A New Method to Construct Non-binary QC-LDPC Codes Based on ACE Value

Xinting Wang[1], Kegang Pan[1(✉)], Rong Lv[2], and Ruixiang Zhao[1]

[1] College of Communication Engineering, Army Engineering University of PLA, Nanjing, China
wangxinting_wxt@163.com, pankg@163.com

[2] The Sixty-third Research Institute, National University of Defense Technology, Nanjing, China
lvrong17@nudt.edu.cn

Abstract. This paper presented a method for constructing NB-QC-LDPC (non-binary quasi-cyclic LDPC) codes. First, the initial base matrix of NB-QC-LDPC code was constructed by two arbitrary subsets in a finite field. Then, by combining the number and connectivity of cycles jointly, the new masking method was proposed to construct a type of NB-QC-LDPC codes with larger ACE average values. The simulation illustrates that proposed codes have better error correction performance compared with binary LDPC codes and other NB-LDPC codes.

Keywords: Non-binary (NB) quasi-cyclic (QC) LDPC codes · Code constructed · Masking · The approximate cycle EMD (ACE)

1 Introduction

As one of the famous channel coding techniques, low-density parity-check (LDPC) codes [2] were proposed in 1962 to concrol errors in communication and data storage system due to their capacity-approaching performances and efficient parallel decoding mechanism.

Until 1998, Davey and Mackey firstly discovered non-binary (NB) LDPC codes [1] and put forward a q-ary sum product algorithm (QSPA) for decoding at the same time. Compared with binary LDPC codes, NB-LDPC codes do have the advantage of correcting random and burst errors that happened simultaneously in channels. In a higher finite field, NB-LDPC codes with much longer code length could approach the Shannon Limit. However, the higher computational complexity in coding and decoding makes it impractical. Therefore, NB-LDPC codes deserve more attention and research effort.

Similar to binary LDPC codes, there are two types of construction methods for NB-LDPC codes, including random construction method and structural construction method. For the former, given a degree distribution, the progressive

Suppported by National Natural Science Foundation of China, No. 61671476.

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 357, pp. 379–390, 2021.
https://doi.org/10.1007/978-3-030-69069-4_32

edge growth (PEG) algorithm [14] generates a parity-check matrix by column in which all connecting labels are randomly selected. In particular, the approximate cycle extrinsic message degree (ACE) algorithm [12] can reduce the error floor performance at high signal-to-noise ratios (SNRs) by ensuring the ACE value of some cycles is bigger than a given value. However, random methods take up a large amount of memory space to store the stochastic parity-check matrix.

The latter methods include algebraic approach [4,5,8,18,19], matrix theory [13,15–17], etc., are used to generate a type of quasi-cycle (QC) LDPC codes [9]. Codes based on finite fields GF(q) are called NB-QC-LDPC codes, which are given by the null space of an array $\boldsymbol{H}$ of sparse circulant matrices of the same size. The sparse circulant matrices in the parity-check array $\boldsymbol{H}$ are circulant permutation matrices (CPMs), which can save hardware storage space and simplify the coding and decoding process. To improve the error correction performance and reduce the computation complexity, several measures was proposed, such as eliminating short cycles [10], maximizing girth, improving Hamming distance [7], improving connectivity of cycles [3], etc.

In this paper, we combine the number with the connectivity of cycles to improve performance. Since not all short cycles are harmful to performance, selectively eliminate cycles with bad connectivity can make a excellent cycle condition [11]. The new masking technique is proposed to construct an irregular NB-QC-LDPC code with short and long code lengths. The simulation results show that the codes with larger ACE average values have excellent decoding performance.

The rest of paper is organized as follows. The basic theory of NB-QC-LDPC codes over a finite field is briefly introduced in Sect. 2. Section 3 describes our model and masking algorithms for constructing NB-QC-LDPC codes. Meanwhile, Sect. 3 illustrates with some examples. Simulation results and analysis will be discussed in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 NB-QC-LDPC CODES

2.1 Definitions and Concepts

QC-LDPC codes given by the null space of an array of sparse circulant permutation matrices (CPM) of the same size over a finite field GF(q) (q > 2) is called NB-QC-LDPC codes. While $q = 2$, they become to binary QC-LDPC codes. As for any positive integer r, let $\boldsymbol{Q}$ be a $r \times r$ CPM in GF(q) with columns and rows labeled from 0 to $r - 1$.

