Additive, Structural and Multiplicative Transformations for the Construction of Quasi-Cyclic LDPC matrices

Alban Derrien, Emmanuel Boutillon, Audrey Cerqueus

Additive, Structural and Multiplicative Transformations for the Construction of Quasi-Cyclic LDPC matrices.

Alban Derrien *, Emmanuel Boutillon *, *Senior Member, IEEE,* and Audrey Cerqueus †
*Université de Bretagne-Sud
Lab-STICC, UMR 6285 CNRS – Lorient, France
⟨firstname⟩.⟨lastname⟩@univ-ubs.fr
†Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F - 42023 Saint-Etienne France
audrey.cerqueus@emse.fr

***Abstract*—The construction of a Quasi-Cyclic Low Density Parity-Check (QC-LDPC) matrix is usually carried out in two steps. In the first step, a prototype matrix is defined according to certain criteria (size, girth, check and variable node degrees, etc.). The second step involves expansion of the prototype matrix. During this last phase, an integer value is assigned to each non-null position in the prototype matrix corresponding to the right-rotation of the identity matrix. The problem of determining these integer values is complex. State-of-the-art solutions use either some mathematical constructions to guarantee a given girth of the final QC-LDPC code or a random search of values until the target girth is satisfied. In this paper, we propose an alternative/complementary method that reduces the search space by defining large equivalence classes of topologically identical matrices through row and column permutations using additive, structural and multiplicative transformations. Selecting only a single element per equivalence class can reduce the search space by a few orders of magnitude. Then, we use the formalism of constraint programming to list the exhaustive sets of solutions for a given girth and a given expansion factor. An example is presented in all sections of the paper to illustrate the methodology.**

I. Introduction

Low Density Parity-Check (LDPC) codes are a family of error correction codes used in many communication standards (TV broadcasting [1], Wi-Fi [2] and next generation cellular networks [3] among others). LDPC codes were invented in the 1960s by Gallager [4]. Their success is due to the existence of a simple iterative decoding algorithm known as the Belief Propagation (BP) algorithm [5]. The BP decoding algorithm (and its simplified versions [6]) exchanges messages (belief on the value of a bit) in a bipartite graph composed of variable and parity-check nodes. LDPC codes were generalized by Davey and McKay in 1998 over a finite field arithmetic [7]. These codes are called Non-Binary LDPC (NB-LDPC) codes. One of the main advantages of NB-LDPC codes is that good codes can be constructed with the degree of the variable nodes set to 2 (each variable is thus connected to exactly two parity-check nodes). NB-LDPC code has been adopted in a recent CCSDS standard [8].

To obtain a hardware-friendly error control code, some structures can be imposed on its structure. In particular, memory conflicts that can appear in a high speed decoder can be resolved using Quasi-Cyclic-LDPC (QC-LDPC) matrices. This type of matrices is used in many communication standards. There is a large literature on the construction of QC-LDPC matrices, some of them based on mathematical properties of particular sets, others on random generation, then selection, and others on exhaustive search [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

The main contribution of this paper is to propose a two step method to construct QC-LDPC matrices. The first step consists of partitioning the search space into equivalence classes of large size by means of additive, structural and multiplicative transformations. Thus, testing only one element per class allows us to reduce by a few orders of magnitude the number of solutions to be explored. The second step consists of using constraint programming tools to enumerate all solutions when they exist. This method can be applied to any type of QC-LDPC code. To explain and illustrate the method, we use an example of the construction of good QC-LDPC matrices from a given prototype matrix. Note that some of the ideas of the paper were independently developed by Tasdighi et al. and recently published in [21]. More specifically, the idea of structural and multiplicative transformations was already mentioned in [19] and has been used to reduce the search space to find high-girth QC-LDPC matrices. But, as recognized by M. Tasdighi, the present approach is much more general than that previously proposed in [19].

The rest of the paper is organized as follows. Section II gives the background of QC-LDPC code construction. Section III presents an equivalence relation between QC-LDPC matrices. Section IV shows how to select a single element of each class. Section V presents a new equivalence relation based on multiplicative properties. Section VI gives the number of solutions of minimal girth for an $L = 3$ rows, $J = 6$ lines protograph with several expansion factors. Finally, section VII sets out the conclusions of this study.

II. Construction of QC-LDPC matrices

In this section, we review the construction of a family of LDPC matrices well suited for hardware implementation called Quasi-Cyclic LDPC matrices. Then, we discuss the conditions that have to be satisfied to obtain QC-LDPC matrices with good topological properties. Finally, we provide some notation to describe the permutations.

A. Definition of a QC-LDPC matrix

An LDPC code can be represented by a bipartite graph that contains two types of nodes: the variable nodes and the check nodes. Variable nodes (respectively check nodes) are only connected to check nodes (respectively variable nodes). In the case of a regular LDPC code, the number d_v of check nodes connected to a given variable node and the number d_c of variable nodes connected to a given check node are constant (d_v and d_c are called the variable node degree and the check node degree, respectively).

LDPC codes were generalized to NB-LDPC codes in 1998 by Mackay and Neal [22]. These authors show that good NB-LDPC codes can be constructed with a constant variable node degree $d_v = 2$. The Belief Propagation (BP) algorithm is equivalent to the Maximum A Posteriori (MAP) decoder if the graph is cycle free [23] and its performance is close to MAP decoder if the bipartite graph representing the code does not have small cycles. Thus, the main topological objectives, in the design of a bipartite graph, are, first, to maximize its girth g (i.e. the length of its minimal cycle), and second, to minimize its multiplicity (i.e. the number of cycles of length g) denoted by $\mathcal{M}(g)$

The bipartite graph can be represented by a matrix H, called the parity-check matrix, where each variable of the code is associated with a column of H and each parity-check is associated with a row. If an edge exists between a check node i and a variable node j, then $H(i,j)$ is not equal to zero, otherwise, $H(i,j) = 0$. In the case of a (d_v, d_c) regular bipartite graph, each row of H contains d_c non-zero values and each column of H contains d_v non zero values. Note that permuting the rows and the columns of H does not affect the topology of the associated graph (see section III).

A QC-LDPC matrix is constructed in two steps. First, a protograph $\mathbf{H}$ of size $J \times L$ is constructed. Then, the protograph matrix is expanded, or lifted, by a factor N to obtain a (JN, LN) matrix [24], [20]. The simplest protograph matrix for $d_v = 2$, $d_c = 4$ is the matrix $\mathbf{H}_2$ of size $J = 2$, $L = 4$ defined as

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{1}$$

Throughout this paper, we illustrate the proposed method using the example of QC-LDPC matrices with a rate 1/2 generated from the matrix $\mathbf{H}_3$ of size $J = 3$, $L = 6$, defined as

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{2}$$

This matrix is well suited for the NB-LDPC code, where $d_v = 2$ gives a good code [5]. Note that, for NB-LDPC code, we also need to optimize the edge values (i.e. value of non-null coefficients of H). This point is not discussed further here since the present study is focused on the topological properties of the graph associated with the expanded matrix. It can be noted that $\mathbf{H}_3$ is unique up to rows/columns permutation (see section III).

The lifting process generates a matrix of size (JN, LN) from $\mathbf{H}$ by replacing each 0 value in $\mathbf{H}$ by O^N, the (N, N) zero matrix, and each 1 value by an (N, N) matrix I_a^N, where a is the shift value associated with the non-null position, see (4). I_a^N is defined as

$$I_a^N = \begin{cases} I_a^N(i, i+a \mod N) = 1, & \forall i \in \{0, \ldots, N-1\} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Note that I_0^N represents the (N, N) identity matrix. The matrix I_a^N, $a \in \{0, \ldots, N-1\}$ is called a circulant matrix [25]. The index a is called the permutation shift. The QC-LDPC matrix H is thus fully described by the protograph matrix $\mathbf{H}$, the factor of expansion N and by the shift values associated with each non-zero value of the circulant permutation matrices [20].

For example, matrix H obtained from $\mathbf{H}_3$ is defined as

$$H = \begin{bmatrix} I_{a_1}^N & I_{b_1}^N & I_{c_1}^N & I_{d_1}^N & O^N & O^N \\ I_{a_2}^N & I_{b_2}^N & O^N & O^N & I_{e_2}^N & I_{f_2}^N \\ O^N & O^N & I_{c_3}^N & I_{d_3}^N & I_{e_3}^N & I_{f_3}^N \end{bmatrix} \tag{4}$$

Matrix H defines a type-I QC-NB-LDPC code since non zero values of the prototype matrix are expanded to a single circulant matrix (a prototype matrix with at least one double circulant matrix is called a type-II protograph [26]). Since the protograph is of type-I, it is convenient to represent the expanded matrix H as

$$\mathcal{H} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 & -1 & -1 \\ a_2 & b_2 & -1 & -1 & e_2 & f_2 \\ -1 & -1 & c_3 & d_3 & e_3 & f_3 \end{bmatrix}, \tag{5}$$

where $\mathcal{H}(i,j)$ represents $I_{\mathcal{H}(i,j)}^N$ if $\mathcal{H}(i,j)$ is a positive integer, or O^N if $\mathcal{H}(i,j) = -1$. In the sequel, -1 is replaced by a simple dot to simplify the notation.

B. Topological properties of QC-LDPC matrices

Due to the correspondence between the Tanner graph and the parity-check matrix, a cycle in the Tanner graph is equivalent to a cycle in the parity-check matrix. In the parity-check matrix, the cycle is composed of a circular consecutive sequence of horizontal and vertices vertices [4]. The construction objective in the lifting process is to maximize the girth of the Tanner graph.

Each cycle of length ℓ in the protograph lifts into one or several cycles of length ℓ' in the expanded graph, with ℓ' being a multiple of ℓ. The necessary and sufficient condition for a cycle of length ℓ to lift into cycles of length strictly greater than ℓ is given in [20]. Let N be the expansion size and C a cycle of length ℓ in the protograph. Let (e_1, e_2,..., e_ℓ) represent the

sequence of edges of cycle C in $\mathcal{H}$ and let $(a_1, a_2, \ldots, a_\ell)$ be the corresponding permutation shifts. Then, it is possible to associate the cycle C with a value $\phi(C)$ defined as

$$\phi(C) = \sum_{i=1}^{\ell} (-1)^{i+1} \; a_i \mod N, \tag{6}$$

where by convention, the first term of the cycle is on the upper left position of the cycle in the parity-check matrix (see Fig. 1 on page 5). Since the graph is bipartite, the length ℓ of a cycle is always even. Thus, the direction of the cycle does not impact the value of $\phi(C)$ since $(-1)^{\ell-i} = (-1)^i$.

Lemma 2.1 (Cycle breaking): Cycle C lifts into cycles of length $\ell' > \ell$ if and only if $\phi(C) \neq 0$.

Proof: See [20] □

The problem can now be modeled as finding the shift values of the circulant matrices to maximize the girth of the expanded matrix, or alternatively, finding the minimum expansion factor N required to obtain a given girth.

