SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Nonbinary LDPC cycle codes: efficient search, design, and code optimization

Hengzhou XU[1,3*], Chao CHEN[2], Min ZHU[3], Baoming BAI[3*] & Bo ZHANG[1]

[1]*School of Network Engineering, Zhoukou Normal University, Zhoukou* 466001*, China;*
[2]*School of Electronic Engineering, Xidian University, Xi'an* 710071*, China;*
[3]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an* 710071*, China*

Appendix A Isomorphism theory

Theorem 1. Let $\mathcal{G}_1(V,C)$ and $\mathcal{G}_2(V,C)$ be two bipartite graphs with the same disjoint vertex sets V and C, and corresponding biadjacency matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively. If $\mathbf{A}_2$ is obtained by permutating the row (column) indices of $\mathbf{A}_1$ under a permutation, then $\mathcal{G}_1(V,C)$ and $\mathcal{G}_2(V,C)$ are isomorphic.

Proof. Consider the permutation π of row indices of $\mathbf{A}_1$ (the permutation of column indices can be also proved in the same way). Suppose that we label the vertices of $\mathcal{G}_1(V,C), v_1, v_2, ..., v_{|V|}, c_1, c_2, ..., c_{|C|}$. That is, the ordered vertices of $\mathcal{G}_2(V,C)$ are $v_1, v_2, ..., v_{|V|}, \pi(c_1), \pi(c_2), ..., \pi(c_{|C|})$. Define the bijection f mapping the vertex sets V and C of $\mathcal{G}_1(V,C)$ into the vertex sets V and C of $\mathcal{G}_2(V,C)$ as follows: $f(v)=v$ if $v\in V$, and $f(c)=\pi(c)$ if $c\in C$. It can be observed that the existence of the edge of $\mathcal{G}_1(V,C)$ between the vertex v and the vertex c is equivalent to that of $\mathcal{G}_2(V,C)$ between the vertex v and the vertex $\pi(c)$. Then $\mathcal{G}_1(V,C)$ and $\mathcal{G}_2(V,C)$ are isomorphic based on the bijection f. This completes the proof.

Theorem 2. Let L be a given positive integer and $\mathbf{I}(p)$ be a CPM of size $L\times L$ for $0\leqslant p\leqslant L-1$. Under a specified permutation π of the row (or column) indices of $\mathbf{I}(p)$, $\mathbf{I}(p)$ becomes the identity matrix $\mathbf{I}(0)$ (written as $\pi(\mathbf{I}(p))=\mathbf{I}(0)$).

Proof. Without loss of generality, we only consider the permutation π of row indices of $\mathbf{I}(p)$. It can be easily checked that the row-i of $I(0)$ is the row-r of $\mathbf{I}(p)$ where $r+p-i=0$ (mod L) for $0\leqslant i\leqslant L-1$. This corresponds to the following permutation π of row indices of $\mathbf{I}(p)$: $\mathbf{I}(p)\to\mathbf{I}(0)$ such that $\pi(L-p)=0, \pi(L-p+1)=1, ..., \pi(L-1)=p-1, \pi(0)=p, ...,$ and $\pi(L-p-1)=L-1$, denoted by

$$\pi=\begin{pmatrix} L-p & L-p+1 & \cdots & L-1 & 0 & 1 & \cdots & L-p-1 \\ 0 & 1 & \cdots & p-1 & p & p+1 & \cdots & L-1 \end{pmatrix}.$$

So $\pi(\mathbf{I}(p))=\mathbf{I}(0)$. This completes the proof.

For the given $L\times L$ CPMs $\mathbf{I}(p)$ and $\mathbf{I}(0)$, if there is a permutation π of row (or column) indices of $\mathbf{I}(p)$ such that $\pi(\mathbf{I}(p))=\mathbf{I}(0)$, there also exists an inverse of π, denoted by π^{-1}, such that $\pi^{-1}(\mathbf{I}(0))=\mathbf{I}(p)$. The permutation π and its inverse π^{-1} are called elementary permutations. If two permutations π_1 and π_2 of row (or column) indices are applied to the CPM $\mathbf{I}(p)$ successively, we denote the composition of permutations as $\pi_2(\pi_1(\mathbf{I}(p)))$. Obviously, $\pi^{-1}(\pi(\mathbf{I}(p)))=\mathbf{I}(p)$. In the above theorem, we can find that the CPM $\mathbf{I}(0)$ can also be obtained by taking the row-$(L-p)$ of $\mathbf{I}(p)$ as the first row and its $L-1$ right cyclic-shifts as the other $L-1$ rows. Let π_1 and π_2 be two elementary permutations such that $\pi_1(\mathbf{I}(p_1))=\mathbf{I}(0)$ and $\pi_2(\mathbf{I}(p_2))=\mathbf{I}(0)$, then $\pi_2^{-1}(\pi_1(\mathbf{I}(p_1)))=\mathbf{I}(p_2)$. That is, the composition of any two elementary permutations, which is applied to a CPM $\mathbf{I}(p)$, can result in another CPM $\mathbf{I}(p')$.