There are two categories of $\boldsymbol{Q}$. If all nonzero elements in $\boldsymbol{Q}$ is single, such $\boldsymbol{Q}$ is called the q-ary CPM and has the following structural characteristics: (1) the first row contains a single nonzero element in GF(q), at position between 0 and $r - 1$; (2) each row in $\boldsymbol{Q}$ is a cyclic right shift of the previous row and the first row is obtained from the last row shift to right. All nonzero elements in q-ary CPM are same, but are different in positions.

Another structure is called α^{λ}-multiplied CPM, in which all nonzero elements are different in both. To be specific, each row, except the first row, is obtained

by cyclic right shift of the previous row and the single nonzero value which alse belongs to GF(q) is the single nonzero value in the above row multiplied by α. Furthermore, q-ary CPM is much simpler in coding and decoding process. Thus, we will take q-ary CPMs as examples.

In the parity-check matrix $\boldsymbol{H}$, a cycle is a closed path that consists of a set of horizontal and vertical lines alternately, in which each vertex is a nonzero element. The length of a cycle must be even. For example, two rows and two columns can form a cycle whose length is 4. The shortest cycle is called girth. Cycle is an important factor during iterative decoding. Reducing short cycles and maximizing girth are common approaches to improve performance.

Various structures of LDPC code have to satisfy the row-column (RC) constraint [6]: no two rows (or two columns) have more than one position where they both have nonzero entries, which ensures the Tanner graph of the code with girth at least 6. In this paper, we are mainly concerned of RC-constrained NB-QC-LDPC codes.

2.2 Construction Principles and Masking

Suppose a finite field GF(q), $\mathrm{q} = 2^{\mathrm{s}}$. Let α be a primitive element, and all elements in GF(q) are represented as $\left\{\alpha^{-\infty}, \alpha^0, \alpha^1, \cdots, \alpha^{q-2}\right\}$. This section explains a construction method of the base matrix $\boldsymbol{B}$ of NB-QC-LDPC codes by using two random subsets in GF(q).

Let $S_1 = \left\{\alpha^{i_0}, \alpha^{i_1}, \cdots, \alpha^{i_{m-1}}\right\}$ and $S_2 = \left\{\alpha^{j_0}, \alpha^{j_1}, \cdots, \alpha^{j_{n-1}}\right\}$ are two random subsets, with $i_k, j_l \in \{-\infty, 0, 1, \cdots, q-2\}$, $0 \leqslant k < m$, $0 \leqslant l < n$, and $i_0 < i_1 < \cdots < i_{m-1}$, $j_0 < j_1 < \cdots < j_{n-1}$. Then we can use following rule to generate a base matrix $\boldsymbol{B}$ [4,5]:

$$\boldsymbol{B} = \left[\gamma\alpha^{i_k} + \alpha^{j_l}\right]_{0\leqslant k<m, 0\leqslant l<n} \tag{1}$$

The constructed base matrix has the following characteristics: (1) all the entries in the row (column) are different elements in GF(q); (2) each row (or column) contains at most one zero element; (3) no two rows (or two columns) have the same item in the same position; (4) any submatrix is nonsingular. According to the structure properties (2) and (4), $\boldsymbol{B}$ satisfies the RC constraint. γ is called the multiplier of $\boldsymbol{B}$.

The parity-check matrix $\boldsymbol{H}$ is generated by extending each element in $\boldsymbol{B}$ with q-ary CPMs of size $r \times r$, $r = q - 1$, whose generator has α^j as its single nonzero component at the position j, which makes sure the Tanner graph of it has girth at least 6 if the maximum value of j is less than r. The key to construct a QC-LDPC code is to design an optional base matrix $\boldsymbol{B}$ of which a number of structural properities determines the iterate decoding performance. Thus, the replacement results in replacing some nonzero elements by zero which means an $r \times r$ q-ary CPM is replaced by an $r \times r$ ZM (zero matrix), which is refered to as masking. The masking matrix $\boldsymbol{M}$ which only consists of "0" and "1" elements can be performed on $\boldsymbol{B}$ to obtain a masked based matrix $\boldsymbol{B}_{\text{mask}}$:

$$\boldsymbol{B}_{\text{mask}} = \boldsymbol{M} \otimes \boldsymbol{B} = \left[\boldsymbol{I}\left(m_{k,l} b_{k,l}\right)\right]_{0\leqslant k<m, 0\leqslant l<n} \tag{2}$$

If $m_{k,l} = 1$, $m_{k,l}b_{k,l} = b_{k,l}$; else if $m_{k,l} = 0$, $m_{k,l}b_{k,l} = 0$.