C. Equivalence relation

A permutation π of size n is a bijection function of set $\mathcal{N} = \{0, 1, \ldots, n–1\}$ to itself: $\pi(\mathcal{N}) = \{\pi(0), \pi(1), \ldots, \pi(n–1)\}$. The identity permutation is denoted Id. It is possible to represent the permutation π by a (n, n) matrix, called the permutation matrix P_π, defined as $P_\pi(i, j) = 1$ if $j = \pi(i)$, 0 otherwise. The inverse permutation of π will be denoted π^{-1} and verifies $\pi^{-1}(\pi) = \pi(\pi^{-1}) = Id$. Moreover, $(P_\pi)^{-1} = P_{\pi^{-1}}$ and $P_{\pi^{-1}} \times P_\pi = P_\pi \times P_{\pi^{-1}} = I_0^n$ with I_0^n the identity matrix of size $(n \times n)$. Finally, $(P_\pi)^{-1} = P_\pi^t$, where P^t means the transposed matrix of P.

Definition 2.2 (Equivalence of matrices): Two matrices A and B of size (n, m) are said to be "permutation equivalent" if and only if there exist two permutation matrices P_r (index r for row) of size $(n \times n)$ and P_c (index c for column) of size $(m \times m)$ such that $A = P_r \times B \times P_c$. The relation is denoted as $A \equiv B$.

Theorem 2.3: The permutation equivalent relation defines an equivalence relation.

Proof: The relation is reflexive ($A \equiv A$), symmetrical ($A = P_r \times B \times P_c$ implies that $B = P_r^{-1} \times A \times P_c^{-1}$) and associative ($A = P_r^a \times B \times P_c^a$ and $B = P_r^b \times C \times P_c^b$ implies $A = P_r^a P_r^b \times C \times P_c^b P_c^a$). □

Let A and B be two lifted matrices from the same prototype matrix $\mathbf{H}$, then, if $A \equiv B$, then A and B are equivalent expanded matrices of $\mathbf{H}$. This relation is denoted $A \equiv_{\mathbf{H}} B$, the subscript $\mathbf{H}$ being omitted when there is no ambiguity.

To summarize, if $A \equiv_{\mathbf{H}} B$, then A and B correspond to the same Tanner graph up to a renumbering of variable and check nodes, and thus, represent the same code. The aim of this paper is twofold: to define classes of equivalent solutions using permutations and present static symmetry breaking [27] to identify a solution per class of equivalence.

III. Additive and structural transformations

In the following, "equivalence class" should be understood in terms of the equivalence relation defined in Section II-C. The main goals of this approach are (i) to determine the number of equivalence classes (possibly, how many elements there are in each equivalence class), and (ii) to obtain a unique representative for each equivalence class. To address the above issues, two transformations that preserve the equivalence relation are presented: the "additive transformations", already proposed in [20] and the "structural transformation". In the sequel, the size N of the matrix is omitted when there is no ambiguity.

A. Additive transformation

In this section, we introduce the additive transformation, a state-of-the-art permutation [20] which reduces the degree of freedom of the problem example from 12 (the number of variables in (5)) down to 4.

Definition 3.1 (Circulant permutation): Let π_a^+ be the permutation over $\{0, 1, \ldots n-1\}$ defined as $\pi_a^+(i) = i + a \mod n$. The permutation matrix P_a^+ associated with π_a^+ is also equal to the circulant matrix $P_a^+ = I_a^n$.

Property 3.2: The product of two circulant matrices P_a^+ and P_b^+ is equal to the circulant matrix $P_a^+ \times P_b^+ = P_{a+b}^+$ [25].

Property 3.3 (Row rearrangement): Let M be an $n \times m$ matrix and π_r a permutation of size n. Then, $P_{\pi_r} \times M$ is the (n, m) matrix obtained by permuting the rows of M according to π_r.

Property 3.4 (Column rearrangement): Let M be a $n \times m$ matrix and π_c a permutation of size m. Then $M \times P_{\pi_c}$ is the (n, m) matrix obtained by permuting the columns of M according to π_c^{-1}.

Proof: Since for any matrix A, $A = (A^t)^t$, then $M \times P_{\pi_c} = (P_{\pi_c}^t \times M^t)^t$. The product $P_{\pi_c}^t \times M^t$ permutes the row of M^t according to $P_{\pi_c}^t$, i.e., according to permutation π_c^{-1}. Thus, its transpose gives the permutation of the column of M according to π_c^{-1}. □

Let us define C_r^+ as the (JN, JN) block diagonal permutation matrix defined as

$C_r^+ = \text{diag}(P_{r(1)}^+, P_{r(2)}^+, \ldots P_{r(J)}^+)$, with $r = (r(1), r(2), \ldots r(J))$ representing a vector of integers.

Property 3.5 (Row shift): $H^{r^+} = C_r^+ \times H$ is the expanded matrix defined by $\mathcal{H}^{r^+}$ with $\mathcal{H}^{r^+} = \mathcal{H}(i, j) + r(i) \;\; \forall i \in \{1, 2, \ldots J\}$, $j \in \{1, 2, \ldots L\}$.

Let us define C_c^+ as the (LN, LN) block diagonal permutation matrix defined as

$C_c^+ = \text{diag}(P_{c(1)}^+, P_{c(2)}^+, \ldots P_{c(L)}^+)$, with $c = (c(1), c(2), \ldots c(L))$ representing a vector of integers.

Property 3.6 (Column shift): $H^{c^+} = H \times C_c^+$ is the expanded matrix defined by $\mathcal{H}^{c^+}$ with $\mathcal{H}^{c^+}(i, j) = \mathcal{H}(i, j) + c(j) \;\; \forall i \in \{1, 2, \ldots, J\}$, $j \in \{1, 2, \ldots, L\}$.

Let $\mathbb{H}_3^N$ denote the set of matrices obtained by an expansion of a factor N of the prototype matrix $\mathbf{H}_3$. Each matrix $H \in \mathbb{H}_3^N$ can be represented as (5). From $r_1 = (-a_1, -a_2, 0)$ we can construct the $(3N, 3N)$ permutation matrix $C_{r_1}^+ = \text{diag}(P_{-a_1}^+, P_{-a_2}^+, P_0^+)$ and $H^{r_1^+} = C_{r_1}^+ \times H$. According to property 3.5, $H^{r_1^+}$ can be represented as

$$\mathcal{H}^{r_1^+} = \begin{bmatrix} 0 & b_1 - a_1 & c_1 - a_1 & d_1 - a_1 & . & . \\ 0 & b_2 - a_2 & . & . & e_2 - a_2 & f_2 - a_2 \\ . & . & c_3 & d_3 & e_3 & f_3 \end{bmatrix} \tag{7}$$

Then, multiplying $H^{r_1^+}$ by the $(6N, 6N)$ permutation matrix C_c^+ with $c = (0, a_1-b_1, a_1-c_1, a_1-d_1, a_2-e_2, a_2-f_2)$, gives $H^{r_1^+,c^+} = H^{r_1^+} \times C_r^+$. According to property 3.6, $H^{r_1^+,c^+}$ can be represented as $\mathcal{H}^{r_1^+,c^+} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & . & . \\ 0 & b_2{-}a_2{+}a_1{-}b_1 & . & . & 0 & 0 \\ . & . & c_3{+}a_1{-}c_1 & d_3{+}a_1{-}d_1 & e_3{+}a_2{-}e_2 & f_3{+}a_2{-}f_2 \end{bmatrix} \tag{8}$$

Finally, multiplying $H^{r_1^+,c^+}$ on the left by $C_{r_2}^+$ with $r_2 = (0, 0, -c_3 - a_1 + c_1)$ gives $H^{r_1^+,c^+,r_2^+} = C_{r_2}^+ \times H^{r_1^+,c^+}$. According to property 3.6, $H^{r_1^+,c^+,r_2^+}$ can be represented as $\mathcal{H}^{r_1^+,c^+,r_2^+}$, which is defined as

$$\mathcal{H}^{r_1^+,c^+,r_2^+} = \begin{bmatrix} 0 & 0 & 0 & 0 & . & . \\ 0 & b & . & . & 0 & 0 \\ . & . & 0 & d & e & f \end{bmatrix}, \tag{9}$$

where

$$\begin{cases} b = b_2 - a_2 + a_1 - b_1, \\ d = d_3 - d_1 - c_3 + c_1, \\ e = e_3 + a_2 - e_2 - c_3 - a_1 + c_1, \\ f = f_3 + a_2 - f_2 - c_3 - a_1 + c_1. \end{cases} \tag{10}$$

Property 3.7 (bdef-pattern): Any matrix $H \in \mathbb{H}_3^N$ is equivalent to a matrix of type (9).

Thus the study of the set $\mathbb{H}_3^N$ can be restricted to the study of matrices defined in (9). To simplify notation, a matrix $H \in \mathbb{H}_3^N$ can be also represented by the 4-tuple $H = \langle b, d, e, f \rangle_N$. In this section, as mentioned before, we apply the results of [20] to the case of the prototype matrix $\mathbf{H}_3$. This transformation is very general and can be applied to any QC-LDPC matrix. In fact, the additive transformation can be used to set the first non-negative coefficients of each column of $\mathcal{H}$ to zero and the first non-negative coefficient of each row of $\mathcal{H}$ to zero, thus reducing the dimension of the search space from Jd_c (or Ld_v, i.e., the number of non-negative values of $\mathcal{H}$) down to $Jd_c - J - L + 1$.

It is worth noticing that matrices of the form $H = \langle b, d, e, f \rangle$ do not give a unique representative of each equivalence class, i.e., two matrices $H = \langle b, d, e, f \rangle$ and $H' = \langle b', d', e', f' \rangle$ may be equivalent, even if $(b, d, e, f) \neq (b', d', e', f')$. The next section presents the first contribution of this paper: the structural transformation that uses a different approach to identify equivalence between matrices.

B. Structural transformation

Let us consider the set $\mathbb{P}_{J,L}$ of pair of permutations (π_r, π_c) of size equal to L and J, respectively, that satisfies

$$\mathbf{H} = \mathbf{P}_{\pi_r} \times \mathbf{H} \times \mathbf{P}_{\pi_c}, \tag{11}$$

where $\mathbf{P}_{\pi_r}$ and $\mathbf{P}_{\pi_c}$ are, respectively, the permutation matrices associated with π_r and π_c.

For example, $(\pi_r, \pi_c) = (\{2, 3, 1\}, \{4, 3, 6, 5, 2, 1\})$ belongs to $\mathbb{P}_{3,6}$ since

$$\mathbf{P}_{\pi_r} \times \mathbf{H}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & . & . \\ 1 & 1 & . & . & 1 & 1 \\ . & . & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & . & . & 1 & 1 \\ . & . & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & . & . \end{bmatrix},$$

then $(\mathbf{P}_{\pi_r} \times \mathbf{H}_3) \times \mathbf{P}_{\pi_c}$ gives

$$\begin{bmatrix} 1 & 1 & . & . & 1 & 1 \\ . & . & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & . & . \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & . & . \\ 1 & 1 & . & . & 1 & 1 \\ . & . & 1 & 1 & 1 & 1 \end{bmatrix} = \mathbf{H}_3. \tag{13}$$

Property 3.4 implies that the right product with P_{π_c} is equivalent to the permutation of the columns according to π_c^{-1}. In the example given here, since in (13), $\pi_c = \{4, 3, 6, 5, 2, 1\}$, we have $\pi_c^{-1} = \{6, 5, 2, 1, 4, 3\}$. It is possible to interpret directly the effect of left multiplication of a matrix A by P_{π_c}: since $\pi_c(6) = 1$, the last column of A would be the first column of $A \times P_{\pi_c}$, and since $\pi_c(5) = 2$, the fifth column of A would be second column of $A \times P_{\pi_c}$ and so on.

