Improved PEG construction of large girth QC-LDPC codes

Madiagne Diouf, David Declercq, Marc Fossorier
ETIS, ENSEA/Univ. Cergy-Pontoise/CNRS UMR-8051
95014 Cergy-Pontoise Cedex, France
{madiagne.diouf, declercq}@ensea.fr

Samuel Ouya
LIRT, ESP/UCAD
Dakar, Senegal
samuel.ouya@gmail.com

Bane Vasić
Dept. of ECE, Univ. of Arizona
Tucson, AZ, USA
vasic@ece.arizona.edu

Abstract— In this paper, we present an improvement of the PEG algorithm for constructing quasi-cyclic low-density parity-check (QC-LDPC) codes with large girth. We introduce the concept of PEG-undetectable cycles on the computation tree in the PEG algorithm for QC-LDPC codes and give a predictive method to avoid these undetected cycles. The aim is to select only the candidates that ensure the maximization of local girth for a code of girth $g \geq 10$ and thus avoid $a$ posteriori verification after the creation of a new edge i.e. keep the predictive philosophy of the PEG algorithm. The proposed method is applicable to both regular and irregular codes and also protograph type-I codes. Simulation results are presented to demonstrate the efficiency of our method in terms of minimum circulant permutation matrix $p_{	ext{min}}$ and error performance.

Index Terms—Low Density Parity Check (LDPC) codes, Progressive Edge-Growth (PEG), quasi-cyclic (QC), Circulant Permutation Matrix (CPM), girth.

I. INTRODUCTION

Quasi-Cyclic LDPC (QC-LDPC) codes are a special class of constrained LDPC codes. They require a smaller amount of memory for storing their parity-check matrices, and their structural properties are easier to analyze than those of random LDPC codes. Indeed, a QC-LDPC code is defined by a parity-check matrix which is organised into sub-matrices based on circulant permutation matrices (CPM). Recently, several methods for constructing QC-LDPC codes have been proposed. Most of these methods have for main objective to avoid short cycles [2]–[5] which are responsible of a performance loss for under iterative sum-product decoding [1]. However many of the proposed methods are more or less limited by the code lengths. Wang et al. [4] proposed an hill-climbing search algorithm and found the best minimum size of circulant permutations matrices for girths $g = 8$ and $g = 10$. The girth, $g$ of a code is the length of the shortest cycle in the code graph and Fossorier showed in [2] that the girth of any QC-LDPC code is at most 12.

A very efficient method for constructing Tanner graphs having a large girth is by using the Progressive-Edge-Growth (PEG) algorithm [7]. In this paper, we use the PEG algorithm and focus on the problem of optimizing the girth of QC-LDPC codes i.e. finding QC-LDPC codes with large girth and smallest size of circulant permutations matrices. We introduce

Bane Vasić acknowledges support off NSF under grants CCF-1314147 and ECCS-1500170.
the computation tree within depth $l$, i.e. for which there exist a path formed with $2l$ edges from the root node $v_n$ to the elements of $\mathcal{N}^l_{v_n}$. We also define the complementary set $\overline{\mathcal{N}}^l_{v_n}$, such that $\mathcal{N}^l_{v_n} \cup \overline{\mathcal{N}}^l_{v_n} = C$, which contains all check nodes that do not appear in the computation tree within depth $l$. Check nodes $c_m$ belonging to $\overline{\mathcal{N}}^l_{v_n}$ could either be connected to the root $v_n$, through a path longer than $2l$ edges, or such that there is no path from $v_n$ to $c_m$ formed by the already allocated edges. Figure 1 shows a computation tree expanded from $v_n$ within depth $l = k$ where the symbols $\circ$ represent the variable nodes and the symbols $\square$ represent check nodes. The black squares are the ones belonging to $\overline{\mathcal{N}}_{v_n}$.

B. Quasi-Cyclic Low Density Parity Check Based on Progressive Edge Growth

QC-LDPC are featured in a variety of communications system standards, such as DVB-S2, IEEE 802.16e. These codes have a parity-check matrix which is organised into submatrices defined as circulant permutation matrices (CPM) [2].