2.3 Cycles and ACE Value

Each column (row) in a base matrix $\boldsymbol{B}$ corresponds to a variable (check) node, and each element in $\boldsymbol{B}$ represents a connecting edge between two types of nodes. The total number of connecting edges of a node is called the node degree.

A cycle is composed of alternate connecting edges between variable nodes and check nodes, which influnce the error floor. The total degree of variable nodes in cycle determines the connectivity of cycle. Larger degree can increase decoding accuracy, which means more parity-check equations could be used to control errors and it can utilize more nodes to exchange useful information during decoding.

In general, the approximate cycle extrinsic message degree (ACE) of a cycle is one of the important characteristics, which used to roughly measure the connectivity of cycle. The ACE values in cycle g are defined as follows [4]:

$$ACE^g = \sum_i (d_i - 2) \tag{3}$$

d_i is ith node's degree in cycle g, $i = 1, 2, ..., g$. The ACE value of the variable node with degree d can be considered as $d - 2$ and the ACE value of the check node is 0.

From (3), the ACE value of cycle is directly determined by the total degree of variable nodes. A LDPC code with larger ACE value usually exhibits better error correction performance than a code with small ACE value. Hence, removing cycles with low connectivity and keeping the high may improve the error correction performance. When the short cycles is removed by masking, the ACE value of the remaining cycles will also be reduced. Making a compromise between them is what we have to discuss next.

3 Masking Algorithm Based on ACE Value

Cycle is an important factor affecting the performance of QC-LDPC codes, particularly length of 4. However, cycles with different length may have different effects on performance. Not only the number of short cycles, but also the connectivity of cycles plays an important role. Remaining cycles with high connectivity rather than reducing short cycles for increasing girth blindly could improve error-floor effectively. The masking algorithm is prensented to distinguish and select more harmful cycles to make a better cycle condition.

In this paper, we emphasize the importance of ACE value and give priority to removing cycles with smaller ACE values, despite the number of cycles may be increased. Therefore, the mathematical model is proposed by jointly considering both the number and connectivity of cycles to choose one place reasonably if there are more than one nonzero position that could be turned to zero.

The entire masking process is showen in Fig. 1, If there is only one maximum in $\boldsymbol{D}$, turn the element in this position to zero and calculate the number of cycle g of the new base matrix $\boldsymbol{B}$; if not, we need to consider the ACE value of cycle g^* to get a base matrix with larger ACE average value.

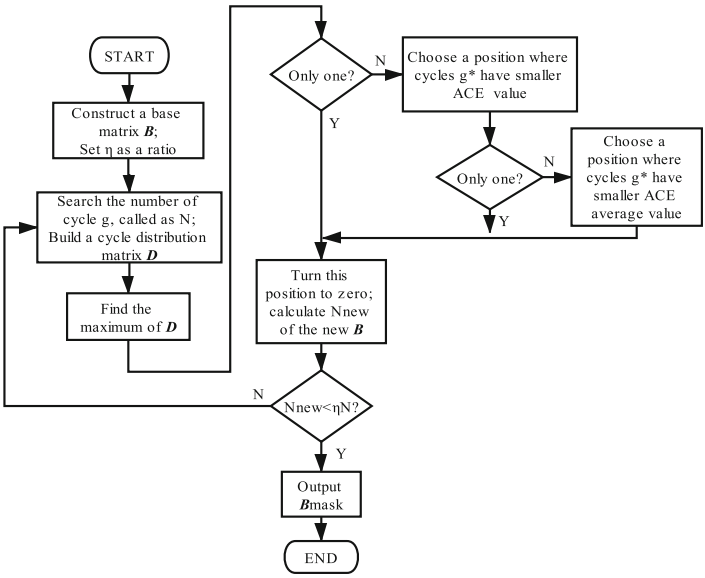

Fig. 1. Masking process

The mathematical model is as follows:

$$\boldsymbol{B}_{\text{mask}} = \arg\max\left(\overline{ACE}^{g^*}\right) \tag{4}$$

$$s.t. \begin{cases} N^g \leqslant \eta N_{\text{init}}^g \\ N^{g'} = 0, g' < g \\ g^* = \begin{cases} g & 0 < \eta \leqslant 1 \\ g+2 & \eta = 0 \end{cases} \end{cases} \tag{5}$$

In (4), $\boldsymbol{B}_{\text{mask}}$ is a masked base matrix; $\overline{ACE}^g$ is a ACE average value of cycle g. A masked base matrix $\boldsymbol{B}_{\text{mask}}$ that has the highest ACE average of cycle g^* is expected to get. However, if all cycles g^* are eventually eliminated, it is meaningless to control its ACE value. Therefore, according to the ratio η, we divide g^* into two situations to discuss.