Let $A \otimes B$ represent the Kronecker product of A and B. Let $P_{\pi_r} = \mathbf{P}_{\pi_r} \otimes I_0^N$, $P_{\pi_c} = \mathbf{P}_{\pi_c} \otimes I_0^N$ and $H \in \mathbb{H}_3^N$, then $H^{\pi_r,\pi_c} = P_{\pi_r} \times H \times P_{\pi_c}$ is also an element of $\mathbb{H}_3^N$ satisfying $H^{\pi_r,\pi_c} \equiv H$. Thus, structural transformations preserve the equivalence relation. In other words, several equivalence classes can be merged owing to the structural transformation, hence reducing the space search.

Going back to the example, let $H = \langle b, d, e, f \rangle$ be an element of $\mathbb{H}_3$ and (π_r, π_c) the pair of permutations defined above, then H^{π_r,π_c} can be represented as

$$\mathcal{H}^{\pi_r,\pi_c} = \begin{bmatrix} 0 & 0 & b & 0 & . & . \\ f & e & . & . & d & 0 \\ . & . & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{14}$$

Then, using (10), H^{π_r,π_c} (and thus H) is equivalent to $\langle e - f, b, b - d + f, b + f \rangle$.

For each of the 6 permutations π_r on the 3 lines of the prototype matrix $\mathbf{H}_3$, there are exactly 8 permutations π_c on columns that give structural transformations. For example, if π_r is equal to the identity, then permuting columns 1 and 2, columns 3 and 4 or columns 5 and 6 does not change the structure of $\mathbf{H}_3$ (2). By combining these 3 permutations, a total of 8 structural permutations are obtained when π_r is equal to the identity. The first column of Tables VI and VII gives an exhaustive enumeration of the $6 \times 8 = 48$ structural transformations. If b, d, $f - e$ and $N/2$ are all distinct and non null, then each of the 48 permutations gives a distinct matrix after the transformation (see Tables VI and VII). If this condition is not fulfilled, then the number of distinct solutions is lower. For example, if $f - e = 0$, i.e., $e = f$,

then permutations 1 and 2 of Table VI give the same matrix $\langle +b, +d, +e = +f, +f = +e\rangle$. More generally, if $e = f$, then for any $i \in \{1, 2, \dots 24\}$, permutations presented in lines $2i-1$ and $2i$ also give the same result: the number of distinct solutions is thus halved.

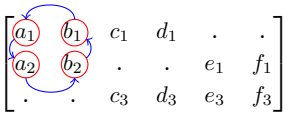
Tables VI and VII show the 48 equivalent 4-tuples of $\langle 1, 13, 15, 23\rangle_{35}$ (subscript 35 indicates a factor of expansion $N = 35$) as well as the formal expression of the transformed 4-tuple.

IV. Structural and additive dominance breaking

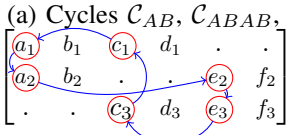
In the previous section, we show that, when applied to any H, the 48 permutation pairs of $\mathbb{P}_{3,6}$ give a matrix H^{π_r,π_c} in the same equivalence class. In this section, we present dominance constraints and use them to find one solution per class.

The constraints are based on the cycles in the protograph matrix. As shown in section II-B, a cycle can be defined by a sequence of consecutive moves (in a column, then in a row, then again in a column and so until going back to the starting point) between the non-null positions of the prototype matrix. In the example, the degree of the variable node is $d_v = 2$, thus, there is no need to specify the row index to define cycles in $\mathbf{H}_3$. A column in $\mathbf{H}_3$ is denoted by a letter, "A" being the first column and "F" the last column. By convention we denote a cycle by a sequence of columns considering that the first move is going down in the first column. The moves on subsequent columns (i.e., either down or up) are uniquely determined and do not need to be specified since the column weight is equal to 2.

Example $\mathcal{C}_{AB}$ denotes the cycle going down on column A, then up in column B. Cycle $\mathcal{C}_{AB}$ is thus going through nodes $(a_1 \to a_2 \to b_2 \to b_1)$ in Figure 1(a). Cycle $\mathcal{C}_{AEC}$ denotes the cycle going down on column A then down on column E and finally up in column C (see Figure 1(b)). It is important to note that, for example, $\mathcal{C}_{ABAB}$ (respectively $\mathcal{C}_{ABABAB}$) denotes the cycle of length 8 (respectively 12) that makes 2 (respectively 3) round trips between columns A and B. According to lemma 2.1, the enumeration of all cycles of length ℓ that uses one or more columns more than once is also required.



(a) Cycles $\mathcal{C}_{AB}$, $\mathcal{C}_{ABAB}$, ...



(b) Cycles $\mathcal{C}_{AEC}$, $\mathcal{C}_{AECAEC}$, ...

Fig. 1. Graphical description of cycles

As previously mentioned, any matrix in $\mathbb{H}_3$ is equivalent to a matrix $H = \langle b, d, e, f\rangle$. It can thus be characterized by the value of the 3 cycles of length four in matrix $\mathbf{H}_3$, i.e. $\mathcal{C}_{AB}, \mathcal{C}_{CD}$ and $\mathcal{C}_{EF}$, i.e., by the tuple $\Phi(H) = \langle \phi(\mathcal{C}_{AB}), \phi(\mathcal{C}_{CD}), \phi(\mathcal{C}_{EF})\rangle_N = \langle b,\ d,\ f-e\rangle_N$, where the $\phi(\mathcal{C})$ is the value of the cycle, as defined in (6).

A. Φ-order constraint: $\phi(\mathcal{C}_{AB}) \le \phi(\mathcal{C}_{CD}) \le \phi(\mathcal{C}_{EF})$.

Lemma 4.1 (Swapping $\mathcal{C}_{AB}$ and $\mathcal{C}_{CD}$): Any matrix $H = \langle b, d, e, f\rangle$ of $\mathbb{H}_3$ characterized by $\Phi(H) = \langle b, d, f-e\rangle$ is equivalent to a matrix $H_1 \in \mathbb{H}_3$ characterized by $\Phi(H_1) = \langle d, b, f-e\rangle$.

Proof: According to line 26 of Table VII, $H = \langle b, d, e, f\rangle$ is equivalent to the matrix $H_1 = \langle d, b, -f, -e\rangle$ owing to the equivalence relation $H_1 = P_{\pi_r} \times H \times P_{\pi_c}$, with $(\pi_r, \pi_c^{-1}) \in \mathbb{P}_{3,6}$ defined as $\pi_r = \{1, 3, 2\}$ and $\pi_c = \{3, 4, 1, 2, 5, 6\}$. Matrix H_1 is characterized by $\Phi(H_1) = \langle d, b, f-e\rangle$, which completes the proof. □

Lemma 4.2 (Swapping $\mathcal{C}_{CD}$ and $\mathcal{C}_{EF}$): Any matrix $H = \langle b, d, e, f\rangle$ of $\mathbb{H}_3$ characterized by $\Phi(H) = \langle b, d, f-e\rangle$ is equivalent to a matrix $H_2 \in \mathbb{H}_3$ characterized by $\Phi(H_2) = \langle b, f-e, d\rangle$.

Proof: Same proof as Lemma 4.1 using the equivalence given in line 13 of Table VI. □

Theorem 4.3 (Φ-Order): Any matrix H of $\mathbb{H}_3$ is equivalent to a matrix $H_0 = \langle b, d, e, f\rangle$, with $\Phi(H_0) = \langle b, d, f-e\rangle$ satisfying $b \le d \le f-e$.

Proof: Lemmas 4.1 and 4.2 show that it is possible to generate equivalent solutions by swapping, respectively, the first and second terms of $\Phi(H)$, then the second and third terms of $\Phi(H)$. By combining these two swapping permutations, any order of $\Phi(H)$ can be obtained, in particular, the one where the values are in increasing order. □

The Φ-order constraint defines a dominance relation over the solutions. Thus, one can restrict the search to solutions respecting this constraint, which reduces the size of the search space by almost a factor of 6.

B. $N/2$-constraints: $\phi(\mathcal{C}_{AB}) \le N/2$, $\phi(\mathcal{C}_{CD}) \le N/2$ and $\phi(\mathcal{C}_{EF}) \le N/2$.

Lemma 4.4: Any matrix $H = \langle b, d, e, f\rangle$ of $\mathbb{H}_3$ characterized by $\Phi(H) = \langle b, d, f-e\rangle$ is equivalent to a matrix $H_3 \in \mathbb{H}_3$ characterized by $\Phi(H_3) = \langle -b, d, f-e\rangle$.

Proof: Similarly to the proof of Lemma 4.1, this involves the transformation of line 5 of Table VI. □

Lemma 4.5: Any matrix $H = \langle b, d, e, f\rangle$ of $\mathbb{H}_3$ characterized by $\Phi(H) = \langle b, d, f-e\rangle$ is equivalent to a matrix $H_4 \in \mathbb{H}_3$ characterized by $\Phi(H_4) = \langle b, -d, f-e\rangle$.

Proof: Similarly to the proof of Lemma 4.1, this involves the transformation of line 3 of Table VI. □

Lemma 4.6: Any matrix $H = \langle b, d, e, f\rangle$ of $\mathbb{H}_3$ characterized by $\Phi(H) = \langle b, d, f-e\rangle$ is equivalent to a matrix $H_5 \in \mathbb{H}_3$ characterized by $\Phi(H_5) = \langle b, d, -(f-e)\rangle$.

Proof: Similarly to the proof of Lemma 4.1, this involves the transformation of line 2 of Table VI. □

Theorem 4.7 (constraint $N/2$): Any matrix H of $\mathbb{H}_3$ is equivalent to a matrix $H_0 = \langle b, d, e, f\rangle$, with $\Phi(H) = \langle b, d, f-e\rangle$ satisfying $b \le N/2$, $d \le N/2$ and $f - e \le N/2$.

Proof: Let us consider $H = \langle b, d, e, f\rangle$ where $b > N/2$, then using Lemma 4.4, H is equivalent to $H_0 = \langle -b, d, e, f\rangle$. Since the operation is performed modulo N, $b > N/2 \Rightarrow (-b \mod N) \le N/2$. The proof is completed by Using lemma 4.5 and 4.6 in a similar way. □

C. Φ-order-$N/2$ constraint: $\phi(\mathcal{C}_{AB}) \leq \phi(\mathcal{C}_{CD}) \leq \phi(\mathcal{C}_{EF}) \leq N/2$.

Theorem 4.8 (Constraint Φ-order-$N/2$): Any matrix H of $\mathbb{H}_3$ is equivalent to a matrix $H_0 = \langle b, d, e, f\rangle$, with $\Phi(H_0) = \langle b, d, f-e\rangle$ satisfying $b \leq d \leq f - e \leq N/2$.