The parity-check matrix $H$ of a $(d_v, d_c)$-regular QC-LDPC code of length $N = p \cdot d_c$ can be represented by [2], [4],

$$H = \begin{bmatrix}
I(0) & I(0) & \cdots & I(0) \\
I(0) & I(p_{1,1}) & \cdots & I(p_{1,d_c-1}) \\
\vdots & \vdots & \ddots & \vdots \\
I(0) & I(p_{d_v-1,1}) & \cdots & I(p_{d_v-1,d_c-1})
\end{bmatrix}$$

(1)

where $p$ is called the lifting factor and $I(p_{j,i})$ represents the $p \times p$ circulant permutation matrix whose rows are obtained by cyclically shifting the rows of the identity matrix to the left by $p_{j,i}$ (i.e. a one at row-$(c + p_{j,i})$ mod $p$ for column-$c$, $0 \leq c \leq p - 1$ and zero elsewhere). Clearly, $I(p_{j,i})$ is the matrix of the subgraph formed by the group of $p$ variable nodes, $v_{ip}, \ldots, v_{ip+1}$ and the group of $p$ check nodes $c_{jp}, \ldots, c_{j(p+1)}$. $I(p_{j,i})$ is also called block in the following. Note that by convention, we fix $p_{0,0} = p_{j,0} = 0 \forall i,j$, and $I(0)$ is the identity matrix.

A $2l$-cycle or cycle of length $2l$ in $G$ is a non-empty alternating sequence $v_{n_0}, c_{m_0}, v_{n_1}, c_{m_1}, \ldots, v_{n_l}$ in $G$ such that $v_{n_0} = v_n$, and all the other nodes are distinct. Two consecutive positions in this sequence define also an edge, which is located in one of the circulant blocks of $H$. A cycle of length $2l$ can further be associated with an ordered series of circulant permutation matrices:

$$I(p_{j_0,i_0}), I(p_{j_1,i_1}), I(p_{j_2,i_2}), \ldots, I(p_{j_{l-1},i_{l-1}}), I(p_{j_{l-1},i_0})$$

(2)

with $j_{l-1} \neq j_0$, $i_{l-1} \neq i_0$ and for $1 \leq x \leq l - 1$, $j_x \neq j_{x-1}$, $i_x \neq i_{x-1}$.

The edges forming a $2l$-cycle belong to the circulant blocks indicated in this series, but the $I(p_{j,i})$ are not necessarily distincts, since a cycle can traverse a given block two times or more, as long as it uses distinct bit positions in $I(p_{j,i})$.

Using those notations, a necessary and sufficient condition for the existence of the $2l$-cycle is [2]:

$$\sum_{x=0}^{l-1} \left(p_{j_x,i_x} - p_{j_{x+1},i_x}\right) = 0 \mod p$$

(3)

This condition depends only on the circulant shifts values $p_{j,i}$, and to build a QC-LDPC code with girth $g$, we need to choose those values such that:

$$\sum_{x=0}^{l-1} \left(p_{j_x,i_x} - p_{j_{x+1},i_x}\right) \neq 0 \mod p$$

(4)

for all $l, 2 \leq l \leq g/2 - 1$.

We can see that the larger $p$ is, the more degrees of freedom one gets to satisfy the set of equations (4). However, finding the minimum circulant size $p_{\min}$ such that (4) is satisfied is not a trivial problem, as stated in [2], [10] where lower bounds on the value of $p_{\min}$ have been either proved or conjectured, or in [4], where elaborate search algorithms were proposed to approach these lower bounds.

One of the proposed solutions to obtain QC-LDPC with minimized circulant sizes is to rely on the PEG algorithm [7], adapted to a quasi-cyclic design, as introduced by Li and Kumar in [5]. First, the set of all variable nodes and check nodes is divided into groups of $p$ consecutive nodes, and the PEG is modified such as to assign the edges to the Tanner graph "group by group" instead of "node by node".

For each variable node group, the computation tree is expanded only from the first variable node $v_n=ip$ in the current group, and once an check node candidate $c_m$ has been chosen, not only the edge between $v_n$ and $c_m$ is created, but also the rest of the edges in the circulant block, corresponding to nodes $v_{ip+1}$ to $v_{i(p+1)p-1}$. Then, the same procedure is applied to the next circulant block, iteratively until all the blocks in $H$ are filled with edges. The quasi-cyclic PEG algorithm is described in Algorithm 1, where we followed the notations used in [5].