Theorem 3. Let

$$\mathbf{H}_b=\begin{bmatrix} \mathbf{I}(p_{0,0}) & \mathbf{I}(p_{0,1}) & \cdots & \mathbf{I}(p_{0,\rho-1}) \\ \mathbf{I}(p_{1,0}) & \mathbf{I}(p_{1,1}) & \cdots & \mathbf{I}(p_{1,\rho-1}) \end{bmatrix},$$

where, for $0\leqslant i\leqslant 1$ and $0\leqslant j\leqslant \rho-1$, $\mathbf{I}(p_{i,j})$ is an $L\times L$ CPM. Based on some permutations (or permutation composition) of row and column indices of CPMs, the matrix $\mathbf{H}_b$ can be transformed into the following form

$$\mathbf{H}=\begin{bmatrix} \mathbf{I}(0) & \mathbf{I}(0) & \cdots & \mathbf{I}(0) \\ \mathbf{I}(p_0)(=\mathbf{I}(0)) & \mathbf{I}(p_1) & \cdots & \mathbf{I}(p_{\rho-1}) \end{bmatrix}, \tag{A1}$$

[*] Corresponding author (email: hzxu@zknu.edu.cn, bmbai@mail.xidian.edu.cn)

where $0 \leqslant p_i \leqslant L-1$ for $1 \leqslant i \leqslant \rho-1$.

Proof. By Theorem 2, there exist ρ permutations $\pi_0, \pi_1, ..., \pi_{\rho-1}$ of column indices of CPMs $\mathbf{I}(p_{0,0}), \mathbf{I}(p_{0,1}), \ldots, \mathbf{I}(p_{0,\rho-1})$ such that $\pi_0(\mathbf{I}(p_{0,0})) = \mathbf{I}(0), \pi_1(\mathbf{I}(p_{0,1})) = \mathbf{I}(0), ..., \pi_{\rho-1}(\mathbf{I}(p_{0,\rho-1})) = \mathbf{I}(0)$. For $0 \leqslant i \leqslant \rho-1$, we apply the permutations π_i to CPMs $\mathbf{I}(p_{0,i})$ of $\mathbf{H}_b$, and obtain the following matrix

$$\mathbf{H}'_b = \begin{bmatrix} \mathbf{I}(0) & \mathbf{I}(0) & \cdots & \mathbf{I}(0) \\ \pi_0(\mathbf{I}(p_{1,0})) & \pi_1(\mathbf{I}(p_{1,1})) & \cdots & \pi_{\rho-1}(\mathbf{I}(p_{1,\rho-1})) \end{bmatrix}.$$

There also exists a permutation π of row indices of $\mathbf{I}(p_{1,0})$ such that $\pi(\mathbf{I}(p_{1,0})) = \mathbf{I}(0)$. By applying the permutation composition $\pi\pi_0^{-1}$ to the CPMs in the second row of $\mathbf{H}'_b$, then the CPM $\pi_0(\mathbf{I}(p_{1,0}))$ can be modified as

$$\pi(\pi_0^{-1}(\pi_0(\mathbf{I}(p_{1,0})))) = \pi(\mathbf{I}(p_{1,0})) = \mathbf{I}(0).$$

It can be seen that for $1 \leqslant i \leqslant \rho-1$, the permutation composition $\pi\pi_0^{-1}$ transforms CPMs $\pi_i(\mathbf{I}(p_{1,i}))$ into another CPMs $\mathbf{I}(p_i)$, i.e., $\pi(\pi_0^{-1}(\pi_i(\mathbf{I}(p_{1,i})))) = \mathbf{I}(p_i)$. Then, the matrix $\mathbf{H}'_b$ becomes

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}(0) & \mathbf{I}(0) & \cdots & \mathbf{I}(0) \\ \mathbf{I}(p_0) & \mathbf{I}(p_1) & \cdots & \mathbf{I}(p_{\rho-1}) \end{bmatrix},$$

where $\mathbf{I}(p_0) = \mathbf{I}(0)$ and $\mathbf{I}(p_i) = \pi(\pi_0^{-1}(\pi_i(\mathbf{I}(p_{1,i}))))$ for $1 \leqslant i \leqslant \rho-1$. This completes the proof.

