of the



(a) 4-cycle



(b) 6-cycle



(c) 8-cycle

Fig. 3. Possible shapes for 4-cycle, 6-cycle, 8-cycle

matrix. There are two elements on the number 1 column on the number 0 and number 2 row in the base matrix, supposed to be A and B, then we consider the elements on the other column on the number 0 and number 2 row respectively, supposed to be C and D. If the value of $A - C + D - B$ equals to the multiple of p, we record the corresponding line number and column number. For 6-cycle as shown in Fig. $4(b)$, there is only one row combinations, therefore we only assign one block, the number of thread is corresponding to the column number processed by this thread. For example, thread $(0, 1)$ processes the number 0 and number 1 column of the matrix. Element A, B, C, D are in a 6-cycle, we only need to consider the elements on the other column on the number 1 and number 2 row respectively, supposed to be E and F, then if the value of $A - C + D - E + F - B$ equals to the multiple of p, record the corresponding row number and column number. The process of 8-cycle is similar to above processes, elements A, B, C, D,

E, F, I form 8-cycle in Fig. $4(c)$.

The result of above algorithm is a $J \times L$ matrix N^i, N^i represents the number of i-cycle of each element in base matrix.

$$N^i = \begin{bmatrix} n^i_{0,0} & n^i_{0,1} & \cdots & n^i_{0,L-1} \\ n^i_{1,0} & n^i_{1,1} & \cdots & n^i_{1,L-1} \\ \vdots & \vdots & \ddots & \vdots \\ n^i_{J-1,0} & n^i_{J-1,1} & \cdots & n^i_{J-1,L-1} \end{bmatrix} \quad (6)$$

where $n^i_{j,l}$ is the number of i-cycle of element $p_{j,l}$.

B. A new shortened cycle elimination algorithm

We put forward a shortened cycle elimination algorithm based on the path statistical algorithms and the idea of array code. Shortened cycle elimination algorithm is a local optimal algorithm. According to the path cost matrix, we optimize an element each time to maximize the number of the path of potential cycle. Finally, we delete the column that can not satisfy the condition, so that we can construct the QC-LDPC code with specified girth successfully.

Path cost matrix is defined as follows:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,L-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,L-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{J-1,0} & a_{J-1,1} & \cdots & a_{J-1,L-1} \end{bmatrix} \quad (7)$$

where $a_{j,l} = [a_{j,l,0}, a_{j,l,1}, ..., a_{j,l,p-1}]$, $a_{j,l,z}$ is the path number of the new element z $(0 \le z \le p-1)$ changed from $p_{i,j}$. Cost matrix describes the path number of the new element changed from another element.

$$a_{j,l,z} = \sum_{i=2}^{g/2-l} q^{(i)}_{j,l,z} \quad (8)$$

where $q^{(i)}_{j,l,z}$ is the number of path with cycle length of $2i$ of z $(0 \le z \le p-1)$ changed from $p_{i,j}$ in B, g is the girth of QC-LDPC code.

We calculate cost matrix A based on the cycle statistics algorithm. For every element $p_{j,l}$ in the base matrix, we replace $p_{j,l}$ with z $(0 \le z \le p-1)$, then

$$q^{(i)}_{j,l,z} = n^i_{j,l} \quad (9)$$

The total number of the path with cycle length contained i in the based matrix is:

$$n^i = \sum_{0 \le j \le J-1} \sum_{0 \le l \le L-1} n^i_{j,l}/i \quad (10)$$

Flow chart of a new shortened cycle elimination algorithm is shown in Fig. 5. And the pseudo code of this algorithm is as follows:

