SCIENCE CHINA
Information Sciences

• LETTER •

August 2018, Vol. 61 089303:1–089303:3
https://doi.org/10.1007/s11432-017-9271-6

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*

Received 22 March 2017/Revised 19 July 2017/Accepted 27 October 2017/Published online 19 April 2018

Citation Xu H Z, Chen C, Zhu M, et al. Nonbinary LDPC cycle codes: efficient search, design, and code optimization. Sci China Inf Sci, 2018, 61(8): 089303, https://doi.org/10.1007/s11432-017-9271-6

Dear editor,

Nonbinary low-density parity-check (NBLDPC) codes have the ability of approaching capacity when decoded iteratively using a probabilistic decoding algorithm. With the increase of the finite field size q, NBLDPC codes over GF(q) have much better performance under iterative decoding for a constant code length. For q sufficiently large, there is no significant performance improvement while increasing q, moreover the column weights of the parity-check matrices of the best codes tend to 2. As an important class of NBLDPC codes, $(2,\rho)$-regular NBLDPC codes over GF(q) ($q \geqslant 64$), whose parity-check matrices have row weight ρ and column weight 2, perform well over various channels. This class of NBLDPC codes, so-called NBLDPC cycle codes, has attracted much attention [1–3]. Among these works, all designed codes have good performance. For a given code rate and code length, it is of great interest to study which one of them has the best error performance.

In this article, we first present an algorithm to search for an ensemble of NBLDPC cycle codes for a given row weight and code length. From the perspective of isomorphism, we classify the resulting codes into non-isomorphic codes based on their corresponding Tanner graphs. By analyzing the cycles of NBLDPC cycle codes, a simple algorithm for counting short cycles, also applicable to (γ,ρ)-regular quasi-cyclic (QC) LDPC codes, is proposed. According to these two algorithms, we can obtain non-isomorphic NBLDPC cycle codes with optimized cycle distribution for a given code length and code rate. Moreover, in order to improve the error performance in the waterfall and error-floor regions, we employ the cycle cancellation method [4] to optimize nonzero field elements in the parity-check matrices of NBLDPC cycle codes. Different from the prior study of [5], we in this article analyze the cycle structure of NBLDPC cycle codes and propose a new algorithm for counting and enumerating cycles in the Tanner graphs of NBLDPC cycle codes. Moreover, we present an efficient exhaustive search of NBLDPC cycle codes based on isomorphism theory. It is shown in [6] that connected cycles may influence the performance of NBLDPC cycle codes. However, in order to find out the best one from all non-isomorphic NBLDPC cycle codes simply, we only consider cycle distribution.

Isomorphism and NBLDPC cycle codes. Two Tanner graphs $\mathcal{G}_1(V_1,C_1)$ and $\mathcal{G}_2(V_2,C_2)$ are isomorphic, denoted by $\mathcal{G}_1(V_1,C_1) \cong \mathcal{G}_2(V_2,C_2)$, if there exist bijections $f_1: V_1 \to V_2$ and $f_2: C_1 \to C_2$ such that there is an edge between $v_1 \in V_1$ and $c_1 \in C_1$ if and only if there is an edge between $f_1(v_1) \in V_2$ and $f_2(c_1) \in C_2$. Let $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ be the biadjacency matrices of $\mathcal{G}_1(V,C)$ and $\mathcal{G}_2(V,C)$,

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

respectively. Equivalently, $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are isomorphic, denoted by $\boldsymbol{A}_1 \cong \boldsymbol{A}_2$, if there are bijections f_1 and f_2 such that the elements $(\boldsymbol{A}_1)_{ij} = 1$ if and only if $(\boldsymbol{A}_2)_{f_1(i)f_2(j)} = 1$. If $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are isomorphic, their corresponding exponent matrices are also isomorphic. Consider two LDPC codes $\mathcal{C}_1$ and $\mathcal{C}_2$ given by the null spaces of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, respectively. If $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are isomorphic, we say that two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are isomorphic.