Theorem 4. Let $\mathbf{H}_{\mathbf{P}_1}$ and $\mathbf{H}_{\mathbf{P}_2}$ be the parity-check matrices of two NBLDPC cycle codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with the same length ρL, and corresponding exponent matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ of size $2 \times \rho$, respectively. Consider

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ p_0(=0) & p_1 & p_2 & \cdots & p_{\rho-1} \end{bmatrix}.$$

The code $\mathcal{C}_1$ is isomorphic to the code $\mathcal{C}_2$, i.e., $\mathcal{C}_1 \cong \mathcal{C}_2$, or equivalently $\mathbf{H}_{\mathbf{P}_1} \cong \mathbf{H}_{\mathbf{P}_2}$ ($\mathbf{P}_1 \cong \mathbf{P}_2$) if $\mathbf{P}_2$ is obtained from one of the following methods.

1. For any permutation π on the set $\{0, 1, \cdots, \rho-1\}$.

$$\mathbf{P}_2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ p_{\pi(0)} & p_{\pi(1)} & p_{\pi(2)} & \cdots & p_{\pi(\rho-1)} \end{bmatrix}. \tag{A2}$$

2. For any $r \in \{0, 1, ..., \rho-1\}$,

$$\mathbf{P}_2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ p'_{j_0} & p'_{j_1} & p'_{j_2} & \cdots & p'_{j_{\rho-1}} \end{bmatrix} \tag{A3}$$

with $p'_{j_i} = (p_{j_i} - p_{j_r}) \pmod L$.

3. For r and L being relatively prime, i.e., $(r, L) = 1$ and $0 \leqslant r \leqslant L-1$,

$$\mathbf{P}_2 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ p'_{j_0} & p'_{j_1} & p'_{j_2} & \cdots & p'_{j_{\rho-1}} \end{bmatrix} \tag{A4}$$

with $p'_{j_i} = (rp_{j_i}) \pmod L$.

Proof. Since isomorphism of the codes is equivalent to isomorphism of their parity-check matrices (or exponent matrices), we only need to prove that $\mathbf{H}_{\mathbf{P}_1} \cong \mathbf{H}_{\mathbf{P}_2}$. Let V_i and C_i be the ordered sets of column and row indices of $\mathbf{H}_{\mathbf{P}_i}$ for $i = 1, 2$, respectively.

1) It is easy to see that there are $\rho!$ permutations on the set $\{0, 1, \ldots, \rho-1\}$. Consider any permutation π. $\mathbf{H}_{\mathbf{P}_1}$ is isomorphic to $\mathbf{H}_{\mathbf{P}_2}$ based on the bijections $f_1 : C_1 \to C_2$ and $f_2 : V_1 \to V_2$ given by $f_1(i) = i$ if $i \in C_1$, and $f_2(j) = m$-th column of $\pi(n)$-th column-SM of $\mathbf{H}_{\mathbf{P}_2}$ if $j \in V_1$ is the m-th column of the n-th column-SM of $\mathbf{H}_{\mathbf{P}_1}(0 \leqslant m \leqslant L-1, 0 \leqslant n \leqslant \rho-1)$.

2) We can see from (A3) that the entries of the second row of $\mathbf{P}_2$ are obtained from that of $\mathbf{P}_1$ by subtracting a constant value c, which is selected from the elements of the second row of $\mathbf{P}_1$, from the corresponding elements in the second row of $\mathbf{P}_1$ modulo L. Without loss of generality, consider $c = p_{j_0}$. $\mathbf{H}_{\mathbf{P}_1}$ is isomorphic to $\mathbf{H}_{\mathbf{P}_2}$ based on the bijections $f_1 : C_1 \to C_2$ and $f_2 : V_1 \to V_2$ given by $f_1(i) = (i-c) \pmod L$ if i is the h-th row of the first row-SM of $\mathbf{H}_{\mathbf{P}_1}$, and $f_1(i) = i$, otherwise ($0 \leqslant h \leqslant L-1$); and $f_2(j) = (m-c) \pmod L + nL$ if $j \in V_1$ is the m-th column of the n-th column-SM of $\mathbf{H}_{\mathbf{P}_1}$ ($0 \leqslant m \leqslant L-1, 0 \leqslant n \leqslant \rho-1$).