Proof: Using Theorem 4.7, H is equivalent to a matrix H_0 satisfying the constraint $N/2$. Then, the proof is completed by applying Theorem 4.3 on H_0 to sort the ϕ values of H_0 in increasing order. □

Constraint $\phi(\mathcal{C}_{AB}) \leq \phi(\mathcal{C}_{CD}) \leq \phi(\mathcal{C}_{EF}) \leq N/2$ defines a dominance relation, and the search of solutions can then be restricted to those respecting the constraint. This relation allows us to reduce by up to 48 the size of the space to explore. For example, in tables VI and VII, the only equivalent solution H_0 of $H = \langle 1, 7, 13, 17\rangle_{35}$ satisfying the Φ-order-$N/2$ constraint is given on line 13 (in a grey cell in the fifth column) by $H_0 = \langle 1, 4, 21, 28\rangle_{35}$. The Φ value of H_0 is given by $\Phi(H_0) = \langle 1, 4, 7\rangle_{35}$ that satisfies $1 \leq 4 \leq 7 \leq 35/2$.

D. Φ^0-order-$N/2$ constraint: $0 < \phi(\mathcal{C}_{AB}) \leq \phi(\mathcal{C}_{CD}) \leq \phi(\mathcal{C}_{EF}) \leq N/2$.

Finally, we can also add an additional constraint to the Φ-order-$N/2$ constraint by imposing $0 < b$. In fact, $b = 0$ leads to weak expanded matrices with a girth of size 4 as explained in section VI-A. In the following, this new constraint is called the Φ^0-order-$N/2$ constraint. We also define $\mathbb{H}_3^{N,0}$ (or $\mathbb{H}_3^0$, to simplify the notation) as the subset of all matrices of $\mathbb{H}_3^N$ equivalent to a matrix $H = \langle b, d, e, f\rangle$ respecting the Φ^0-order-$N/2$ constraint, i.e. $0 < b \leq d \leq f - e \leq N/2$. Hence, $\mathbb{H}_3^{N,0}$ is the subset of $\mathbb{H}_3^N$ with expanded matrices of girth $g \geq 6$.

V. Multiplicative transformation

In this section, we present a new transformation which relies on the multiplication of the values of the circulant matrices of $\mathcal{H}$. The properties of this transformation are presented in Section V-A and used in Section V-B to identify new forms of equivalent matrices, thus reducing further the search space.

A. Properties

Definition 5.1 (Product-matrix): Let $k \in \mathbb{Z}/N\mathbb{Z}$ be an integer coprime with N (the greatest common divisor of k and N is equal to 1, i.e., $\gcd(k, N) = 1$), then $P_k^\times$ is the permutation matrix associated with the permutation $\pi_k^\times(i) = k \times i \mod N$.
Since k and N are coprime, then there exists an integer k^{-1} so that $k \times k^{-1} = 1 \mod N$ and thus, $(\pi_k^\times)^{-1} = \pi_{k^{-1}}^\times$ and $(P_k^\times)^{-1} = P_{k^{-1}}^\times$.

Theorem 5.2 (Multiplication over a circulant matrix): Let $k \in \mathbb{Z}/N\mathbb{Z}$ coprime with N and I_a^N the a-circulant matrix, then $P_k^\times \times I_a^N \times P_{k^{-1}}^\times$ is equal to the $(ka \mod N)$-circulant matrix I_{ak}^N.

Proof: $P_k^\times \times I_a^N \times P_{k^{-1}}^\times$ is the permutation matrix associated with the permutation $\pi = \pi_k^\times(\pi_a^+(\pi_{k^{-1}}^\times))$. For all $i \in \mathbb{Z}/N\mathbb{Z}$, we have $\pi_{k^{-1}}^\times(i) = k^{-1}i$, thus $\pi_a^+(\pi_{k^{-1}}^\times(i)) = k^{-1}i + a$ and therefor, $\pi(i) = \pi_k^\times(k^{-1}i + a) = k(k^{-1}i + a) = i + ka$. Thus, π is the ka-circular permutation, and therefore, its associated permutation matrix is I_{ka}. □

Example 5.3: Consider $N = 32$, $k = 7$ and $a = 3$. Since $\gcd(7, 32) = 1$, then 7^{-1} exists over $\mathbb{Z}/32\mathbb{Z}$. Since $7 \times 23 = 161 = 1 + 5 \times 32 = 1 \mod 32$, then $7^{-1} = 23$ over $\mathbb{Z}/32\mathbb{Z}$. Thus, according to Theorem 5.2, $P_7^\times \times I_3^{32} \times P_{23}^\times = I_{21}^{32}$. Another way to reach this result has been presented in reference [19]. It can be extended to the whole prototype matrix to identify a new type of transformation (i.e., multiplicative transformation) that also preserves the equivalence relation.

Theorem 5.4 (Multiplicative rearrangement): $H_k^\times = (I_0^J \otimes P_k^\times) \times H \times (I_0^L \otimes P_{k^{-1}}^\times)$ is the expanded matrix defined as $\mathcal{H}_k^\times(i, j) = k \times \mathcal{H}(i, j)$.

Proof: The proof of this theorem comes directly from Theorem 5.2 since each (N, N) expanded matrices of $H_k^\times$ (see (4)) is equal either to $P_k^\times \times 0^N \times P_{k^{-1}}^\times = 0^N$ or $P_k^\times \times I_a^N \times P_{k^{-1}}^\times = I_{ka}^N$. □

Combined with additive and structural transformations, multiplicative transformation allows us to identify more matrices in the same equivalent class, thus reducing the search space since only one representative of each equivalence class needs to be tested.

B. Symmetry breaking

In this section, we use Theorem 5.4 to identify new subsets of equivalent matrices in $\mathbb{H}_3$. Note that this method can be applied to any prototype matrix.

Property 5.5 (Coprime multiplication): Let k be a number coprime with N, then any matrix $H = \langle b, d, e, f\rangle_N$ of $\mathbb{H}_3$ is equivalent to the matrix $\langle kb, kd, ke, kf\rangle_N$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & . & . \\ 0 & b & . & . & 0 & 0 \\ . & . & 0 & d & e & f \end{bmatrix} \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & . & . \\ 0 & kb & . & . & 0 & 0 \\ . & . & 0 & kd & ke & kf \end{bmatrix} \quad (15)$$

Proof: Derived directly from Theorem 5.4. □

Once a solution is found, it is possible to build the class of all equivalent solutions for all k coprime with N.

Property 5.6 (Divisors of N): If $b \neq 0$, b can be restricted to values in the set $\mathcal{D}(N)/N$ of the divisors of N with N excluded.

Proof: From Bezout's theorem, there exist two integers u, v so that $ub + vN = \gcd(b, N)$, where gcd is the "greatest common divisor". For any integer ρ, let us define k_ρ as $k_\rho = u + \rho \times N/\gcd(b, N)$. Thus, by construction, $k_\rho \times b = \gcd(b, N) \mod N$. According to Diriclet's prime number theorem, since u and $N/\gcd(b, N)$ are coprime, the set $\{k_\rho\}_{\rho\in\mathbb{N}}$ contains an infinity of prime numbers, thus, there exists a k_ρ coprime with N which complete the proof. □

For example, let $N = 30$ and $H = \langle 12, 13, 4, 17\rangle_{30}$. Since $b = 12$, we have $\gcd(b, N) = 6$ and $N/\gcd(b, N) = 5$. Taking $u = 3$ and $v = -1$ gives $3 \times 12 + (-1) \times 30 = 6$. Thus, $k_\rho = 3 + 5\rho$ takes respectively the values $\{3, 8, 13, 18, 23, 28\}$ for $\rho = 0, 1, \ldots, 5$. In this series, $k_2 = 13$ and $k_4 = 23$ are the only integers coprime with $N = 30$. Thus, using k_2, $H = \langle 12, 13, 4, 17\rangle_{30}$ is equivalent to $\langle 6, 19, 22, 11\rangle_{30}$, and using k_4, H is also equivalent to $\langle 6, 29, 2, 1\rangle_{30}$.

Theorem 5.7 (Global Constraint): Any matrix H of $\mathbb{H}_3^0$ is equivalent to a matrix $H_n = \langle b_n, d_n, e_n, f_n\rangle$, with $\Phi(H_n) =$

$\langle b_n, d_n, f_n - e_n\rangle$ verifying $0 < b_n \leq d_n \leq f_n - e_n \leq N/2$ and b_n a divisor of N verifying $b_n \leq \gcd(d_b, N)$ and $b_n \leq \gcd(f_n - e_n, N)$.

Proof:

The proof of this Theorem is performed by recursion, by showing the existence of a series of equivalent matrices $H_0 = \langle b_0, d_0, e_0, f_0\rangle$, $H_1 = \langle b_1, d_1, e_1, f_1\rangle$, ..., $H_n = \langle b_n, d_n, e_n, f_n\rangle$ satisfying the Φ^0-order-$N/2$ constraint, with decreasing value of b, i.e. $(b_0 > b_1 > b_2 \cdots > b_n \geq 1)$ until b_n reaches the value of a divisor of N (eventually, $b_n = 1$).

Initial condition: From Theorem 4.8, any matrix H of $\mathbb{H}_3^0$ is equivalent to a matrix $H_0 = \langle b_0, d_0, e_0, f_0\rangle$ satisfying the Φ^0-order-$N/2$ constraint.

Recursion: Let us assume that H is equivalent to a matrix $H_n = \langle b_n, d_n, e_n, f_n\rangle$ satisfying the Φ^0-order-$N/2$ constraint with $b_n \geq 1$. If b_n is a divisor of N, the condition of the theorem is fulfilled. Otherwise, Theorem 5.4 shows that there exists k such that H_n is equivalent to $\langle \gcd(b_n, N), kd_n, ke_n, kf_n\rangle$. Note that $\gcd(b_n, N) < b_n$ since b_n is not a divisor of N. Due to the multiplication by k and the reduction modulo N, $\langle \gcd(b_n, N), kd_n, ke_n, kf_n\rangle$ may no longer respect the Φ^0-order-$N/2$ constraint. Nevertheless, it is still possible to find an equivalent matrix $H_{n+1} = \langle b_{n+1}, d_{n+1}, e_{n+1}, f_{n+1}\rangle$ that respects the constraint. Since $\gcd(b_n, N) \leq N/2$, it will be only affected by swapping operations during the construction of an equivalent matrix respecting the Φ^0-order-$N/2$ constraint (see proof of Theorem 4.8), and thus, $1 \leq b_{n+1} \leq \gcd(b_n, N) < b_n$.

Finally, if $b_n > \gcd(d_n, N)$, then d_n and b_n can be swapped and the same process will lead to $b_{n+1} = \gcd(d_n, N) < b_n$. The same method can be applied if $b_n > \gcd(f_n - e_n, N)$ □

To conclude this section, we use multiplicative permutation to show that the number of equivalent classes (or equivalently, the space of search, since only one element per class needs to be tested) is upper bounded by $\rho(N)$, where $\rho(N)$ is the number of distinct tuples $\langle b, d, e, f\rangle$ respecting the Φ^0-ordre-$N/2$ constraint and $b \in \mathcal{N}$. The value $\rho(N)$ can be computed as

$$\rho(N) = \sum_{b\in\mathcal{D}(N)/N} \sum_{d=b}^{N/2} \sum_{d\leq f-e\leq N/2} 1. \tag{16}$$

Let us focus on the last of the three sum operators of (16). For each value of e, there are $\lfloor N/2\rfloor - d + 1$ possible values of f that fulfilled the constraint $d \leq (f - e) \leq N/2$, so the last summation term is equal to $N(\lfloor N/2\rfloor - d + 1)$. According to the second summation term of (16), d varies from b to $\lfloor N/2\rfloor$, thus $(\lfloor N/2\rfloor - d + 1)$ takes all the values from 1 up to $\lfloor N/2\rfloor - b + 1$, thus

$$\rho(N) = N \sum_{b\in\mathcal{D}(N)/N} \frac{(\lfloor N/2\rfloor - b + 1)(\lfloor N/2\rfloor - b + 2)}{2}. \tag{17}$$

Note that if N is a prime number greater than 2, then $\mathcal{D}(N)/N = \{1\}$ and thus, $\rho(N) = (N^3 - N)/8$. For example, for $N = 5$, the $\rho(5) = 15$ solutions are $\langle 1, 1, 0, 1\rangle$, $\langle 1, 1, 0, 2\rangle$, $\langle 1, 1, 1, 2\rangle$, $\langle 1, 1, 1, 3\rangle$, $\langle 1, 1, 2, 3\rangle$, $\langle 1, 1, 2, 4\rangle$, $\langle 1, 1, 3, 4\rangle$, $\langle 1, 1, 3, 0\rangle$, $\langle 1, 1, 4, 0\rangle$, $\langle 1, 1, 4, 1\rangle$, $\langle 1, 2, 0, 2\rangle$, $\langle 1, 2, 1, 3\rangle$, $\langle 1, 2, 2, 4\rangle$, $\langle 1, 2, 3, 0\rangle$ and $\langle 1, 2, 4, 0\rangle$.