Based on Theorems 1 and 3 in Appendix A, the parity-check matrix of an NBLDPC cycle code of length ρL can be simplified as

$$\boldsymbol{H}_{\boldsymbol{P}} = \begin{bmatrix} \overline{\boldsymbol{I}}(0) & \overline{\boldsymbol{I}}(0) & \cdots & \overline{\boldsymbol{I}}(0) \\ \overline{\boldsymbol{I}}(p_0)(=\overline{\boldsymbol{I}}(0)) & \overline{\boldsymbol{I}}(p_1) & \cdots & \overline{\boldsymbol{I}}(p_{\rho-1}) \end{bmatrix}, \quad (1)$$

where $p_0 = 0$ and $0 \leqslant p_i \leqslant L-1$ for $1 \leqslant i \leqslant \rho-1$. For $a \in \{p_0, \ldots, p_{\rho-1}\}$, $\overline{\boldsymbol{I}}(a)$ is an $L \times L$ square matrix obtained by replacing ones of circulant permutation matrix (CPM) $\boldsymbol{I}(a)$ with the nonzero elements of GF(q). In order to avoid the cycles of length 4 in the NBLDPC cycle codes, we assume that $\rho \leqslant L$. Here we denote such square matrices $\overline{\boldsymbol{I}}(a)$ over GF(q) as SMs $\overline{\boldsymbol{I}}(a)$ for short. Obviously, the exponent matrix of $\boldsymbol{H}_{\boldsymbol{P}}$ is

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ p_0(=0) & p_1 & \cdots & p_{\rho-1} \end{bmatrix}. \quad (2)$$

Notice that such $\boldsymbol{P}$ can also be obtained based on Corollary 1 in [7]. Different from their study, we give the conclusion from the perspective of CPMs and their row permutations.

Cycles in NBLDPC cycle codes. A cycle in the Tanner graph of the NBLDPC cycle code $\mathcal{C}$ over GF(q) given by the null space of $\boldsymbol{H}_{\boldsymbol{P}}$ is associated with a family of the ordered SMs in $\boldsymbol{H}_{\boldsymbol{P}}$. As shown in [5], a $2i$-cycle in the Tanner graph of $\mathcal{C}$ (or $\boldsymbol{H}_{\boldsymbol{P}}$) is represented by an ordered sequences of SMs

$$\overline{\boldsymbol{I}}(0), \overline{\boldsymbol{I}}(0), \overline{\boldsymbol{I}}(p_{j_1}), \overline{\boldsymbol{I}}(p_{j_2}), \ldots, \overline{\boldsymbol{I}}(0), \overline{\boldsymbol{I}}(p_{j_{i-1}}), \overline{\boldsymbol{I}}(p_{j_i}),$$

where $j_{i+1} = j_1$, $0 \leqslant j_k \leqslant \rho-1$, $j_k \neq j_{k+1}$, and $0 \leqslant p_{j_k} \leqslant L-1$ for $1 \leqslant k \leqslant i$. More simply, such a $2i$-cycle can be written as type $(p_{j_1}, p_{j_2}, \ldots, p_{j_i})$. In order to classify cycles into distinct types, we give the equivalent relation of cycles as follows.

Theorem 1. Let $p_m, p'_m \in \{0, 1, \ldots, L-1\}$, for $0 \leqslant m \leqslant i-1$. Type $(p_0, p_1, \ldots, p_{i-1})$ and type $(p'_0, p'_1, \ldots, p'_{i-1})$ are equivalent if there exist some c such that $p'_m = p_{m+c \pmod i}$ for all m.

Let g be the girth of the code $\mathcal{C}$. For $g \leqslant 2i \leqslant 2g-2$, the necessary and sufficient condition for the existence of a $2i$-cycle in the Tanner graph of the code $\mathcal{C}$ corresponds to the following relation:

$$\sum_{k=1}^{i} (-1)^k (0 - p_{j_k}) = 0 \pmod L, \quad (3)$$

with $j_1 = j_{i+1}$ and $j_k \neq j_{k+1}$.