The parameters in formulas are explained in (5). $\boldsymbol{N}^g$ is the total number of cycle g; $N^{g'}$ is the total number of cycle g', $g' < g$; N_{init}^g is a initial number of

cycle g; η is a ratio of number, $\eta = N^g/N^g_{\text{init}}$; g_{end} is a stopped length. The ratio η determines whose ACE value we should focus on during this process. When $0 < \eta \leqslant 1$, there are some cycles g left after finished, and $N^g > 0$. Thus cycles g with small ACE value need to be elimitated to ensure the final masked base matrix $\boldsymbol{B}_{\text{mask}}$ has large ACE average. In addition, when $\eta = 0$, $N^g > 0$ after finished, it will focus on deleting cycles g with small ACE value. The specific steps are as follows:

First, the original base matrix $\boldsymbol{B}$ is constructed by using the approach as mentioned above according to the code length and code rate.

Then, a replacement preprocessing is proposed to replace some elements in $\boldsymbol{B}$ by orderly searching for a nonzero element from 1 to p to initially reduce cycle g. See Algorithm 1 for details:

In algorithm, $\boldsymbol{D}^g$ is a cycle participation distribution matrix of cycle g.

$$\boldsymbol{D}^g = \left[d^g_{k,l}\right]_{0\leqslant k<m, 0\leqslant l<n} \tag{6}$$

$d^g_{k,l}$ represents the number of cycle g in which the element $b_{k,l}$ participates.

The core idea of the algorithm is to ensure that N^g decreases sharply, meanwhile, $N^{g'} = 0$, $g' < g$. If there is a nonzero element that makes $N^g = 0$, the next round will begin for cycle $g = g + 2$ until reach the maximum searching time. Output $\boldsymbol{B}_{\text{replace}}$ in the end. In general, no cycle 4 in the basis matrix is a fundamental requirement.

Finally, a cycle elimination masking algorithm based on ACE value is proposed to mask the replaced basis matrix $\boldsymbol{B}_{\text{replace}}$ and continue to eliminate cycles. Masking a position means all cycles related to it are removed and the ACE value of cycles that the other positions in the same row or same column participated in are decreasing. Therefore. It is crucial to choose a rational place to eliminate cycles with smaller ACE value. Two cases are going to be discussed.

When set $\eta = 0$ for cycle g. In this condition, we need to keep cycle $g+2$ with larger ACE value. Find the biggest value in $\boldsymbol{D}^g$ by row and save its position. If there are several candidate positions, compare their ACE minimum values and ACE average values of cycle $g+2$. Choose a place whose ACE minimum value is the smallest at first. Then choose the smallest ACE average value if all the above conditions are the same. Stop masking cycle g and $g = g + 2$ until it satisfies condition $N^g = 0$. When $g > g_{\text{end}}$, output $\boldsymbol{B}_{\text{mask}}$

When set $0 < \eta < 1$ for cycle g. We need to selectively keep cycle g with ACE value as large as possible because cycle g still exists in the end. The basic selection principle is as same as in the first case. Stop reducing cycle g until it satisfies condition $N^g = \eta N^g_{\text{init}}$ and $g = g + 2$. When $g > g_{\text{end}}$, output $\boldsymbol{B}_{\text{mask}}$ in the end. See Algorithm 2 for details:

A few examples are given below to explain validity of algorithms.

E.g.1: Let α be a primitive element in a finite field GF(16). Let $\boldsymbol{S}_1 = \{\alpha^1, \alpha^2, \alpha^5\}$ and $\boldsymbol{S}_2 = \{\alpha^6, \alpha^7, \alpha^9, \alpha^{10}, \alpha^{12}, \alpha^{14}\}$ be two arbitrary subsets of GF(16). Set $\gamma = 1$. We get a 3×6 base matrix over GF(16) in the form given by (1).