C. Discussion

The links between structural additive equivalence and multiplicative transformation are still an open problem. In Tables VI and VII, the last column shows three equivalent matrices generated from the multiplicative transformation of $\langle 1, 7, 13, 17\rangle_{35}$ satisfying the Φ^0-order-$N/2$ constraint, meaning that the three matrices (in grey in the last column) belong to the same equivalence class. So far, we have been unable to find an a priori method to determine a single element of this class. Nevertheless, once the matrices of interest are generated for a given girth, it is possible a posteriori to prune the space of solutions by seeking equivalent solutions using multiplicative transformation and keeping only one element per equivalent class. The present authors consider that, while the problem may be of interest to group theorist specialists, an exhaustive analysis of the group properties is beyond the scope of their paper. Finally, it is worth mentioning that the method used here can be generalized for any prototype matrix to reduce the search space. The difficulty is first to find the structural set $\mathbb{P}_{\mathbf{H}}$ of pair of permutations associated with the prototype matrix $\mathbf{H}$, and then find a simple constraint to determine a single matrix per equivalence class. This is a challenging problem to solve in the general case!

Finally, readers interested in the mathematical aspects of the present study are invited to consult the on-line note written of Xavier Giraud based on an early version of our paper [28]. M. Giraud also contributed to the paper by pointing out, in the global theorem, that b can be also be taken smaller or equal than $\gcd(d, N)$ and than $\gcd(f - e, N)$.

VI. Construction of high girth matrices

In this section, we construct H matrices of minimum size from the $\mathbf{H}_3$ prototype matrix with a girth ranging from $g = 8$ up to $g = 14$. Then, we give an explicit rule based on the multiplicity $\mathcal{M}(g)$ to select the matrix. Finally, to illustrate the efficiency of the proposed method, we present a very early promising result for another type of prototype matrix.

A. Optimal lifting of the $\mathbf{H}_3$ protograph.

The problem of optimal lifting by a factor N of the $\mathbf{H}_3$ protograph matrix can be formalized in two steps. The first step is to define the maximum achievable girth $\bar{g}_N$ as

$$\bar{g}_N(H) = \max_{H\in\mathbb{H}_3^N} g(H), \tag{18}$$

where $g(H)$ is the girth of matrix H. From $\bar{g}_N(H)$, we can define $\mathbb{H}_3^{N,o}$ as the subset of $\mathbb{H}_3^N$ of matrices of maximum girth $\bar{g}_N$. Then the optimal lifted matrices H^o is given by

$$H^o = \arg\min_{H\in\mathbb{H}_3^{N,o}} \mathcal{M}(\bar{g}_N(H)), \tag{19}$$

where $\mathcal{M}(\bar{g}_N(H))$ is the number of cycles of length $\bar{g}_N(H)$ of the matrix H. Note that if several distinct matrices lead

to the same $\mathcal{M}(\bar{g}_N(H))$ minimum value, then the value of $\mathcal{M}(\bar{g}_N(H)+2)$ can be used as a second criteria and so on.

In the following, we present some results on the inverse problem, i.e., to find the minimum value of N required to guaranty a given girth g_0. To do so, we first express all the cycles of length $\ell = 4$, 6, 8, 10, 12 and 14 of $\mathbf{H}_3$ using the method presented in [29]. Note that the software to perform this enumeration is available online [30]. The number $\mathcal{N}(\ell)$ of cycles for each value of ℓ is given in Table I. Each cycle gives a new constraint on the lifted matrix according to equation (6).

	$\ell=4$	$\ell=6$	$\ell=8$	$\ell=10$	$\ell=12$	$\ell=14$
$\mathcal{N}(\ell)$	3	8	11	40	139	336
Total	3	11	22	62	201	537

TABLE I
NUMBER OF CONSTRAINTS TO BE TAKEN INTO ACCOUNT BY THE CONSTRAINT PROGRAMMING SOFTWARE

Then, using the Φ^0-order-$N/2$ constraint, we obtained a well-defined constraint optimization problem to find the matrix $\langle b,d,e,f\rangle_N$ giving a girth greater of equal than g_0: according to lemma 2.1, any cycle C of the protograph matrix of length $\ell < g_0$ should satisfy $\Phi(C) \neq 0$. The total number of constraints $\mathbb{N}l < g_0$ is given in the last line of Table I. It is noteworthy that some constraints are redundant. For example, the cycle $\mathcal{C}_{AECBEFB}$ of length $\ell = 14$ gives the same constraint as the cycle $\mathcal{C}_{AEFEC}$ of length $\ell' = 10$ since $\Phi(\mathcal{C}_{AECBEFB}) = \Phi(\mathcal{C}_{AEFEC}) = f - 2e$. Nevertheless, introducing redundant constraints in a constraint programming tool is not a problem.

Table II presents the number of solutions for a targeted girth $g_0 = 8$ up to $g_0 = 16$, with respect to the expansion size N. In the first column, we give the highest value of N for which no solution can be found for the corresponding girth. We also give the number of solutions for the next 10 values of N to show the evolution of the number of solutions with increasing N. Line 2 gives the number of solutions n_0 when only the additive transformation is used to reduce the search space (matrices of type $H = \langle b,d,e,f\rangle$) with no particular constraint on b, d, e and f), line 3 gives the number of solution n_1 when the Φ-order -$N/2$ is used, while line 4 shows the number of solutions n_2 when both Φ-order-$N/2$ constraint and multiplicative transformation are applied. The last line shows the final fraction of remaining search space $R = n_2/n_0$. For a given expansion factor N, if a solution with a girth g exist, this solution will be enumerated in the list of solutions obtained for a targeted girth $g_0 = g - 2$. For example, in Table II, among the $n_2 = 14$ solutions obtained for a targeted girth $g_0 = 8$ with an expansion factor of $N = 8$, two have a girth equal to 10. When $g_0 \geq 10$, any two cycles of length four have distinct values. The Φ^0-order-$N/2$ constraint can thus be expressed with strict inequality as $0 < \Phi(\mathcal{C}_{AB}) < \Phi(\mathcal{C}_{CD}) < \Phi(\mathcal{C}_{EF}) < N/2$. Then, each of the 48 additive and structural transformations gives a distinct solution. In that case, n_1 is just equal to $n_1 = n_0/48$. Moreover, due to the multiplicative transformation (Theorem 5.5), some of the leftover solutions can be proved equivalent, thus the reduction can be significantly greater than a factor 48. For example, in Table II when N equals 23, there are 23760 solutions when no equivalence is used. This number reduces to $n_1 = 23760/48 = 495$ when using Φ^0-order-$N/2$ equivalence. Among those 495 solutions, 470 are proved to be equivalent due to multiplicative transformation. Thus, only a maximum of $n_2 = 25$ out of 23760 are distinct solutions (all solutions have a distinct multiplicity spectrum), which correspond to only $R = 0.19\%$ of the initial space. Note that the computation time is also significantly reduced: 3.9 seconds to determine n_2 and its related solutions as against 97.6 s to determine n_0 and its related solutions for $N = 45$ and $g = 16$.

B. Elements of differentiation of the solutions

The above section presents the formulation of constraints forbidding cycles of a given size and shows the result of numerical experiments. From those experiments, we can highlight the reduction of the number of solutions found by using dominance breaking. Except for some rare cases, there exist several solutions for a given girth and expansion size and our method finds them all.

Due to the significant reduction in the number of solutions, it is now possible to analyze in more details the properties of each solution by computing the multiplicity $\mathcal{M}(g)$ of cycle of length g, and eventually, the multiplicity of greater length. Table III shows an example of the 5 solutions obtained for $g = 14$ and $N = 30$, giving the matrices and the values of $\mathcal{M}(14)$ and $\mathcal{M}(16)$.

One can note that, in this example, all solutions are distinct since they have different cycle multiplicities and that solution 4 gives the code with the best topological properties.

One of the most surprising results on Table II is that the number n_2 of distinct solutions does not always increase when N increases. For $g \geq 8$, the number n_2 of unique solutions for $N = 5, 6, \ldots 13$ is respectively $n_2 = 1, 7, 4, 17, 14, 37, 20, 99$ and 35. Even more surprising, for $g = 16$, there exist $n_2 = 4$ solutions when $N = 36$, none when $N = 37$ and $N = 38$, then 2 for $N = 39$, 6 for $N = 40$ and one for $N = 41$! In other words, the girth is not necessarily an increasing function of the expansion factor of the lifting factor N, since $g = 16$ for $N = 36$, $g = 14$ for $N = 37$ and $N = 38$, g equals 16 again for $N \geq 39$. This result is rather counter-intuitive so it has been carefully checked. It is worth mentioning that the method proposed here has been used to construct several good Non-Binary-LDPC matrices of several sizes and code rates available on line [30]).

Finally, Table VIII in the Appendix gives the best expanded matrices found for values of N between 4 and 43. For each value of N, a NB-LDPC codes over GF(64) is constructed. The choice of the Galois Field coefficients is carried out following the rules given in [31]. Figure 2 compares the energy per symbol versus the Energy of the noise (E_s/N_0, in dB) required to obtain a Frame Error Rate (FER) of 10^{-2} and 10^{-3} , respectively. The decoding algorithm is the Extended Min Sum (EMS) algorithm of parameter $n_m = 20$ [32] with 10 decoding iterations. The modulation is the Binary Phase Shift Keying (BPSK) modulation over the Additive White Gaussian Noise (AWGN) channel. The channel outputs are quantified on 5 bits before entering the decoder.