3) The elements of the second row of $\mathbf{P}_2$ in (A4) are obtained from that of $\mathbf{P}_1$ by multiplying a constant value r, which is coprime with L, by the corresponding elements in the second row of $\mathbf{P}_1$ modulo L. $\mathbf{H}_{\mathbf{P}_1}$ is isomorphic to $\mathbf{H}_{\mathbf{P}_2}$ based on the bijections $f_1 : C_1 \to C_2$ and $f_2 : V_1 \to V_2$ given by $f_1(i) = (rh) \pmod L + kL$ if i is the h-th row of the k-th row-SM of $\mathbf{H}_{\mathbf{P}_1}$ ($0 \leqslant h \leqslant L-1, 0 \leqslant k \leqslant 1$); and $f_2(j) = (rm) \pmod L + nL$ if i is the m-th column of the n-th column-SM of $\mathbf{H}_{\mathbf{P}_1}$ ($0 \leqslant m \leqslant L-1, 0 \leqslant n \leqslant \rho-1$).

That is, $\mathbf{H}_{\mathbf{P}_1} \cong \mathbf{H}_{\mathbf{P}_2}$ ($\mathbf{P}_1 \cong \mathbf{P}_2$), or equivalently $\mathcal{C}_1 \cong \mathcal{C}_2$.

Note that the number of r in 3) of Theorem 4 is determined by the following function $\phi(L)$, so-called Euler function,

$$\phi(L) = L \prod_{p \mid L} \left(1 - \frac{1}{p}\right),$$

where the subscript $p \mid L$ denotes for the prime divisors of L.

Appendix B Some search results of non-isomorphic exponent matrices

In this appendix, we briefly provide some search results of non-isomorphic exponent matrices of NBLDPC cycle codes of length ρL with girth $g = 12$. For $\rho = 3, 4, 5$ and some given values of L, the derived results are recorded in Table B1. Notice that some isomorphic cycle codes can be also found in [1–6].

Table B1 Exponent matrices and cycle distribution of all or some selected non-isomorphic girth-12 codes for $\rho = 3, 4, 5$

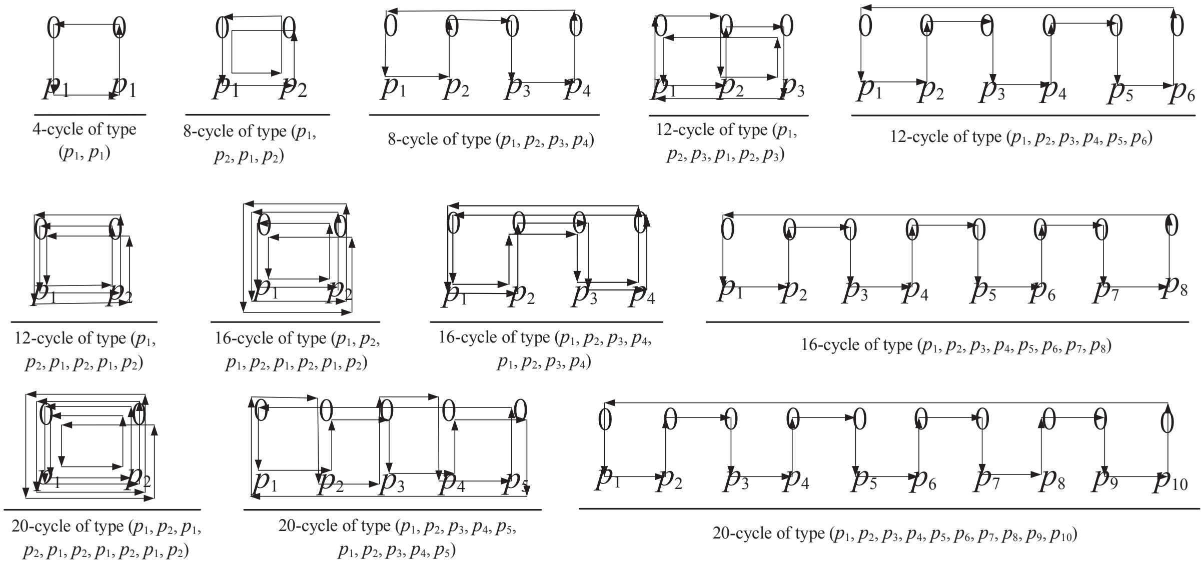
ρ	L	Number of	Second row of all or some	Cycle distribution		
		non-isomorphic codes	selected exponent matrices	12-cycles	16-cycles	20-cycles
3	7	1	$(0, 1, 3)$	28	21	84
	8	1	$(0, 1, 3)$	24	30	96
	9	1	$(0, 1, 3)$	21	36	99
	10	1	$(0, 1, 3)$	20	35	102
	11	1	$(0, 1, 3)$	22	22	121
	12	3	$(0, 1, 4)$	16	39	120
			$(0, 1, 5)$	16	42	96
			$(0, 1, 3)$	24	15	120
4	13	1	$(0, 1, 4, 6)$	234	702	5616
	14	1	$(0, 1, 4, 6)$	224	735	5712
	15	3	$(0, 1, 3, 7)$	210	810	5616
			$(0, 2, 3, 7)$	215	765	5763
			$(0, 1, 4, 6)$	215	765	5766
	16	2	$(0, 2, 3, 7)$	208	788	5744
			$(0, 1, 4, 6)$	208	788	5760
	17	2	$(0, 1, 3, 7)$	204	782	5882
			$(0, 1, 4, 6)$	204	799	5712
	32	34	$(0, 1, 5, 12)$	128	992	5888
			$(0, 2, 6, 11)$	160	784	6400
	64	205	$(0, 1, 11, 15)$	256	384	6656
			$(0, 1, 7, 17)$	256	400	5888
5	21	1	$(0, 2, 7, 8, 11)$	1120	7560	102816
	22	None				
	23	1	$(0, 2, 7, 8, 11)$	1081	7728	103523
	24	2	$(0, 1, 4, 9, 11)$	1040	8100	102240
			$(0, 2, 7, 8, 11)$	1064	7806	103632
	25	2	$(0, 1, 4, 9, 11)$	1025	8150	102460
			$(0, 2, 7, 8, 11)$	1050	7850	103905