Algorithm 1. A Replacement Preprocessing Algorithm

Input: $\boldsymbol{B}$, r, maxtime
Output: $\boldsymbol{B}_{\text{replace}}$
Set $\boldsymbol{B}_{\text{replace}} = \boldsymbol{B}$;
for $t = 1$ to maxtime **do**
  Calculate $\boldsymbol{D}^g$ and N^g_{init}. Find the maximum in $\boldsymbol{D}^g$ by global searching and save its row and column positions as $Index$
  for $k = 1$ to length($Index$) **do**
    for $z = 1$ to p **do**
      Let $\boldsymbol{B}_{\text{replace}}(Index(k)) = z$. Calculate $\boldsymbol{D}^g$ and $\boldsymbol{D}^{g'}$ of the new $\boldsymbol{B}_{\text{replace}}$
      if $N^{g'} = 0$ && $\min(N^g)$ **then**
        Record all alternative values z in Z
      end if
    end for
    if ~isempty(Z) **then**
      If there are multiple alternative values, select v at random and set $\boldsymbol{B}_{\text{replace}}(Index(k)) = q$
      break
    else
      continue
    end if
  end for
end for
output $\boldsymbol{B}_{\text{replace}}$

Set $\eta = 0$ of cycle 6. Algorithm 1 and 2 are used to generate a masked base matrix $\boldsymbol{B}_{1,\text{mask}}$. The parity-check H is generated by extending the masked base matrix $\boldsymbol{B}_{1,\text{mask}}$ with 16-ary CPMs and ZMs of size 15×15. It is a NB-QC-LDPC code over GF(16) whose length is 90 symbols and rate is 0.5. The structure of $\boldsymbol{B}_{1,\text{mask}}$ is as follows:

$$\boldsymbol{B}_{1,\text{mask}} = \begin{bmatrix} \alpha^{11} & \alpha^{14} & 0 & \alpha^{8} & \alpha^{13} & 0 \\ 0 & \alpha^{12} & \alpha^{11} & \alpha^{4} & \alpha^{7} & \alpha^{13} \\ \alpha^{9} & \alpha^{13} & \alpha^{6} & 0 & \alpha^{14} & 0 \end{bmatrix}$$

For comparison, another masked base matrix $\boldsymbol{B}_{2,mask}$ of NB-QC-LDPC code in GF(16) is generated with girth is at least 10. The structure of $\boldsymbol{B}_{2,mask}$ is as follows:

$$\boldsymbol{B}_{2,\text{mask}} = \begin{bmatrix} \alpha^{11} & \alpha^{14} & 0 & \alpha^{8} & 0 & 0 \\ 0 & 0 & \alpha^{11} & \alpha^{4} & \alpha^{7} & \alpha^{13} \\ \alpha^{9} & \alpha^{13} & \alpha^{6} & 0 & \alpha^{14} & \alpha^{12} \end{bmatrix}$$

E.g.2: Let α be a primitive element over a finite field GF(64). Let $S_1 = \{\alpha^7, \alpha^8, \alpha^9, \alpha^{10}\}$ and $S_2 = \{\alpha^{53}, \alpha^{54}, \alpha^{55}, \alpha^{56}, \alpha^{57}, \alpha^{58}, \alpha^{59}, \alpha^{60}\}$ be two arbitrary subsets of GF(64). Set a $\gamma = 1$. The base matrix $\boldsymbol{B}_1$ is given by (1).

Set $\eta = 0$ for cycle 6 and $\eta = 0.2$ for cycle 8. A masked base matrix $\boldsymbol{B}_{3,\text{mask}}$ that is generated by Algorithm 1 and 2 has a handle of cycle 8 with larger ACE

Algorithm 2. A Cycle Elimination Masking Algorithm based on ACE Value

Input: $\boldsymbol{B}_{\text{replace}}$, g_{end}, η
Output: $\boldsymbol{B}_{\text{mask}}$
1: Set $\boldsymbol{B}_{\text{mask}} = \boldsymbol{B}_{\text{replace}}$, $[m, n] = \text{size}(\boldsymbol{B}_{\text{replace}})$;
2: Based on η, set $g^* = \begin{cases} g+2 & \eta = 0 \\ g & 0 < \eta < 1 \end{cases}$
3: Calculate N^g_{init} of $\boldsymbol{B}_{\text{mask}}$.
4: **while** $(g < g_{\text{end}})$ **do**
5: Calculate $\boldsymbol{D}^g$ and N^g of $\boldsymbol{B}_{\text{mask}}$
6: **for** $j = 1$ to m **do**
7: Find the maximum in $\boldsymbol{D}^g$ and save its position as *col*
8: **if** length(*col*) ≥ 1 **then**
9: Calculate the ACE minimum value and average value of cycle g^* in which the nonzero elements in the same row or column participate. Choose a position s where the former value is minimum at first, then the last one is minimum
10: **else**
11: continue
12: **end if**
13: Set $\boldsymbol{B}_{\text{mask}}(j, s) = 0$. Calculate $\boldsymbol{D}^g$ and N^g for the next round
14: **if** $N^g \leqslant \eta N^g_{\text{init}}$ **then**
15: $g = g + 2$
16: Set new η and g^*
17: **end if**
18: **end for**
19: **end while**
20: output $\boldsymbol{B}_{\text{mask}}$