	N	< 4	4	5	6	7	8	9	10	11	12	13
$g_0 = 8$	n_0	0	2	16	92	288	702	1440	2648	4480	7130	10800
	n_1	0	2	2	14	18	55	67	150	175	335	378
	n_2	0	1	1	7	4	17	14	37	20	99	35
	R	N/A	50 %	6.25 %	7.61 %	1.39 %	2.42 %	0.97 %	1.40 %	0.45 %	1.39 %	0.32%
	N	< 8	8	9	10	11	12	13	14	15	16	17
$g_0 = 10$	n_0	0	48	336	480	1680	2160	5184	6096	12336	13968	24960
	n_1	0	1	7	10	35	45	108	127	257	291	520
	n_2	0	1	2	3	4	15	10	24	40	45	35
	R	N/A	2.08 %	0.59 %	0.62 %	0.24 %	0.69 %	0.19 %	0.39 %	0.32 %	0.32 %	0.14 %
	N	< 14	14	15	16	17	18	19	20	21	22	23
$g_0 = 12$	n_0	0	480	432	1056	1920	4032	5184	7296	12960	17520	23760
	n_1	0	10	9	22	40	84	108	152	270	365	495
	n_2	0	3	2	4	3	18	7	25	30	41	25
	R	N/A	0.62 %	0.46 %	0.38 %	0.16 %	0.45 %	0.13 %	0.34 %	0.23 %	0.23 %	0.11 %
	N	< 28	28	29	30	31	32	33	34	35	36	37
$g_0 = 14$	n_0	0	1440	1344	1632	5760	6528	12480	13824	30336	30048	60480
	n_1	0	30	28	34	120	136	260	288	632	626	1260
	n_2	0	3	1	5	4	12	14	18	28	57	35
	R	N/A	0.21 %	0.07 %	0.30 %	0.069 %	0.18 %	0.11 %	0.13 %	0.09 %	0.19 %	0.058 %
	N	< 36	36	37	38	39	40	41	42	43	44	45
$g_0 = 16$	n_0	0	1536	0	0	1728	2784	1920	5088	4032	15840	12096
	n_1	0	32	0	0	36	58	40	106	84	330	252
	n_2	0	4	0	0	2	6	1	10	2	18	13
	R	N/A	0.26 %	N/A	N/A	0.12 %	0.21 %	0.052 %	0.20 %	0.050 %	0.11 %	0.11 %

TABLE II

COMPARISON OF EXECUTION TIME AND NUMBER OF SOLUTIONS WITH AND WITHOUT Φ-ORDER-$N/2$ CONSTRAINT FOR $g_0 = 8, 10, 12, 14$ AND 16.

#	$\langle b, d, e, f \rangle_{30}$	$\mathcal{M}(14)$	$\mathcal{M}(16)$
1	$\langle 1, 3, 7, 19 \rangle_{30}$	180	825
2	$\langle 1, 5, 7, 16 \rangle_{30}$	180	840
3	$\langle 1, 6, 25, 8 \rangle_{30}$	180	810
4	$\langle 1, 9, 2, 13 \rangle_{30}$	90	1080
5	$\langle 3, 5, 4, 13 \rangle_{30}$	240	750

TABLE III

THE FIVE SOLUTIONS FOR $g = 14$, $N = 30$

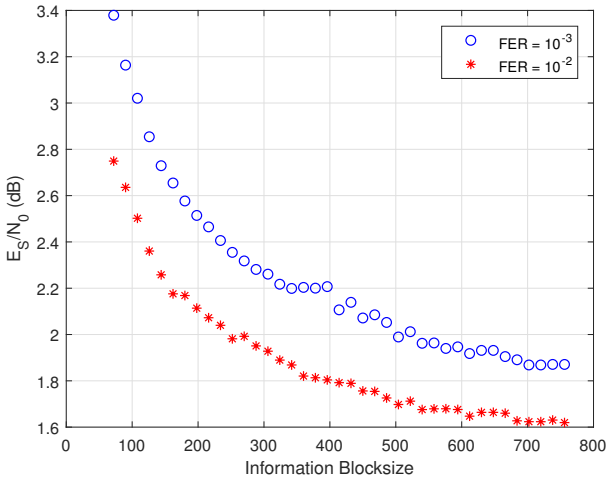


Fig. 2. Required E_s/N_0 to obtain a FER of 10^{-2} and 10^{-3} for NB-LDPC matrices over GF(64) constructed from matrices of Table VIII. The EMS decoding algorithm is used with 10 decoding iterations. Received samples are quantified on 5 bits.

C. Extension to the complete (3-L) QC-LDPC matrix

Before presenting the conclusion of this paper, we give an additional example to highlight the efficiency of the proposed method to construct QC-LDPC matrices. Let us consider the set of complete prototype matrices $\mathbf{H}_{(3,L)}$ of size $3 \times L$, ($\mathbf{H}_{(3,L)}$ is composed of 3 lines of L ones)). The problem of finding the minimum lifting factor N giving a QC-LDPC matrix of girth 8 and 10 is well studied in the literature. Table IV shows the evolution of the minimum value of N cited in the literature for L varying from 4 to 12 and girth g equal to 8.

These solutions are found by a heuristic search with no proof of optimality. Using the theory and the tools developed in the present study for this problem, we were rapidly able (in one day) to:

1) Enumerate all the cycles of length 4 and 6 of the $\mathbf{H}_{(3,L)}$ matrices.
2) Prove the optimality of the existing solutions for $L = 4$ up to $L = 10$ [1].
3) Find a new solution for $L = 11$ with an expansion factor of $N = 40$.

This new matrix $\mathcal{H}^{40}_{(3,11)}$ generated with the constraint programming tool is

$$\mathcal{H}^{40}_{(3,11)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 5 & 8 & 12 & 16 & 19 & 21 & 23 & 24 \\ 0 & 2 & 7 & 18 & 33 & 15 & 38 & 39 & 10 & 31 & 35 \end{bmatrix} \quad (20)$$

Note that we do not have any proof that $N = 40$ is optimal (i.e. minimum expansion factor) for $L = 11$. Without a deeper study of this problem involving symmetry breaking, the running time required to demonstrate optimality is too long. Property 5.6 and Theorem 4.3 would help in finding new symmetries to reduce the size of the space to explore. We leave this as an open question.

For the girth $g = 10$, the minimum lifting factor N is greatly reduced compared to the state of the art. For example, for $L =$

[1] Taking the constraints $\mathcal{H}^{N}_{(3,L)}(2,2)$ as a divisor of N thanks to multiplicative transformation and $\mathcal{H}^{N}_{(3,L)}(2,j) \leq \mathcal{H}^{40}_{(3,L)}(2,j+1)$, $j = 1, \ldots, L-1$ thanks to column permutations of the matrix.

		$L=4$	$L=5$	$L=6$	$L=7$	$L=8$	$L=9$	$L=10$	$L=11$	$L=12$
2004	[20]	9	14	18	21	26	33	39	46	54
2006	[16]	9	13	18	22	27	34	40	49	55
2013	[15]	9	13	18	21	25	30	35	41	47
2015	[33]	9	13	18	21	25	30	35	40	46
2017	[19]	9	13	18	21	25	30	35	41	45

TABLE IV
RANDOMLY FOUND SOLUTION FOR A COMPLETE PROTOMATRIX, WITH $J=3$ LINES AND L COLUMNS

		$L=4$	$L=5$	$L=6$	$L=7$	$L=8$	$L=9$	$L=10$	$L=11$
2013	Lower bound [34]	37	61	91	127	168	217	271	331
2008	[35]	39	63	103	160	233	329	439	577
2012	[36]	37	61	101	159	219	319	439	560
2016	[33];[37]	37	61	91	155	227	323	429	571
2017	[21]	37	61	91	-	-	-	-	-
2018	Proposed	37	61	91	139	201	280	383	503

TABLE V
SMALLEST LIFTING FACTOR N REQUIRED TO OBTAINED A GIRTH $g=10$ FOR A COMPLETE (3,L) QC-LDPC MATRIX

11, N is reduced between 2008 and 2016 from $N=577$ down to $N=560$. In the present study, we propose a new value of N equal to 506! Smallest values of N are also obtained for L = 7, 8, 9 and 10 as shown in Table V. The corresponding matrices are

$$\mathcal{H}^{139}_{(3,7)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 26 & 31 & 54 & 100 & 106 \\ 0 & 3 & 15 & 104 & 7 & 44 & 122 \end{bmatrix} \quad (21)$$

$$\mathcal{H}^{201}_{(3,8)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 5 & 20 & 32 & 99 & 105 & 158 \\ 0 & 3 & 16 & 86 & 155 & 45 & 133 & 194 \end{bmatrix} \quad (22)$$

$$\mathcal{H}^{280}_{(3,9)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 5 & 12 & 97 & 112 & 162 & 233 & 249 \\ 0 & 3 & 13 & 41 & 207 & 75 & 101 & 216 & 158 \end{bmatrix} \quad (23)$$

$$\mathcal{H}^{383}_{(3,10)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 5 & 12 & 32 & 55 & 160 & 237 & 261 & 305 \\ 0 & 3 & 13 & 29 & 74 & 246 & 111 & 170 & 350 & 132 \end{bmatrix} \quad (24)$$

and

$$\mathcal{H}^{503}_{(3,11)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 5 & 12 & 32 & 50 & 184 & 269 & 385 & 410 & 432 \\ 0 & 3 & 13 & 29 & 68 & 109 & 276 & 415 & 216 & 195 & 371 \end{bmatrix} \quad (25)$$

VII. CONCLUSION

In this paper, we have defined the notion of equivalence classes between QC-LDPC matrices. We have proposed three transformations that preserve the equivalence between matrices: the additive transformation (already known), the structural transformation and the multiplicative transformation. As an example, we have applied these transformations to a particular 3-row, 6-column protomatrix with variable node degree of 2 and check node degree 4. For this problem, we proposed criteria to select a single element for each equivalence class of matrices, thus allowing a very fast exhaustive exploration. In fact, since an equivalent class can contain from a few tens up to a few thousand elements, the search space is reduced accordingly. Finally, we make use of a tool from the constraint programming literature to prove and find solutions to the construction problem.

We should note that the new equivalence relations between QC-LDPC matrices are very general and can be used for any construction of QC-LDPC matrices from the lifting of a prototype matrix. Finally, we conclude with an open question: is it possible to use additive, structural and multiplicative transformations on their own to generate the equivalent class of any QC matrix?

ACKNOWLEDGMENT

The authors would like to thank several people that have also help for the realization of this paper: Ahmed Abdmouleh for his proposal of the prototype matrix, Marc Sevaux for his reading of a very early version of the paper, Hassan Harb, Cédric Marchand, Titouan Gendron and Franklin Cochachin to have developed some useful tools and performed simulations. A special thanks to Xavier Giraud, Valentin Savin, Alireza Tasdighi and the anonymous reviewers for their reading and their suggestions to improve the paper, Laura Conde-Canencia for her reading of the paper and Michael Carpenter for copy-editing of the final version. Part of the work uses ressources funded by the Région Bretagne and the ANR through the CPER Sophie.

APPENDIX

REFERENCES

[1] DVB standard committee, "DVB-S2." https://www.dvb.org/standards/dvb-s2.
[2] IEEE standard committee, "WI-FI." http://ieeexplore.ieee.org/browse/standards/get-program/page/series?id=68.
[3] 3GPP, "5G standard." https://www.3gpp.org.
[4] R. G. Gallager, "Low-density parity-check codes," 1963.
[5] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, pp. 1645–, Aug 1996.
[6] E. Boutillon, C. Douillard, and G. Montorsi, "Iterative Decoding of Concatenated Convolutional Codes: Implementation Issues," *Proceedings of the IEEE*, vol. 95, pp. 1201–1227, June 2007.
[7] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over GF(q)," in *1998 Information Theory Workshop (Cat. No.98EX131)*, pp. 70–71, Jun 1998.