In Step-1.a the PEG assigns the first edge of the current variable node $v_n$ randomly amongst the check nodes of lowest degree. In Step-2.a the other edges of the variable node $v_n$ are assigned after expansion of the computation tree. The new edge connects the root node $v_n$ to a candidate check node $c_m$ in $\overline{\mathcal{N}}_{v_n}$. Step-1.b and Step-2.b represent the creation of the rest of circulant blocks once the first edge has been selected. If during Step-2.a there are no candidates, i.e. $\overline{\mathcal{N}}_{v_n} = \emptyset$, we
Input: $p, d_l, d_u$
Output: $G$
for $i = 0$ to $d_u - 1$
do for $j = 0$ to $d_l - 1$
if $j = 0$
Step-1.a: Edge $(c_m, v_{ip}) \rightarrow E_{v_{ip}}^0$.
Step-1.b: for $p_1 = 1$ to $p - 1$
do
Edge $(c_{(m/p)}p + m d(m + 1, p), v_{ip + p}) \rightarrow E_{v_{ip + p}}^0$
end
end
Step-2.a: Expand a tree from variable node $v_{ip}$ up to depth $\ell$
under the current graph such that $\mathcal{X}_{v_{ip}}^\ell \neq \emptyset$ but $\mathcal{X}_{v_{ip}}^{\ell + 1} = \emptyset$ or
$|\mathcal{X}_{v_{ip}}^\ell| = |\mathcal{X}_{v_{ip}}^{\ell + 1}|$. If $\mathcal{X}_{v_{ip}}^\ell \neq \emptyset$ then Edge
$(c_m, v_{ip}) \rightarrow E_{v_{ip}}^0$ and go to step 2.b; otherwise go to step 2.c.
Step-2.b: for $p_1 = 1$ to $p - 1$
do
Edge $(c_{(m/p)}p + m d(m + 1, p), v_{ip + p + 1}) \rightarrow E_{v_{ip + p + 1}}^0$
end
break
Step-2.c: Delete $E_{v_{ip}}^0, \ldots, E_{v_{ip + p - 1}}^0$, go to step 1.a
choose $c_m$ as another check node which has the lowest check
degree or second lowest check degree.
end

Algorithm 1: QC-PEG Algorithm

delete all edges of nodes $v_{ip}$ to $v_{(i+1)p-1}$ and try again with
another first assignment in Step-1.a.

Improvements of the PEG algorithm based on the computation
tree have been proposed, for example to minimize the cycles
multiplicity [8], to introduce the ACE constraint for
irregular codes [9], or to avoid trapping sets build from several
cycles [6].

III. CONDITIONS TO AVOID PEG-UNDetectABLE CYCLES

A. Notion of an PEG-undetectable cycles

In the PEG algorithm, check nodes $c_m$ in $\mathcal{X}_{v_{ip}}^\ell$ are chosen
such that they do not create cycles of length equal or less
than than a target girth $2l = g$. In the original PEG, this
property is ensured by the definition of the computation tree
expanded until depth $\ell$, but in the QC-PEG this property is
no longer guaranteed. To solve this design problem, QC-
PEG improvements have been proposed in the literature, using
a backtracking stage [5], [6]. After a circulant assignment,
an additional search procedure checks whether the can-
dicate location $c_m$ created unwanted cycles, and removes the
assigned circulant to choose another one in case one or
several unwanted cycles exist. Using backtracking leads to
valid QC-LDPC codes designs, but does not take advantage
of the predictive nature of the PEG algorithm. Being able to
predict the cycles that will be indeed created after a circulant
assignment has many advantages. The first one is to save
computation time as backtracking could be computationally
intensive when $p$ is large or when the target girth $g$ is large.
Another advantage is that using a predictive approach, we can
also control the number of cycles of length $g$ that are created
by computing the number of occurrences of each check node,
and therefore minimize their number in order to build better
QC-LDPC codes.

To maintain the predictive feature of the PEG algorithm, we
propose a method to characterize cycles that are not detected
in the computation tree. We start by a formal definition and
categorization.

Definition 1: A 2l-cycle is said PEG-undetectable, if it is
created after the choice of a candidate $c_m$ in $\mathcal{X}_{v_{ip}}^\ell$ in the QC-
PEG algorithm.

The following lemma allows characterizing PEG-
undetectable cycle: .