average value. The $\boldsymbol{H}$ is generated by extending $\boldsymbol{B}_{3,\text{mask}}$ with 64-ary CPMs and ZMs of size 63×63, whose code length is 504 symbols and rate is 0.5.

$$\boldsymbol{B}_{3,\text{mask}} = \begin{bmatrix} \alpha^{37} & 0 & \alpha^{15} & \alpha^{45} & 0 & 0 & \alpha^{21} & 0 \\ 0 & \alpha^{38} & 0 & 0 & \alpha^{46} & \alpha^{30} & \alpha^{61} & 0 \\ 0 & \alpha^{18} & \alpha^{39} & \alpha^{26} & 0 & 0 & \alpha^{31} & \alpha^{62} \\ \alpha^{49} & \alpha^{47} & \alpha^{19} & \alpha^{40} & \alpha^{27} & \alpha^{18} & 0 & \alpha^{32} \end{bmatrix}$$

For comparison, the base matrix $\boldsymbol{B}_{4,\text{mask}}$ is generated with girth is at least 10. The structure of $\boldsymbol{B}_{4,\text{mask}}$ is as follows:

$$\boldsymbol{B}_{4,\text{mask}} = \begin{bmatrix} \alpha^{37} & 0 & \alpha^{15} & 0 & \alpha^{29} & \alpha^{60} & \alpha^{21} & \alpha^{58} \\ 0 & \alpha^{38} & 0 & \alpha^{16} & 0 & \alpha^{30} & \alpha^{61} & \alpha^{22} \\ \alpha^{46} & 0 & \alpha^{39} & \alpha^{26} & \alpha^{17} & 0 & \alpha^{31} & 0 \\ 0 & \alpha^{47} & \alpha^{19} & \alpha^{40} & 0 & \alpha^{18} & 0 & \alpha^{32} \end{bmatrix}$$

At present, most of works focus on constructing binary QC-LDPC codes by eliminating short cycles and maximum girth of the Tanner graph [13,15], which are unilateral if just consider a single factor. In [8], a class of LDPC codes

proposed by using different combinations of the scyclotomic cosets in a finite field performed better than some EG-LDPC codes. In [5], a very large class of q-ary CPM QC-LDPC codes were constructed based on two arbitrary subsets of a finite field. Then, through masking by a masking matrix $\boldsymbol{M}$, the masked base matrix had a good cycle distribution. The result demonstrated a phenomenon that a larger girth does not necessary to a better error performance. Thus, not only the number of short cycles, but also the connectivity may influence the error correction performance.

The number and The ACE average value of cycles in NB-QC-LDPC code under different code lengths and finite fields are mentioned in Table 1.

Table 1. The number and The ACE average value of cycles in NB-QC-LDPC Code under different code lengths and finite fields

Type	Source	Number of cycle 4	Number of cycle 6	Number of cycle 8	ACE value					ACE value average
					1	2	3	4	5	
GF(16) (90,45)	$\boldsymbol{B}_{1,\text{mask}}$	0	0	180	0	3	4	5	0	3.167
	$\boldsymbol{B}_{2,\text{mask}}$	0	0	0	0	0	0	0	0	0
	In [8]	0	0	150	4	4	2	0	0	1.800
GF(64) (504,252)	$\boldsymbol{B}_{3,\text{mask}}$	0	0	189	0	0	1	2	0	3.667
	$\boldsymbol{B}_{4,\text{mask}}$	0	0	0	0	0	0	0	0	0
	In [5]	0	0	252	0	1	1	2	0	3.250

Since the base matrix $\boldsymbol{B}_{1,\text{mask}}$ satisfies the RC-constraints when $r = 15$, it has girth at least 6. From Table 1, it also has the number of cycles of length 8 is 180 with ACE average value is 3.167. A masked base matrix $\boldsymbol{B}_{2,\text{mask}}$ over GF(16) is constructed for comparation, whose Tanner graph has girth at least 10. The codeword in [8] has girth at 6 and the number of cycles of length 8 is 150 with ACE average value is 1.800. It could be seen that although the number of cycles 8 has increased, the ACE average value is higher.