#	π_r	π_c	$\langle b,d,e,f\rangle_N \equiv$	$\langle 1,7,13,17\rangle_N$	b divisor of N
1	$\{1,2,3\}$	$\{1,2,3,4,5,6\}$	$\langle +b,+d,+e,+f\rangle_N$	$\langle 1,7,13,17\rangle_{35}$	$\langle 1,7,13,17\rangle_{35}$
2	$\{1,2,3\}$	$\{1,2,3,4,6,5\}$	$\langle +b,+d,+f,+e\rangle_N$	$\langle 1,7,17,13\rangle_{35}$	$\langle 1,7,17,13\rangle_{35}$
3	$\{1,2,3\}$	$\{1,2,4,3,5,6\}$	$\langle +b,-d,-d+e,-d+f\rangle_N$	$\langle 1,28,6,10\rangle_{35}$	$\langle 1,28,6,10\rangle_{35}$
4	$\{1,2,3\}$	$\{1,2,4,3,6,5\}$	$\langle +b,-d,-d+f,-d+e\rangle_N$	$\langle 1,28,10,6\rangle_{35}$	$\langle 1,28,10,6\rangle_{35}$
5	$\{1,2,3\}$	$\{2,1,3,4,5,6\}$	$\langle -b,+d,+b+e,+b+f\rangle_N$	$\langle 34,7,14,18\rangle_{35}$	$\langle 1,28,21,17\rangle_{35}$
6	$\{1,2,3\}$	$\{2,1,3,4,6,5\}$	$\langle -b,+d,+b+f,+b+e\rangle_N$	$\langle 34,7,18,14\rangle_{35}$	$\langle 1,28,17,21\rangle_{35}$
7	$\{1,2,3\}$	$\{2,1,4,3,5,6\}$	$\langle -b,-d,+b-d+e,+b-d+f\rangle_N$	$\langle 34,28,7,11\rangle_{35}$	$\langle 1,7,28,24\rangle_{35}$
8	$\{1,2,3\}$	$\{2,1,4,3,6,5\}$	$\langle -b,-d,+b-d+f,+b-d+e\rangle_N$	$\langle 34,28,11,7\rangle_{35}$	$\langle 1,7,24,28\rangle_{35}$
9	$\{2,1,3\}$	$\{1,2,5,6,3,4\}$	$\langle -b,-e+f,-e,+d-e\rangle_N$	$\langle 34,4,22,29\rangle_{35}$	$\langle 1,31,13,6\rangle_{35}$
10	$\{2,1,3\}$	$\{1,2,5,6,4,3\}$	$\langle -b,+e-f,-f,+d-f\rangle_N$	$\langle 34,31,18,25\rangle_{35}$	$\langle 1,4,17,10\rangle_{35}$
11	$\{2,1,3\}$	$\{1,2,6,5,3,4\}$	$\langle -b,-e+f,+d-e,-e\rangle_N$	$\langle 34,4,29,22\rangle_{35}$	$\langle 1,31,6,13\rangle_{35}$
12	$\{2,1,3\}$	$\{1,2,6,5,4,3\}$	$\langle -b,+e-f,+d-f,-f\rangle_N$	$\langle 34,31,25,18\rangle_{35}$	$\langle 1,4,10,17\rangle_{35}$
13	$\{2,1,3\}$	$\{2,1,5,6,3,4\}$	$\langle +b,-e+f,-b-e,-b+d-e\rangle_N$	$\langle 1,4,21,28\rangle_{35}$	$\langle 1,4,21,28\rangle_{35}$
14	$\{2,1,3\}$	$\{2,1,5,6,4,3\}$	$\langle +b,+e-f,-b-f,-b+d-f\rangle_N$	$\langle 1,31,17,24\rangle_{35}$	$\langle 1,31,17,24\rangle_{35}$
15	$\{2,1,3\}$	$\{2,1,6,5,3,4\}$	$\langle +b,-e+f,-b+d-e,-b-e\rangle_N$	$\langle 1,4,28,21\rangle_{35}$	$\langle 1,4,28,21\rangle_{35}$
16	$\{2,1,3\}$	$\{2,1,6,5,4,3\}$	$\langle +b,+e-f,-b+d-f,-b-f\rangle_N$	$\langle 1,31,24,17\rangle_{35}$	$\langle 1,31,24,17\rangle_{35}$
17	$\{2,3,1\}$	$\{3,4,5,6,1,2\}$	$\langle -e+f,-b,+e,-d+e\rangle_N$	$\langle 4,34,13,6\rangle_{35}$	$\langle 1,26,12,19\rangle_{35}$
18	$\{2,3,1\}$	$\{3,4,5,6,2,1\}$	$\langle +e-f,-b,+f,-d+f\rangle_N$	$\langle 31,34,17,10\rangle_{35}$	$\langle 1,9,22,15\rangle_{35}$
19	$\{2,3,1\}$	$\{3,4,6,5,1,2\}$	$\langle -e+f,-b,-d+e,+e\rangle_N$	$\langle 4,34,6,13\rangle_{35}$	$\langle 1,26,19,12\rangle_{35}$
20	$\{2,3,1\}$	$\{3,4,6,5,2,1\}$	$\langle +e-f,-b,-d+f,+f\rangle_N$	$\langle 31,34,10,17\rangle_{35}$	$\langle 1,9,15,22\rangle_{35}$
21	$\{2,3,1\}$	$\{4,3,5,6,1,2\}$	$\langle -e+f,+b,+b+e,+b-d+e\rangle_N$	$\langle 4,1,14,7\rangle_{35}$	$\langle 1,9,21,28\rangle_{35}$
22	$\{2,3,1\}$	$\{4,3,5,6,2,1\}$	$\langle +e-f,+b,+b+f,+b-d+f\rangle_N$	$\langle 31,1,18,11\rangle_{35}$	$\langle 1,26,13,6\rangle_{35}$
23	$\{2,3,1\}$	$\{4,3,6,5,1,2\}$	$\langle -e+f,+b,+b-d+e,+b+e\rangle_N$	$\langle 4,1,7,14\rangle_{35}$	$\langle 1,9,28,21\rangle_{35}$
24	$\{2,3,1\}$	$\{4,3,6,5,2,1\}$	$\langle +e-f,+b,+b-d+f,+b+f\rangle_N$	$\langle 31,1,11,18\rangle_{35}$	$\langle 1,26,6,13\rangle_{35}$

TABLE VI

EQUIVALENT MATRICES THROUGH STRUCTURAL AND ADDITIVE TRANSFORMATIONS. PART I: LINES 1 TO 24.

#	π_r	π_c	$\langle b,d,e,f\rangle_N \equiv$	$\langle 1,7,13,17\rangle_N$	b divisor of N
25	$\{1,3,2\}$	$\{3,4,1,2,5,6\}$	$\langle +d,+b,-e,-f\rangle_N$	$\langle 7,1,22,18\rangle_{35}$	$\langle 7,1,22,18\rangle_{35}$
26	$\{1,3,2\}$	$\{3,4,1,2,6,5\}$	$\langle +d,+b,-f,-e\rangle_N$	$\langle 7,1,18,22\rangle_{35}$	$\langle 7,1,18,22\rangle_{35}$
27	$\{1,3,2\}$	$\{3,4,2,1,5,6\}$	$\langle -d,+b,+d-e,+d-f\rangle_N$	$\langle 28,1,29,25\rangle_{35}$	$\langle 7,34,6,10\rangle_{35}$
28	$\{1,3,2\}$	$\{3,4,2,1,6,5\}$	$\langle -d,+b,+d-f,+d-e\rangle_N$	$\langle 28,1,25,29\rangle_{35}$	$\langle 7,34,10,6\rangle_{35}$
29	$\{1,3,2\}$	$\{4,3,1,2,5,6\}$	$\langle +d,-b,-b-e,-b-f\rangle_N$	$\langle 7,34,21,17\rangle_{35}$	$\langle 7,34,21,17\rangle_{35}$
30	$\{1,3,2\}$	$\{4,3,1,2,6,5\}$	$\langle +d,-b,-b-f,-b-e\rangle_N$	$\langle 7,34,17,21\rangle_{35}$	$\langle 7,34,17,21\rangle_{35}$
31	$\{1,3,2\}$	$\{4,3,2,1,5,6\}$	$\langle -d,-b,-b+d-e,-b+d-f\rangle_N$	$\langle 28,34,28,24\rangle_{35}$	$\langle 7,1,7,11\rangle_{35}$
32	$\{1,3,2\}$	$\{4,3,2,1,6,5\}$	$\langle -d,-b,-b+d-f,-b+d-e\rangle_N$	$\langle 28,34,24,28\rangle_{35}$	$\langle 7,1,11,7\rangle_{35}$
33	$\{3,1,2\}$	$\{5,6,1,2,3,4\}$	$\langle -d,+e-f,+e,+b+e\rangle_N$	$\langle 28,31,13,14\rangle_{35}$	$\langle 7,4,22,21\rangle_{35}$
34	$\{3,1,2\}$	$\{5,6,1,2,4,3\}$	$\langle -d,-e+f,+f,+b+f\rangle_N$	$\langle 28,4,17,18\rangle_{35}$	$\langle 7,31,18,17\rangle_{35}$
35	$\{3,1,2\}$	$\{5,6,2,1,3,4\}$	$\langle +d,+e-f,-d+e,+b-d+e\rangle_N$	$\langle 7,31,6,7\rangle_{35}$	$\langle 7,31,6,7\rangle_{35}$
36	$\{3,1,2\}$	$\{5,6,2,1,4,3\}$	$\langle +d,-e+f,-d+f,+b-d+f\rangle_N$	$\langle 7,4,10,11\rangle_{35}$	$\langle 7,4,10,11\rangle_{35}$
37	$\{3,1,2\}$	$\{6,5,1,2,3,4\}$	$\langle -d,+e-f,+b+e,+e\rangle_N$	$\langle 28,31,14,13\rangle_{35}$	$\langle 7,4,21,22\rangle_{35}$
38	$\{3,1,2\}$	$\{6,5,1,2,4,3\}$	$\langle -d,-e+f,+b+f,+f\rangle_N$	$\langle 28,4,18,17\rangle_{35}$	$\langle 7,31,17,18\rangle_{35}$
39	$\{3,1,2\}$	$\{6,5,2,1,3,4\}$	$\langle +d,+e-f,+b-d+e,-d+e\rangle_N$	$\langle 7,31,7,6\rangle_{35}$	$\langle 7,31,7,6\rangle_{35}$
40	$\{3,1,2\}$	$\{6,5,2,1,4,3\}$	$\langle +d,-e+f,+b-d+f,-d+f\rangle_N$	$\langle 7,4,11,10\rangle_{35}$	$\langle 7,4,11,10\rangle_{35}$
41	$\{3,2,1\}$	$\{5,6,3,4,1,2\}$	$\langle +e-f,-d,-e,-b-e\rangle_N$	$\langle 31,28,22,21\rangle_{35}$	$\langle 1,28,12,21\rangle_{35}$
42	$\{3,2,1\}$	$\{5,6,3,4,2,1\}$	$\langle -e+f,-d,-f,-b-f\rangle_N$	$\langle 4,28,18,17\rangle_{35}$	$\langle 1,7,22,13\rangle_{35}$
43	$\{3,2,1\}$	$\{5,6,4,3,1,2\}$	$\langle +e-f,+d,+d-e,-b+d-e\rangle_N$	$\langle 31,7,29,28\rangle_{35}$	$\langle 1,7,19,28\rangle_{35}$
44	$\{3,2,1\}$	$\{5,6,4,3,2,1\}$	$\langle -e+f,+d,+d-f,-b+d-f\rangle_N$	$\langle 4,7,25,24\rangle_{35}$	$\langle 1,28,15,6\rangle_{35}$
45	$\{3,2,1\}$	$\{6,5,3,4,1,2\}$	$\langle +e-f,-d,-b-e,-e\rangle_N$	$\langle 31,28,21,22\rangle_{35}$	$\langle 1,28,21,12\rangle_{35}$
46	$\{3,2,1\}$	$\{6,5,3,4,2,1\}$	$\langle -e+f,-d,-b-f,-f\rangle_N$	$\langle 4,28,17,18\rangle_{35}$	$\langle 1,7,13,22\rangle_{35}$
47	$\{3,2,1\}$	$\{6,5,4,3,1,2\}$	$\langle +e-f,+d,-b+d-e,+d-e\rangle_N$	$\langle 31,7,28,29\rangle_{35}$	$\langle 1,7,28,19\rangle_{35}$
48	$\{3,2,1\}$	$\{6,5,4,3,2,1\}$	$\langle -e+f,+d,-b+d-f,+d-f\rangle_N$	$\langle 4,7,24,25\rangle_{35}$	$\langle 1,28,6,15\rangle_{35}$

TABLE VII

EQUIVALENT MATRICES THROUGH STRUCTURAL AND ADDITIVE TRANSFORMATIONS. PART II: LINES 25 TO 48).