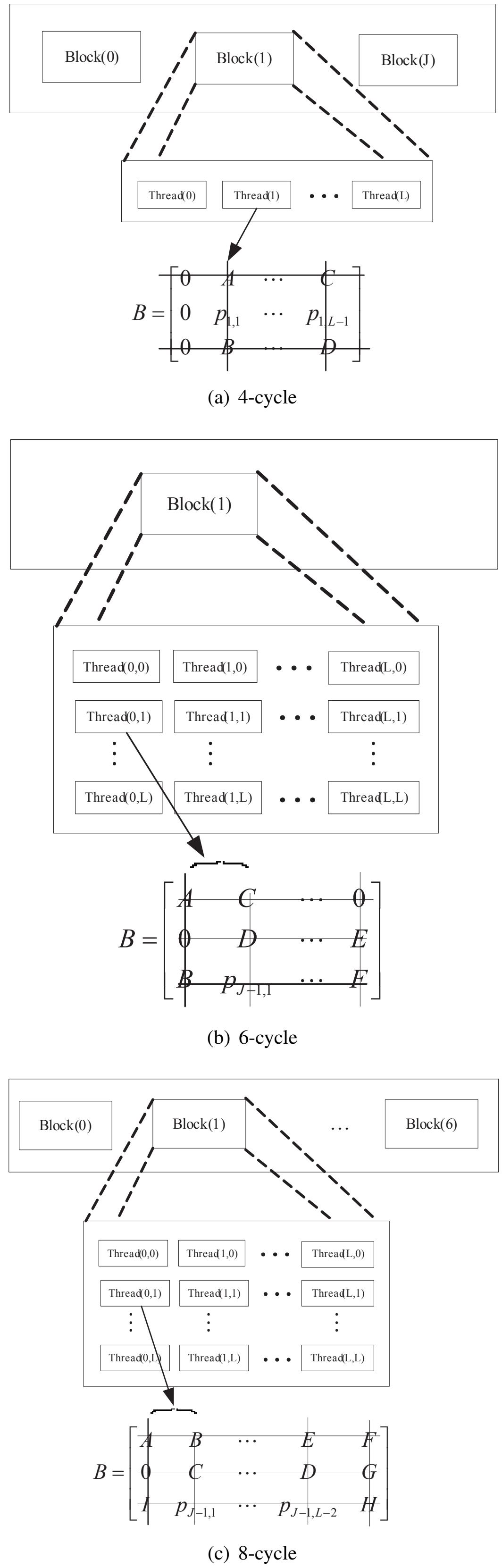


Fig. 4. Processing for cycles with different length

Step 1: according to given code parameters J, L, p, randomly generate base matrix B, and the elements of B are different from each other.

Step 2: use A cycle statistics algorithm to get $N_i(4 \le i \le g-2)$.

Step 3: calculate the number of cycles whose length is less than girth g,

$$N = \sum_{4 \le i \le g-2} N^i \tag{11}$$

Step 4: set $n_{max} = \max_{(j,l):0\le j\le J-1, 0\le l\le L-1} n_{j,l}$

$$(j_{max}, l_{max}) = \arg\max_{(j,l):0\le j\le J-1, 0\le l\le L-1} n_{j,l} \tag{12}$$

For $0 \le z \le p-1$

set $p_{j_{max},l_{max}} = z$, repeat step 2 and step 3 to get N'.

If $N'_{j_{max},l_{max}} = 0$

get to step 2

End

End

If $N'_{j_{max},l_{max}} \ne 0$

Delete the column with number l_{max} in B,

go to step 2

End

For base matrix B after elimination of μ columns, when $0 \le j \le J-1$, $0 \le l \le L-\mu$, $n_{j,l} = 0$, then exit successfully.

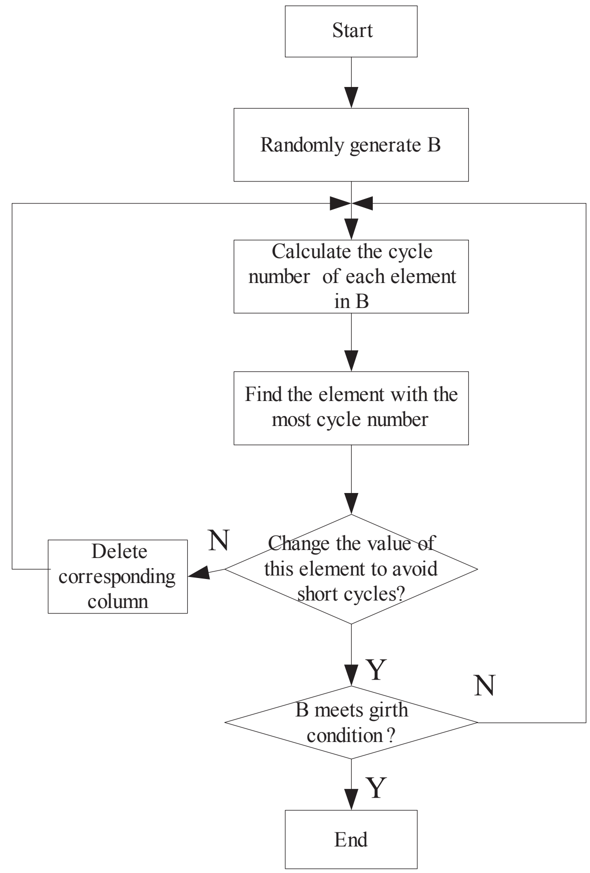


Fig. 5. Flow chart of a new shortened cycle elimination

IV. Simulation results

We adopt the high performance GPU chip-GTX460 of NVIDIA, and integrated CUDA3.2 edition runtime library into the Visual Studio 2008 platform so as to form a mixed compiling environment.

For example, the parameters of QC-LDPC code are $J = 3$, $L = 10$, $p = 765$. Therefore, the code length $N = pL = 765 \times 10 = 7650$, and the code rate is $7/10$.