Appendix C Types of cycles of lengths 4, 8, 12, 16, and 20 and their corresponding number of cycles

Under the equivalence relation, we classify nonequivalent cycles of lengths 4, 8, 12, 16, and 20 into twelve types, shown in Figure C1. Following the cyclic-shift structure of CPMs and a simple counting argument, we can obtain the corresponding number of cycles of each nonequivalent type, recorded in Table C1.

Appendix D Complexity analysis

To show the effectiveness of our proposed algorithm, a comparison with four competitive algorithms in [7–10] is given in Table D1. Notice that g is the girth of a $(2, \rho)$-regular LDPC code given by the null space of $\mathbf{H}$ in (A1) and L is the lifting degree. Generally, since $\rho < L$, the complexity of our proposed algorithm is much lower than that of the algorithm in [7]. Compared with the algorithms in Class-II of Table D1, our proposed algorithm may have lower complexity if L is sufficiently large and ρ is small. Note that these algorithms are only suitable for counting short cycles of lengths $g, g + 2, ..., 2g - 2$ and the algorithm in [7] only can count the cycles of lengths 4, 6, 8, and 10. But our proposed algorithm is capable of counting the number of $2i$-cycles in the Tanner graph for $i \geqslant g/2$, while the computational complexity will increase exponentially with the increase of i. Moreover, the algorithms in Class-II of Table D1 cannot enumerate the cycles to count, but our proposed algorithm can enumerate all cycles in a regular manner. This is important for optimizing NBLDPC codes by using the cycle cancellation method [11].



4-cycle of type (p_1, p_1)

8-cycle of type (p_1, p_2, p_1, p_2)

8-cycle of type (p_1, p_2, p_3, p_4)

12-cycle of type $(p_1, p_2, p_3, p_1, p_2, p_3)$

12-cycle of type $(p_1, p_2, p_3, p_4, p_5, p_6)$

12-cycle of type $(p_1, p_2, p_1, p_2, p_1, p_2)$

16-cycle of type $(p_1, p_2, p_1, p_2, p_1, p_2, p_1, p_2)$

16-cycle of type $(p_1, p_2, p_3, p_4, p_1, p_2, p_3, p_4)$

16-cycle of type $(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)$

20-cycle of type $(p_1, p_2, p_1, p_2, p_1, p_2, p_1, p_2, p_1, p_2)$

20-cycle of type $(p_1, p_2, p_3, p_4, p_5, p_1, p_2, p_3, p_4, p_5)$

20-cycle of type $(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10})$

Figure C1 All types of cycles of lengths 4, 8, 12, 16, and 20.