Characterization 1: Let $v_{n=ip}$ be the current variable node
in the PEG, and let $I(p_{j,i})$ be the newly assigned circulant
block. Any cycle which traverses $I(p_{j,i})$ at least twice is an
PEG-undetectable cycle.

Before we present the details of the characterization for
PEG-undetectable 8-cycles and 10-cycles, let us finish this
section by stating that 4-cycles and 6-cycles cannot be PEG-
undetectable. Indeed, with the definitions of [2], all cycles
of length 4 are formed by 4 blocks with two distinct row indices
and 2 distinct columns indices $(j_0 \neq j_1$ and $i_0 \neq i_1)$, and all 6
cycles are formed by 6 blocks belonging to a submatrix with 3
distincts row indices and 3 distinct column indices $(j_x \neq j_{x+1}$
and $i_x \neq i_{x+1}$ for $0 \leq x \leq 2$ with $j_0 = j_3$ and $i_0 = i_3$).

B. Characterization of PEG-undetectable 8-cycles

A necessary and sufficient condition to have a 8-cycle in $H$
is:

$$\sum_{x=0}^{3} (p_{j_x,i_x} - p_{j_{x+1},i_x}) = 0 \mod p \quad (5)$$

where all $j_x \neq j_{x+1}$, and all $i_x \neq i_{x+1}$.

However, we can have $j_0 = j_2$, $j_1 = j_3$, $i_0 = i_2$, or
$i_1 = i_3$, while still verifying equation (5). Assuming that the
newest assigned block is $I(p_{j_3,i_3})$, the cases $j_3 = j_1$ and
$i_3 = i_1$ are the important ones for our purpose, and provide
four scenarios of occurrence of an PEG-undetectable 8-cycle.
The four scenarios are described on the figure 2, where $p_{j,i}^{\ast}$
represents the circulant shift associated with a block which is
traversed $\alpha$ times by the considered cycle. We describe the
different configurations by compact groups of $r \times q$ blocks,
when the cycle series (2) is composed of r distinct row indices
and q distinct column indices.

Figure 2(a) corresponds to 8-cycle formed by $2 \times 2$ blocks,
in which case all blocks which form the 8-cycle are traversed
twice. Figure 2(b) corresponds to 8-cycle formed by $2 \times 3$
blocks, where the two blocks in the last column are traversed
twice. Figure 2(c) corresponds to 8-cycle formed by $3 \times 2$
blocks and is the transposed case of Figure 2(b). Finally, figure
2(d) corresponds to 8-cycle formed by $3 \times 3$ blocks where two
block are not traversed and only the last block is traversed
twice.

Putting the cases illustrated on figure 2 into equations
involving the circulant shifts, we get the following theorem,
which characterizes the constraint that we need to incorporate
in the PEG algorithm in order to avoid the unwanted creation
of PEG-undetectable 8-cycles.
Thus to avoid PEG-undetectable 8-cycles formed by $2 \times 3$ blocks, it is necessary and sufficient to have:

$$2(p_{j_1,i_1} - p_{j_0,i_0}) + p_{j_0,i_0} - p_{j_1,i_0} + p_{j_0,i_2} - p_{j_1,i_2} \neq 0 \mod p$$

$$\Rightarrow 2(p_{j_1,i_1} - p_{j_0,i_1}) + p_{j_0,i_0} - p_{j_1,i_0} + p_{j_0,i_2} - p_{j_1,i_2} \neq \lambda p$$

$$\Rightarrow p_{j_1,i_1} \neq \lambda p - 2p_{j_0,i_0} + p_{j_1,i_0} + 2p_{j_0,i_2} - p_{j_1,i_2} \mod p$$

where $\lambda \in \{0, 1, 2, 3\}$.

Proofs for the third and fourth inequality are similar to that of Case 1 and Case 2, and we do not report them in this paper for lack of space.

In order to ensure that the QC-PEG algorithm will not create PEG-undetectable 8-cycles, one simply needs to restrict the check node candidates $c_m$ in $\mathcal{N}_{v_{i,p}}$ to those for which the indexes $m = j_1 p + p_{j_1,i_1}$ verify the set of equations (6). By doing so, we are able to build a QC-LDPC code with target girth $g = 10$ and given circulant size $p$ without a posteriori verification.