Similarly, the masked base matrix $\boldsymbol{B}_{3,\text{mask}}$ satisfies the RC-constraints when $r = 63$. The length of cycle is at least 8. Table 1 illustrates that it has the number of cycles 8 is 189 with ACE average value is 3.667. A masked base matrix $\boldsymbol{B}_{4,\text{mask}}$ over GF(64) is constructed for comparation, whose Tanner graph has girth at least 10. As for [5], it has some cycles 6 with ACE average value is 3.250. It could be seen that those methods in this paper make a reasonable compromise between the number and the connectivity of cycles.

4 Result and Discussion

The block error performance of NB-QC-LDPC codes are simulated in BPSK modulation and AWGN channel, which are decoded with 50 iterations of the FFT-QSPA in Fig. 2 and Fig. 3.

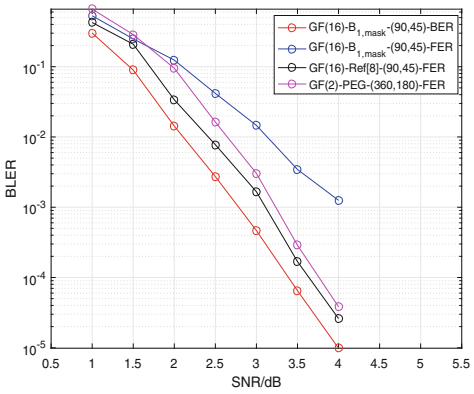

Fig. 2. Block error performance of the 16-ary (90,45) code.

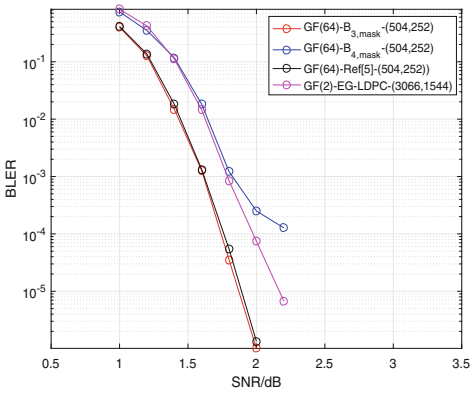

Fig. 3. Block error performance of the 64-ary (504,252) code.

Shown in Fig. 2 the block error performance of the codeword constructed in this paper is the best. For the codeword with code length of 90 symbols and rate of 0.5, the block error performance of codeword constructed by $\boldsymbol{B}_{1,\mathrm{mask}}$ is far superior to the codeword constructed by $\boldsymbol{B}_{2,\mathrm{mask}}$, which means that blindly increasing girth does not necessarily to bring performance improvement. Then, the performance of codeword constructed by $\boldsymbol{B}_{1,\mathrm{mask}}$ is about 0.4 dB better than that of codeword in [8] when BLER is 10^{-4}, which shows the superiority of LDPC

codes under larger ACE values. It is also about 0.5 dB better than the binary LDPC code [8] when BLER is 10^{-4}. This phenomenon shows that NB-LDPC codes have stronger error correction capabilities than binary LDPC codes.

In addition, Fig. 3 illustrates that for code length of 504 symbols and rate of 0.5, the block error performance of codeword constructed by $\boldsymbol{B}_{3,\text{mask}}$ in this paper is much better than that of codeword constructed by $\boldsymbol{B}_{4,\text{mask}}$ which exists error floor when BLER is 10^{-4}. This fully demonstrates the feasibility of improving performance by eliminating poorly connected cycles. Meanwhile, the performance of codeword constructed by $\boldsymbol{B}_{3,\text{mask}}$ is a little better than that of the codeword in [5] when BLER is 10^{-4} because both codewords have good ACE values, but the constructed codeword is larger. Similarly, the comparison with binary EG-LDPC codes in [9] also illustrates the superiority of NB-LDPC codes.