Table C1 Nonequivalent types and their corresponding number of cycles

	Nonequivalent type of cycles	Corresponding number of cycles
4-cycle	(p_1, p_1)	L
8-cycle	(p_1, p_2, p_1, p_2)	$L/2$
8-cycle	(p_1, p_2, p_3, p_4)	L
12-cycle	$(p_1, p_2, p_1, p_2, p_1, p_2)$	$L/3$
12-cycle	$(p_1, p_2, p_3, p_1, p_2, p_3)$	$L/2$
12-cycle	$(p_1, p_2, p_3, p_4, p_5, p_6)$	L
16-cycle	$(p_1, p_2, p_1, p_2, p_1, p_2, p_1, p_2)$	$L/4$
16-cycle	$(p_1, p_2, p_3, p_4, p_1, p_2, p_3, p_4)$	$L/2$
16-cycle	$(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)$	L
20-cycle	$(p_1, p_2, p_1, p_2, p_1, p_2, p_1, p_2, p_1, p_2)$	$L/5$
20-cycle	$(p_1, p_2, p_3, p_4, p_5, p_1, p_2, p_3, p_4, p_5)$	$L/2$
20-cycle	$(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10})$	L

Table D1 Comparison of algorithms for counting short cycles in the $(2, \rho)$-regular girth-g LDPC code of length ρL

	Proposed algorithm	Algorithm [7] (J. Fan, 2006)	Algorithm [8] (M. Karimi, 2012)	Algorithm [9] (M. Karimi, 2013)	Algorithm [10] (J. Li, 2015)
Complexity of counting $2i$-cycles	$\mathcal{O}(\rho^i)$	$\mathcal{O}((2L)^{i+1})$	$\mathcal{O}(\rho^3 L)$	$\mathcal{O}(\rho^2 g L^2)$	$\mathcal{O}(\rho g L)$
Lengths k of cycles to count	$k \geqslant g$	$k = 4, 6, 8, 10$	$g \leqslant k \leqslant 2g-2$	$g \leqslant k \leqslant 2g-2$	$g \leqslant k \leqslant 2g-2$
If enumerate cycles, or not	Yes	Yes	No	No	No
Classification	Class-I	Class-I	Class-II	Class-II	Class-II

Appendix E Numerical simulation results and analysis

In this appendix, we first take an example to study the effect of girth and cycle distribution on the iterative decoding performance.

Example 1. As shown in Table B1, there are 34 non-isomorphic LDPC cycle codes with girth 12 for $\rho = 4$ and $L = 32$. Based on the second row of exponent matrix in Table B1, e.g., $(0, 1, 5, 12)$, we can obtain a $(128, 64)$ LDPC cycle code $\mathcal{C}_1$ over GF(256). The cycle distribution of this code is also given in Table B1. In order to compare with this code, two $(128, 64)$ LDPC cycle codes $\mathcal{C}_2$ and $\mathcal{C}_3$ over GF(256) are constructed, whose exponent matrices are given by

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 6 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 8 & 12 \end{bmatrix},$$

respectively. The cycle distributions of $\mathcal{C}_2$ and $\mathcal{C}_3$ are $0x^4 + 32x^8 + 224x^{12} + 960x^{16} + 4704x^{20} + \cdots$ and $0x^4 + 0x^8 + 416x^{12} + 1576x^{16} + 11520x^{20} + \cdots$, respectively. The nonzero field elements in the parity-check matrices of these three codes are chosen at random. Their word error performances over the AWGN channel with BPSK signaling are shown in Figure E1. The FFT-QSPA with 50 iterations is employed to decode these three codes. We can see from Figure E1 that $\mathcal{C}_1$ performs about 0.4 dB and 0.6 dB better than $\mathcal{C}_2$ and $\mathcal{C}_3$ at WER $= 10^{-5}$, respectively. It is clear that $\mathcal{C}_1$ and $\mathcal{C}_3$ have girth 12, and that the number of 12-cycles, 16-cycles, and 20-cycles of $\mathcal{C}_1$ is less than that of $\mathcal{C}_3$, respectively. Although the girth of $\mathcal{C}_2$ is 8, but the number of short cycles of $\mathcal{C}_2$ is also less than that of $\mathcal{C}_3$. Moreover, the total number of 4-cycles, 8-cycles, 12-cycles, and 16-cycles of $\mathcal{C}_1$ is smaller than that of $\mathcal{C}_2$, but larger than that of $\mathcal{C}_2$ if the number of 20-cycles is added to it. So short cycles, especially the cycles of lengths not more than 16, severely affect the performance of NBLDPC cycle codes when decoded with the iterative algorithm. That is, girth and cycle distribution play important roles in the design of LDPC codes.