C. Characterization of PEG-undetectable 10-cycles

For PEG-undetectable 10-cycles, we follow the same approach as for the 8-cycles. Equation (5) becomes

$$\sum_{x=0}^{4} (p_{j_x,i_x} - p_{j_{x+1},i_{x+1}}) = 0 \mod p$$  (7)

An important difference with 8-cycles is that not all $r \times q$ configurations can give raise to a length 10-cycle. In [2] it is shown that all cycles formed by $2 \times q$ blocks or $r \times 2$ blocks have lengths that are multiple of 4. Hence, 10-cycles are necessarily formed by $r \times q$ blocks where $3 \leq \{r, q\} \leq 5$. For $r = 5$ or $q = 5$, no block is traversed twice, and the only remaining possibility are formed by $r \times q$ blocks where $3 \leq \{r, q\} \leq 4$.

These possibilities provide eight scenarios of occurrence of an PEG-undetectable 10-cycle. We give the necessary and sufficient condition to avoid an PEG-undetectable 10-cycles with the following theorem.

Theorem 2: A necessary and sufficient condition to avoid PEG-undetectable 10-cycle in Step-2.a of the QC-PEG algorithm is:

$$p_{j_1,i_1} \neq \lambda p - 2p_{j_0,i_0} + p_{j_1,i_0} + 2p_{j_0,i_2} - p_{j_1,i_2} \mod p$$

where $\lambda \in \{0, 1, 2, 3, 4\}$, $j_x \neq j_y$, $i_x \neq i_y$ for $x \neq y$ and the check node picked in $\mathcal{N}_{v_{i,p}}$ is $c_m = j_1 p + p_{j_1,i_1}$.

The proof is analogous to that of Theorem 1.
By restricting the candidates in $\mathcal{N}_{v_n}$ following both equations (6) and 8, with $m = j_{ip} + p_{j_{ip}}$, we can then build QC-LDPC codes with girth $g = 12$, without backtracking or a posteriori verification.

IV. PROPERTIES AND PERFORMANCE DESIGNED QC-LDPC CODES

Although Theorems 1 and 2 are general and applicable to regular or irregular QC-LDPC codes, we restrict the design examples in this paper to the case of regular $(d_v = 3, d_c)$ codes for which the matrix $H_B$ is fully filled with non-zero circulants.

We first compare the minimum size of the circulant matrices, $p_{\min}$, obtained by our QC-PEG algorithm and other construction methods in the literature. The minimum cycle distribution of QC-PEG algorithm improves $p_{\min}$ as shown in Table I. Furthermore, we can see our QC-PEG algorithm give $p_{\min}$, that achieves the minimum bound defined by Karimi and Banihashemi in [10] for $4 \leq d_c \leq 6$.

![Table I: Minimum circulant size $p_{\min}$, for $(3,d_c)$-regular QC-LDPC codes with girth $g = 10$](https://example.com/table1.png)

Due to very large size of circulant matrices of girth $g = 12$ QC-LDPC codes, the hill-climbing algorithm is no longer effective. Algebraic designs of $g = 12$ QC-LDPC codes have been proposed in [3], for which we report the values of $p_{\min}$. As we can see, our improved QC-PEG algorithm gives shorter circulant sizes than the state-of-the-art.

![Table II: Minimum circulant size $p_{\min}$, for $(3,d_c)$-regular QC-LDPC codes with girth $g = 12$](https://example.com/table2.png)

In Table III, we compare the statistics of regular QC-LDPC codes constructed with the classical QC-PEG algorithm and with our improved QC-PEG. We can see that the use of the PEG-undetectable 8-cycles condition in the QC-PEG algorithm allows a better minimization of short cycles.

Finally, we show the impact of removing PEG-undetectable cycles on the decoding performance. All simulation results are presented for BPSK modulation over the AWGN channel, and the iterative decoding algorithm is the sum-product-algorithm with maximum number of iterations equal to 50. Fig. 3 shows the performance on the codes designed in the table III, for $d_c = 6$ and $d_c = 9$. As expected, our codes perform better in the error floor region, due to the better cycle distribution of the codes.

V. CONCLUSION

In this paper, we introduced the notion of PEG-undetectable cycles for the QC-PEG algorithm. We proposed an efficient method to detect the PEG-undetectable cycles and avoid a posteriori tests through backtracking. Our designs show improvements in terms of minimum circulant size $p_{\min}$ and cycle distribution compared to other design methods from the literature.

REFERENCES