5 Conclusion

In this paper, we presented a simple and flexible masking method for constructing irregular NB-QC-LDPC codes. The proposed method aimed to ensure that the left cycles in a masked base matrix have larger ACE average value under the constraint of the number of cycles. Examples in different range of lengths and finite fields were presented to illustrate the advantage of this feature. Simulation results showed that the codewords constructed in paper have better block error performance.

References

1. Davey, M.C., MacKay, D.J.C.: Low density parity check codes over gf(q). In: 1998 Information Theory Workshop (Cat. No.98EX131), pp. 70–71 (1998)
2. Gallager, R.: Low-density parity-check codes. IRE Trans. Inf. Theory **8**(1), 21–28 (1962)
3. Han, G., Guan, Y.L., Kong, L.: Construction of irregular QC-LDPC codes via masking with ace optimization. IEEE Commun. Lett. **18**(2), 348–351 (2014)
4. Li, J., Liu, K., Lin, S., Abdel-Ghaffar, K.: Algebraic quasi-cyclic LDPC codes: construction, low error-floor, large girth and a reduced-complexity decoding scheme. IEEE Trans. Commun. **62**(8), 2626–2637 (2014)
5. Li, J., Liu, K., Lin, S., Abdel-Ghaffar, K.: A matrix-theoretic approach to the construction of non-binary quasi-cyclic LDPC codes. IEEE Trans. Commun. **63**(4), 1057–1068 (2015)
6. Liu, K., Huang, Q., Lin, S., Abdel-Ghaffar, K.: Quasi-cyclic LDPC codes: construction and rank analysis of their parity-check matrices. In: 2012 Information Theory and Applications Workshop, pp. 227–233 (2012)
7. Liu, L., Huang, J., Zhou, W., Zhou, S.: Computing the minimum distance of non-binary LDPC codes. IEEE Trans. Commun. **60**(7), 1753–1758 (2012)
8. Lu, M., Zhang, L.: Constructions of irregular nonbinary QC-LDPC codes: cyclotomic coset approach. In: 2012 IEEE 11th International Conference on Signal Processing, vol. 2, pp. 1468–1472 (2012)
9. Ryan, W., Shu, L.: Channel codes classical and modern (2009)

10. Tao, X., Feng, D., Zhang, Y., Huang, A.: Construction of non-binary LDPC codes with very large girth. In: 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4 (2012)
11. Tian, T., Jones, C.R., Villasenor, J.D., Wesel, R.D.: Selective avoidance of cycles in irregular LDPC code construction. IEEE Trans. Commun. **52**(8), 1242–1247 (2004)
12. Vukobratovic, D., Djurendic, A., Senk, V.: Ace spectrum of LDPC codes and generalized ace design. In: 2007 IEEE International Conference on Communications, pp. 665–670 (2007)
13. Wang, D., Wang, L., Chen, X., Fei, A., Ju, C., Wang, Z.: Construction of QC-LDPC codes based on pre-masking and local optimal searching. IEEE Commun. Lett. **22**(6), 1148–1151 (2018)
14. Hu, X.-Y., Eleftheriou, E., Arnold, D.: Irregular progressive edge-growth (peg) tanner graphs. In: Proceedings IEEE International Symposium on Information Theory, p. 480 (2002)
15. Xu, H., Feng, D., Luo, R., Bai, B.: Construction of quasi-cyclic LDPC codes via masking with successive cycle elimination. IEEE Commun. Lett. **20**(12), 2370–2373 (2016)
16. Xu, J., Chen, L., Djurdjevic, I., Lin, S., Abdel-Ghaffar, K.: Construction of regular and irregular LDPC codes: geometry decomposition and masking. IEEE Trans. Inf. Theory **53**(1), 121–134 (2007)
17. Yang, F., Wang, L., Wang, D., Chen, X., Cui, M.: Construction of irregular QC-LDPC codes via arbitrary degree distribution masking algorithm. In: 2018 Asia Communications and Photonics Conference (ACP), pp. 1–3 (2018)
18. Zeng, L., Lan, L., Tai, Y.Y., Song, S., Lin, S., Abdel-Ghaffar, K.: Transactions papers - constructions of nonbinary quasi-cyclic LDPC codes: a finite field approach. IEEE Trans. Commun. **56**(4), 545–554 (2008)
19. Zhou, B., Kang, J., Tai, Y.Y., Lin, S., Ding, Z.: High performance non-binary quasi-cyclic LDPC codes on Euclidean geometries LDPC codes on Euclidean geometries. IEEE Trans. Commun. **57**(5), 1298–1311 (2009)