Remark 1. For the same code rate, code length, and finite field, NBLDPC codes with similar cycle distribution, almost have the same performance in the low SNR (or FER $\geqslant 10^{-5}$) region when decoded with the iterative algorithms, but the code whose Tanner graph has smaller number of short cycles, especially cycles of lengths less than 20, will perform better in the high SNR region if the effect of the nonzero field elements on the performance is not considered. So we say all NBLDPC cycle codes, the Tanner graphs of which have fewer number of short cycles (such as 8-cycles, 12-cycles, 16-cycles), can be taken as “good” codes for a given degree distribution and code length if the nonzero elements are chosen at random. When the nonzero elements are optimized for these codes, it is difficult to determine which one is the best here.

Remark 2. Following our search results, we find that some existing $(2, \rho)$-regular LDPC cycle codes have good cycle distributions for a given degree distribution and code length, and they have similar performance in the low SNR region. Which one is best in the high SNR region depends on the following two aspects:

- the number of short cycles in the corresponding Tanner graph;
- optimization of nonzero field elements in the corresponding parity-check matrices.

As stated in [12, 13], some codewords in the low-weight regime of the distance spectrum can degrade the decoding performance of NBLDPC cycle codes by performing the iterative algorithms. In order to increase symbol/bit distance of the proposed NBLDPC cycle codes, we employ the cycle cancellation method [11] to optimize nonzero finite field elements of their parity-check matrices. Next, we employ an example to illustrate the effect of nonzero finite field elements and also to show the performance of the optimized codes.

Example 2. Consider a $(96, 48)$ LDPC cycle code over GF(64) whose exponent matrix is

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 10 \end{bmatrix}.$$

Its cycle distribution is $0x^4 + 0x^8 + 144x^{12} + 1032x^{16} + 5616x^{20} + \cdots$. Using the proposed method for the optimization of nonzero field elements, 12-cycles and 16-cycles of $\mathcal{C}_2$ have been cancelled, while 108 20-cycles cannot be cancelled. The nonzero elements in the parity-check matrix of this code are represented by the power of α, where α is a primitive element of GF(64) created by using the primitive polynomial $p(x) = 1 + x + x^6$. The power numbers of nonzero field elements are given in Table E1. The word error performance of this code is given in Figure E2. The simulation parameters in this example are given as follows: the received words are decoded with the FFT-QSPA (50 iterations) and the transmission is over the AWGN channel with BPSK modulation. For comparison, we also plot the simulation results of two another $(96, 48)$ LDPC cycle codes over GF(64). The first code with girth 12 is created by using the PEG algorithm [14]. The second code is also constructed based on the above exponent matrix $\mathbf{P}$. Notice that the nonzero field elements in the parity-check matrices of these two codes are randomly chosen. It is easy to find that for the second code, there are one 12-cycle, twenty 16-cycles, and one hundred and one 20-cycles not being cancelled. It can be seen from Figure E2 that as expected, our optimized code outperforms the other two codes and only performs 0.55 dB away from the finite length bound [15] of block length 576 bits and code rate 0.5 at the WER of 10^{-5}. In other words, the optimization of nonzero field elements by employing the cycle cancellation method takes good effect in improving the performance under iterative decoding.

References

1 Huang J, Zhou S L, Zhu J, et al. Group-theoretic analysis of Cayley-graph-based cycle GF(2^p) codes. IEEE Transactions on Communications, 2009, 57: 1560–1565

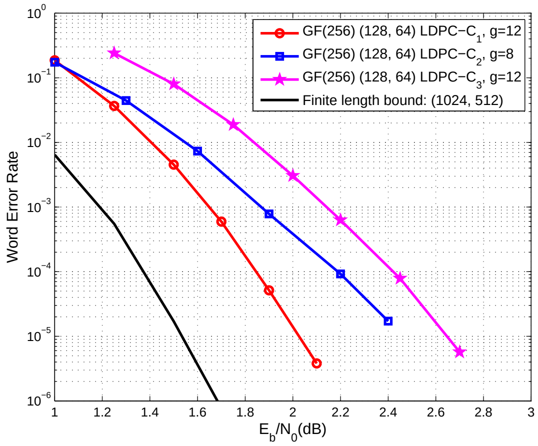


Figure E1 Word error performances of three proposed (128, 64) QC LDPC cycle codes over GF(256) when decoded using the FFT-QSPA with 50 iterations.