[8] "Consultative Committee for Space Data Systems (CCSDS), Telecommand Sync and Channel Coding Specification using advanced Block Codes." https://https://public.ccsds.org/.

[9] K. Liu, Q. Huang, S. Lin, and K. A. S. Abdel-Ghaffar, "Quasi-Cyclic LDPC codes: Construction and rank analysis of their parity-check matrices," in *2012 Information Theory and Applications Workshop, ITA 2012, San Diego, CA, USA, February 5-10, 2012*, pp. 227–233, 2012.

[10] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching Protograph Codes," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 876–888, Aug. 2009.

[11] D. Divsalar, S. Dolinar, and C. C. R. Jones, "Low-rate LDPC codes with simple protograph structure," *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, pp. 1622–1626, 2005.

[12] J. Thorpe, K. Andrews, and S. Dolinar, "Methodologies for designing LDPC codes using protographs and circulants," *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, pp. 238–, 2004.

[13] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "On the girth of (3, L) Quasi-Cyclic LDPC Codes based on Complete Protographs," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 431–435, June 2015.

[14] G. Zhang, R. Sun, and X. Wang, "Construction of Girth-Eight QC-LDPC Codes from Greatest Common Divisor," *IEEE Communications Letters*, vol. 17, no. 2, pp. 369–372, 2013.

[15] Y. Wang, J. S. Yedidia, and S. C. Draper, "Construction of high-girth QC-LDPC codes," in *2008 5th International Symposium on Turbo Codes and Related Topics*, pp. 180–185, Sept 2008.

[16] M. E. O'Sullivan, "Algebraic construction of sparse matrices with large girth," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 718–727, 2006.

[17] J. Zhang and G. Zhang, "Deterministic Girth-Eight QC-LDPC Codes with Large Column Weight," *IEEE Communications Letters*, vol. 18, no. 4, pp. 656–659, 2014.

[18] S. Vafi and N. Majid, "Combinatorial design based Quasi Cyclic LDPC codes with girth eight," *Digital Communications and Networks (2018)*,

g	$\langle b,d,e,f\rangle_N$	$\mathcal{M}(8)$	$\mathcal{M}(10)$	$\mathcal{M}(12)$	$\mathcal{M}(14)$	$\mathcal{M}(16)$	$\mathcal{M}(18)$	$\mathcal{M}(20)$
8	$\langle 2,2,1,3\rangle_4$	30	0	112	0	1038	0	7196
8	$\langle 1,1,2,3\rangle_5$	15	30	80	180	382	1128	3422
8	$\langle 1,1,2,4\rangle_6$	6	48	86	120	392	1228	3282
8	$\langle 1,1,2,4\rangle_7$	7	35	91	147	422	1208	2881
10	$\langle 1,2,3,6\rangle_8$	0	48	80	144	486	1072	2929
10	$\langle 1,2,3,6\rangle_9$	0	45	75	144	513	1122	2798
10	$\langle 1,2,3,6\rangle_{10}$	0	40	75	150	525	1150	2694
10	$\langle 1,2,3,6\rangle_{11}$	0	33	77	165	528	1177	2593
10	$\langle 1,4,1,6\rangle_{12}$	0	24	82	192	522	1128	2557
10	$\langle 1,2,5,8\rangle_{13}$	0	13	104	156	624	1053	2444
12	$\langle 1,2,6,9\rangle_{14}$	0	0	126	168	602	952	2744
12	$\langle 3,5,13,4\rangle_{15}$	0	0	95	240	495	1200	2736
12	$\langle 1,2,5,12\rangle_{16}$	0	0	96	224	520	1152	2688
12	$\langle 1,2,5,13\rangle_{17}$	0	0	102	170	646	952	2924
12	$\langle 1,2,5,13\rangle_{18}$	0	0	90	180	657	1074	2619
12	$\langle 1,2,5,14\rangle_{19}$	0	0	95	114	798	912	2907
12	$\langle 1,4,6,14\rangle_{20}$	0	0	70	180	700	1120	2738
12	$\langle 1,2,5,12\rangle_{21}$	0	0	49	231	609	1302	2562
12	$\langle 1,3,9,14\rangle_{22}$	0	0	55	154	814	1034	2739
12	$\langle 1,3,9,14\rangle_{23}$	0	0	46	138	897	1012	2783
12	$\langle 1,2,6,14\rangle_{24}$	0	0	44	216	624	1320	2496
12	$\langle 1,2,6,17\rangle_{25}$	0	0	50	125	850	975	3000
12	$\langle 1,2,6,15\rangle_{26}$	0	0	39	182	676	1274	2704
12	$\langle 1,2,6,18\rangle_{27}$	0	0	27	135	891	1062	2916
14	$\langle 1,6,23,8\rangle_{28}$	0	0	0	168	882	1288	2478
14	$\langle 1,4,6,15\rangle_{29}$	0	0	0	174	812	1421	2349
14	$\langle 1,9,2,13\rangle_{30}$	0	0	0	90	1080	1050	3195
14	$\langle 1,3,9,14\rangle_{31}$	0	0	0	124	992	1147	2697
14	$\langle 1,8,11,26\rangle_{32}$	0	0	0	64	1176	832	3264
14	$\langle 1,10,2,14\rangle_{33}$	0	0	0	99	1023	1155	2871
14	$\langle 1,3,8,23\rangle_{34}$	0	0	0	102	918	1326	2601
14	$\langle 1,4,12,18\rangle_{35}$	0	0	0	70	1015	1085	3150
16	$\langle 1,11,6,19\rangle_{36}$	0	0	0	0	1296	648	3834
14	$\langle 1,3,12,17\rangle_{37}$	0	0	0	37	1184	851	3219
14	$\langle 1,3,10,15\rangle_{38}$	0	0	0	114	988	1140	2565
16	$\langle 1,12,7,21\rangle_{39}$	0	0	0	0	1209	897	3393
16	$\langle 4,10,7,19\rangle_{40}$	0	0	0	0	1330	0	5760
16	$\langle 1,3,14,19\rangle_{41}$	0	0	0	0	1312	574	3567
16	$\langle 1,3,14,19\rangle_{42}$	0	0	0	0	1260	728	3528
16	$\langle 1,3,14,19\rangle_{43}$	0	0	0	0	1204	774	3483

TABLE VIII
BEST MATRICES FOUNDED FOR SEVERAL EXPANSION FACTORS N.

2018. doi: 10.1016/j.dcan.2018.01.001.

[19] A. Tasdighi, A. H. Banihashemi, and M. Sadeghi, "Symmetrical Constructions for Regular Girth-8 QC-LDPC Codes," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 14–22, 2017.

[20] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Transactions on Information Theory*, vol. 50, pp. 1788–1793, Aug. 2004.

[21] A. Tasdighi, A. H. Banihashemi, and M. R. Sadeghi, "Efficient Search of Girth-Optimal QC-LDPC Codes," *IEEE Transactions on Information Theory*, vol. 62, pp. 1552–1564, April 2016.

[22] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, pp. 457–458, Mar 1997.

[23] F. R. Kschischang, B. J. Frey, and H. . Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[24] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," tech. rep., JPL IPN Progress Report 42-154, 2003.

[25] D. G. M. Mitchell, R. Smarandache, and D. J. Costello, "Quasi-Cyclic LDPC Codes Based on Pre-Lifted Protographs," *IEEE Transactions on Information Theory*, vol. 60, pp. 5856–5874, Oct 2014.

[26] R. Smarandache and P. O. Vontobel, "On regular quasicyclic LDPC codes from binomials," in *IEEE International Symposium on Information Theory, 2004. ISIT 2004.*, p. 275, 2004.

[27] G. Chu and P. J. Stuckey, "Dominance breaking constraints," *Constraints*, vol. 20, no. 2, pp. 155–182, 2015.

[28] X. Giraud, "Comments on Additive, Structural and Multiplicative transformations for the construction of Quasi Cyclic LDPC matrices." http://www-labsticc.univ-ubs.fr/nb_ldpc/. [online since 2018].

[29] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, "Design of Finite-Length Irregular Protograph Codes with Low Error Floors over the Binary-Input AWGN Channel Using Cyclic Liftings," *IEEE Transactions on Communications*, vol. 60, pp. 902–907, April 2012.

[30] C. Marchand and et al., "Non-Binary WEB page of Lab-STICC." http://www-labsticc.univ-ubs.fr/nb_ldpc/. [online since 2015].

[31] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular (2,dc)-LDPC codes over GF(q) using their binary images," *IEEE Transactions on Communications*, vol. 56, pp. 1626–1635, October 2008.

[32] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields," *IEEE Transactions on Communications*, vol. 58, pp. 1365–1375, May 2010.

[33] M. Diouf, *Conception avancée des codes LDPC binaires pour des applications pratiques*. PhD thesis, Université de Cergy Pontoise ; Université de Cheikh Anta DIOP, 2015.

[34] M. Karimi and A. H. Banihashemi, "On the Girth of Quasi-Cyclic Protograph LDPC Codes," *IEEE Transactions on Information Theory*, vol. 59, pp. 4542–4552, July 2013.

[35] Y. Wang, J. S. Yedidia, and S. C. Draper, "Construction of high-girth QC-LDPC codes," in *2008 5th International Symposium on Turbo Codes and Related Topics*, pp. 180–185, Sept 2008.

[36] I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, and R. V. Satyukov, "Searching for Voltage Graph-Based LDPC Tailbiting Codes With Large Girth," *IEEE Transactions on Information Theory*, vol. 58, pp. 2265–2279, April 2012.

[37] M. Diouf, D. Declercq, M. Fossorier, S. Ouya, and B. Vasić, "Improved PEG construction of large girth QC-LDPC codes," in *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, pp. 146–150, Sept 2016.