The base matrix of QC-LDPC code with a girth of 6 is obtained as follows:

$$B_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 119 & 58 & 64 & 185 & 600 & 210 & 149 & 257 & 76 \\ 0 & 190 & 760 & 162 & 61 & 417 & 468 & 698 & 394 & 522 \end{bmatrix} \tag{13}$$

The number of 6-cycle of each element in B_6 is obtained as follows:

$$B_6{}' = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{14}$$

The base matrix of QC-LDPC code with a girth of 8 is obtained as follows:

$$B_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 8 & 9 & 10 \\ 0 & 7 & 13 & 18 & 26 & 33 & 40 & 49 & 59 & 62 \end{bmatrix} \tag{15}$$

The number of 6-cycle of each element in B_8 is obtained as follows:

$$B_8{}' = \begin{bmatrix} 214 & 250 & 260 & 254 & 282 & 280 & 270 & 228 & 226 & 200 \\ 218 & 258 & 256 & 278 & 296 & 298 & 268 & 266 & 234 & 196 \\ 112 & 140 & 116 & 112 & 150 & 158 & 122 & 110 & 104 & 84 \end{bmatrix} \tag{16}$$

The base matrix of QC-LDPC code with a girth of 10 is obtained as follows:

$$B_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 73 & 130 & 183 & 228 & 289 & 328 & 408 & 468 & 433 \\ 0 & 286 & 125 & 114 & 271 & 33 & 40 & 49 & 59 & 62 \end{bmatrix} \tag{17}$$

In [1], Fossorier used random way to get the smallest value of p (p_{min}) when the girth of QC-LDPC code is 8 and $J = 3$. Hill climbing algorithm improved the upper bound of p_{min}. Our shortened cycle elimination algorithm can get better upper

TABLE I
THE UPPER BOUND OF p_{min} OF THREE ALGORITHMS

L	4	5	6	7	8	9	10	11	12
Random	9	14	18	21	26	33	39	46	54
Hill climbing	9	13	18	21	25	33	39	41	54
Shortened cycle elimination	9	13	18	21	25	32	38	40	52

bound of p_{min} as shown in TABLE I.

V. CONCLUSIONS

In this paper we proposed a new algorithm to construct QC-LDPC codes with large girth, and the core of this algorithm is a cycle statistics algorithm. The cycle statistical algorithm can be used to calculate the number of cycles of each element in the base matrix. Shorten cycle elimination algorithm changes the value of the element in the base matrix one by one in order to eliminate short cycles. If the search process fails to find qualified elements, we delete the corresponding column in which this element is located in order to make the algorithm converge. The proposed algorithm is applicable to the cases with the settings for various parameters. In addition, the method in this paper can also be extended to produce girth-12 QC-LDPC codes.

ACKNOWLEDGEMENT

This paper is supported by National Natural Science Foundation of China (No. 60972037), the Fundamental Research Program of Shenzhen City (No. JC200903120101A and No. JC201005250067A) and International Cooperative Program of Shenzhen City (No. ZYA201106090040A).

REFERENCES

[1] M. P. C. Fossorier, “Quasi-cyclic low-density parity-check codes from circulant permutation matrices,” *IEEE Transactions on Information Theory,* vol. 50, no.8, pp. 1788-1793, Aug. 2004.

[2] Haotian Zhang,J. M. F. Moura, “Geometry based designs of LDPC codes,” *2004 IEEE International Conference on Communications,* vol. 2, pp. 762-766, June 2004.

[3] O. Y. Takeshita, “A Compact Construction for LDPC Codes using Permutation Polynomials,” *2006 IEEE International Symposium on Information Theory,* pp. 79-82, July 2006.

[4] M. E. O’Sullivan, “Algebraic construction of sparse matrices with large girth,” *IEEE Transactions on Information Theory,* vol. 52, no. 2, pp. 718-727, Feb. 2006.

[5] O. Milenkovic, N. Kashyap, D. Leyba, “Shortened Array Codes of Large Girth,” *IEEE Transactions on Information Theory,* vol. 52 , no. 8, pp. 3707-3722, Aug. 2006.

[6] Xiongfei Tao, Jong-man Kim, Weizhong Liu, Li Kong, “Improved Construction of Low-Density Parity-Check Codes Based on Lattices,” *2007 International Symposium on Information Technology Convergence,* pp. 208-212, Nov. 2007.

[7] Fan Zhang, Xuehong Mao, Wuyang Zhou, H. D. Pfister, “Girth-10 LDPC Codes Based on 3-D Cyclic Lattices,” *IEEE Transactions on Vehicular Technology,* vol. 57, no. 2, pp. 1049-1060, Mar. 2008.

[8] Xiaofu Wu, Xiaohu You, Chunming Zhao, “A necessary and sufficient condition for determining the girth of quasi-cyclic LDPC codes,” *IEEE Transactions on Communications,* vol. 56, no. 6, pp. 854-857, June 2008.

[9] Yige Wang, J. S. Yedidia, S. C. Draper, “Construction of high-girth QC-LDPC codes,” *2008 5th International Symposium on Turbo Codes and Related Topics,* pp. 180-185, Nov. 2008.