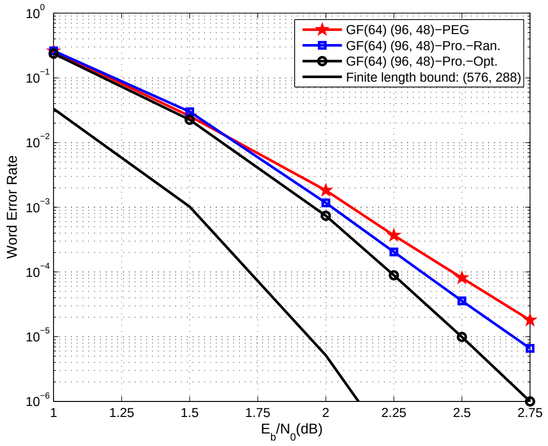


Figure E2 Word error performances of three (96, 48) LDPC cycle codes over GF(64) given in Example 2.

Table E1 The nonzero field elements in the parity-check matrix of the optimized (96, 48) LDPC cycle code over GF(64) in Example 2

Row	Nonzero field elements				Row	Nonzero field elements				Row	Nonzero field elements			
index	(power number)				index	(power number)				index	(power number)			
0	44	18	7	0	16	22	37	0	9	32	9	19	0	37
1	9	19	0	37	17	9	37	0	22	33	22	9	37	0
2	0	9	37	19	18	0	44	18	7	34	19	0	9	37
3	18	0	7	44	19	22	0	9	37	35	18	0	44	7
4	37	0	9	22	20	18	44	7	0	36	0	9	19	37
5	0	9	37	22	21	37	9	19	0	37	18	7	44	0
6	22	9	37	0	22	9	0	37	22	38	9	0	19	37
7	19	37	9	0	23	44	0	18	7	39	0	37	9	19
8	0	37	9	22	24	22	9	0	37	40	9	37	19	0
9	22	0	37	9	25	22	9	0	37	41	9	0	37	22
10	0	37	22	9	26	37	22	0	9	42	0	37	9	19
11	44	18	7	0	27	9	19	0	37	43	0	9	22	37
12	9	37	0	19	28	0	18	7	44	44	18	7	44	0
13	0	18	44	7	29	22	37	0	9	45	37	9	0	22
14	9	19	37	0	30	18	0	7	44	46	44	7	0	18
15	9	37	19	0	31	0	37	19	9	47	37	9	0	22

2 Huang J, Zhou S L, Willett P. Structure, property, and design of nonbinary regular cycle codes. IEEE Transactions on Communications, 2010, 58: 1060–1071

3 Chen C, Bai B M, Wang X M. Construction of nonbinary quasi-cyclic LDPC cycle codes based on singer perfect difference set. IEEE Communications Letters, 2010, 14: 181–183

4 Mohammad G, Mehdi S. Design of binary and nonbinary codes from lifting of girth-8 cycle codes with minimum lengths. IEEE Communications Letters, 2013, 17: 777–780

5 Gholami M, Ghaffar R. Large girth column-weight two and three LDPC codes. IEEE Communications Letters, 2014, 18: 1671–1674

6 Zhao S, Ma X. Construction of high-performance array-based nonbinary LDPC codes with moderate rates. IEEE Communications Letters, 2016, 20: 13–16

7 Fan J, Xiao Y. A method of counting the number of cycles in LDPC codes. In: Proceedings of the 8th International Conference on Signal Processing, 2006. 2183–2186

8 Karimi M, Banihashemi A H. Counting short cycles of quasi cyclic protograph LDPC codes. IEEE Communications Letters, 2012, 16: 400–403

9 Karimi M, Banihashemi A H. Message-passing algorithm for counting short cycles in a graph. IEEE Transactions on Communications, 2013, 61: 485–495

10 Li J, Lin S, Abdel-Ghaffar K. Improved message-passing algorithm for counting short cycles in bipartite graphs. In: Proceedings of IEEE International Symposium on Information Theory, HongKong, 2015. 416–420

11 Poulliat C, Fossorier M P C, Declercq D. Design of regular $(2, d_c)$ LDPC codes over GF(q) using their binary images. IEEE Transactions on Communications, 2008, 56: 1626–1635

12 Hu X Y, Eleftheriou E. Binary representation of cycle Tanner-graph GF(2^b) codes. In: Proceedings of IEEE International Conference on Communications, Paris, 2004. 528–532

13 Liu L, Huang J, Zhou W, et al. Computing the minimum distance of nonbinary LDPC codes. IEEE Transactions on Communications, 2012, 60: 1753–1758

14 Hu X Y, Eleftheriou E, Arnold D. Regular and irregular progressive edge-growth Tanner graphs. IEEE Transactions on Information Theory, 2005, 51: 386–398

15 Polyanskiy Y, Poor H V, Verdú S. Channel coding rate in the finite blocklength regime. IEEE Transactions on Information Theory, 2010, 56: 2307–2359