An efficient exhaustive search of non-isomorphic exponent matrices. We can see from (2) that the size of the search space of $\boldsymbol{P}$ is $(L)^{\rho-1}$, since $0 \leqslant p_i \leqslant L-1$, $1 \leqslant i \leqslant \rho-1$. As L and ρ increase, this number is horrible. By employing isomorphism theory, we can further reduce the search space. Notice that, for ρ being large, the size of the search space is still too large although it can be reduced. Moreover, for high rate codes, i.e., $\rho \approx L$, the search space is limited. Hence, the exhaustive search is suitable for NBLDPC cycle codes with moderate code rates.

Based on Theorem 4 in Appendix A, isomorphism of the exponent matrices (or their corresponding parity-check matrices and codes) reduces strongly the size of search space. An exhaustive search of non-isomorphic exponent matrices can be achieved. According to (1) in Theorem 4 in Appendix A, we can construct NBLDPC cycle codes with girth at least 8, and then we only need an exhaustive search for testing all possible combinations of $p_1, p_2, \ldots, p_{\rho-1}$ in the exponent matrix $\boldsymbol{P}$ given by (2), where $p_1, p_2, \ldots, p_{\rho-1} \in \{1, 2, \ldots, L-1\}$ are distinct. Therefore, the size of search space is further limited to $\binom{L-1}{\rho-1} = \frac{(L-1)!}{(\rho-1)!(L-\rho)!}$ with $L \geqslant \rho$. This also results in $\binom{L-1}{\rho-1}$ exponent matrices. Based on these exponent matrices, we can easily construct $\binom{L-1}{\rho-1}$ NBLDPC cycle codes with girth at least 8. Furthermore, we can also classify these codes (or exponent matrices) into non-isomorphic codes (or non-isomorphic exponent matrices) based on (2) and (3) in Theorem 4 in Appendix A. The resulting non-isomorphic codes (or exponent matrices) are served as output. On the other hand, we can also look for the non-isomorphic codes (or non-isomorphic exponent matrices) in the process of searching the combinations of $p_1, p_2, \ldots, p_{\rho-1}$. We first find an exponent matrix, then calculate out its isomorphic exponent matrices based on (2) and (3) in Theorem 4 in Appendix A. The resulting matrices are stored and available for the following process. Next, we continue to search another exponent matrix which is different from the ones obtained earlier and then find its isomorphic exponent matrices as well. Repeat the above process until all the $\binom{L-1}{\rho-1}$ combinations of $p_1, p_2, \ldots, p_{\rho-1}$ are found. In our search, the latter procedure is employed. To show the effectiveness of our proposed algorithm, some search results of non-isomorphic exponent matrices are given in Table B1 in Appendix B.

An algorithm for counting and enumerating cycles in NBLDPC cycle codes. Based on the afore-

mentioned, a $2i$-cycle can be expressed as an i-tuple, i.e., type $(p_0, p_1, \ldots, p_{i-1})$. Hence, to find $2i$-cycles is equivalent to the search of i-tuples, whose elements are chosen from the elements of the second row in the exponent matrix $\boldsymbol{P}$ such that (3) holds. The proposed algorithm consists of three steps: (i) determine the nonequivalent types of cycles and their corresponding number of cycles; (ii) exhaustively search for nonequivalent types of cycles based on the exponent matrix $\boldsymbol{P}$ in (2); (iii) calculate the number of cycles with different lengths.

Based on (3), the necessary and sufficient condition for the existence of a $2i$-cycle of type $(p_{j_1}, p_{j_2}, \ldots, p_{j_i})$ is

$$-p_{j_1} + p_{j_2} + \cdots + (-1)^i p_{j_i} = 0 \pmod{L},$$

with $j_k \neq j_{k+1}$, $j_{i+1} = j_1$, $j_k \in \{0, 1, \ldots, \rho - 1\}$, and $g \leqslant 2i \leqslant 2g - 2$. Hence, the search of such an i-tuple $(p_{j_1}, p_{j_2}, \ldots, p_{j_i})$ can be easily done by a computer program. It is obvious that this is an exhaustive search and the computational complexity is about $\mathcal{O}(\rho^i)$, where $2i$ is the length of the searched cycles. When $2i \geqslant 2g$, Eq. (3) is not sufficient, since i-tuple which consists of several short cycles also satisfies (3). Hence, the tuple (or type of large cycles) which consists of types of several small cycles should be removed. For example, 4-tuple (or type) $(1, 1, 2, 2)$ contains type $(1, 1)$ and type $(2, 2)$ of 4-cycles, then there exist no 8-cycles of type $(1, 1, 2, 2)$. Finally, nonequivalent types are obtained. With these resultant types and the number of their corresponding cycles, the total number of $2i$-cycles can be calculated. Notice that types of cycles of lengths 4, 8, 12, 16, 20 and their corresponding number of cycles are given in Figure C1 and Table C1 in Appendix C, and a comparison with four competitive algorithms for counting short cycles is given in Table D1 in Appendix D.

Optimized NBLDPC cycle codes. It was shown in [8] that NBLDPC codes with large girth and small number of shortest cycles have good performance. Based on the above algorithm for counting short cycles, cycle distributions of all non-isomorphic NBLDPC cycle codes can be obtained for a given ρ and L. Thus, NBLDPC cycle codes with larger girth and smaller number of short cycles can be chosen as optimized codes.

In order to improve error performance in the waterfall and error-floor regions, we optimize nonzero finite field elements of the parity-check matrices to increase symbol/bit distance of the proposed NBLDPC cycle codes by using the cycle cancellation method [4]. The key of the method is to cancel short cycles as much as possible. It is remarkable that we can clearly find out short cycles of NBLDPC cycle codes to be optimized. Therefore, the cancel process of short cycles can be easily performed by properly selecting nonzero field elements of $\mathrm{GF}(q)$. Note that the optimized code, in which most of short cycles have been cancelled, is used as the output code. To show the good performance of our proposed codes, numerical simulation results and analysis are given in Appendix E.

Conclusion. In this article, we studied the isomorphism of NBLDPC cycle codes and proposed an efficient exhaustive algorithm to find an ensemble of non-isomorphic LDPC cycle codes for a given row weight and code length. Also proposed is a simple algorithm for counting and enumerating short cycles of NBLDPC cycle codes. Based on these two algorithms and the cycle cancellation method, we can easily construct an NBLDPC cycle code with optimized cycle distribution for a given code rate and code length. Simulation results show that the designed codes perform well under iterative decoding.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. U1504601, 61771364, 61701368, 61671342, 61372074, 91438101).

Supporting information Appendixes A–E. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Dolecek L, Divsalar D, Sun Y, et al. Non-binary protograph-based ldpc codes: enumerators, analysis, and designs. IEEE Trans Inf Theory, 2014, 60: 3913–3941
2 Zhao S C, Ma X. Construction of high-performance array-based non-binary LDPC codes with moderate rates. IEEE Commun Lett, 2016, 20: 13–16
3 Li J E, Liu K K, Lin S, et al. A matrix-theoretic approach to the construction of non-binary quasi-cyclic LDPC codes. IEEE Trans Commun, 2015, 63: 1057–1068
4 Chen C, Bai B M, Shi G M, et al. Nonbinary LDPC codes on cages: structural property and code optimization. IEEE Trans Commun, 2015, 63: 364–375
5 Sun C, Xu H Z, Feng D, et al. (3, L) quasi-cyclic LDPC codes: simplified exhaustive search and designs. In: Proceedings of the 9th International Symposium on Turbo Codes and Iterative Information Processing, Brest, 2016. 271–275
6 Zhao S C, Huang X J, Ma X. Structural analysis of array-based non-binary LDPC codes. IEEE Trans Commun, 2016, 64: 4910–4922
7 Tasdighi A, Banihashemi A H, Sadeghi M R. Efficient search of girth-optimal QC-LDPC codes. IEEE Trans Inf Theory, 2016, 62: 1552–1564
8 Xu H Z, Bai B M. Superposition construction of Q-ary LDPC codes by jointly optimizing girth and number of shortest cycles. IEEE Commun Lett, 2016, 20: 1285–1288