Additive, Structural and Multiplicative Transformations for the Construction of Quasi-Cyclic LDPC matrices
Alban Derrien, Emmanuel Boutillon, Audrey Cerqueus

To cite this version:
Alban Derrien, Emmanuel Boutillon, Audrey Cerqueus. Additive, Structural and Multiplicative Transformations for the Construction of Quasi-Cyclic LDPC matrices, IEEE Transactions on Communications, Institute of Electrical and Electronics Engineers, 2019, 67 (4), pp.2647-2659. 10.1109/TCOMM.2018.2890251 . hal-01950474

HAL Id: hal-01950474
https://hal.archives-ouvertes.fr/hal-01950474
Submitted on 10 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 of check nodes connected to a given variable node and the number 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 $M(g)$.

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 $d_v$ non-zero values and each column of $H$ contains $d_c$ 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 $H$ 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]. The simplest protograph matrix for $d_v = 2$, $d_c = 4$ is the matrix $H_2$ of size $J = 2$, $L = 4$ defined as

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \quad (1)$$

Throughout this paper, we illustrate the proposed method using the example of QC-LDPC matrices with a rate 1/2 generated from the matrix $H_3$ of size $J = 3$, $L = 6$, defined as

$$H_3 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (2)$$

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 $H_3$ is unique up to rows/columns permutation (see section III).

The lifting process generates a matrix of size $(JN, LN)$ from $H$ by replacing each 0 value in $H$ 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

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

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 $H$, the factor of expansion $N$ and by the shift values associated with each non-zero value of the circulant permutation matrices $H_a$.

For example, matrix $H$ obtained from $H_3$ is defined as

$$H = \begin{bmatrix} I^N_{a_1} & I^N_{b_1} & I^N_{c_1} & I^N_{d_1} & O^N & O^N \\ I^N_{a_2} & I^N_{b_2} & O^N & I^N_{d_2} & O^N & O^N \\ O^N & O^N & I^N_{e_3} & I^N_{d_3} & I^N_{e_3} & I^N_{f_3} \end{bmatrix}. \quad (4)$$

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 doubly 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

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

where $H(i, j)$ represents $I_a^N(j, i)$ if $H(i, j)$ is a positive integer, or $O^N$ if $H(i, j) = -1$. 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 vertical vertices [4]. The construction objective in the lifting process is to maximize the girth of the Tanner graph.

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

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

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 3). Since the graph is bipartite, the length $\ell$ of a cycle is always even. Thus, the direction of the cycle does not impact the value of $\phi(C)$ since $(-1)^{\ell-1} = (-1)^\ell$.

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 $\pi$ of size $n$ is a bijection function of set $\mathcal{N} = \{1, 0, 1, \ldots, n-1\}$ to itself: $\pi(\mathcal{N}) = \{\pi(0), \pi(1), \ldots, \pi(n-1)\}$. The identity permutation is denoted $I_d$. It is possible to represent the permutation $\pi$ by a $(n, n)$ matrix, called the permutation matrix $P_{\pi}$, defined as $P_{\pi}(i, j) = 1$ if $j = \pi(i)$, 0 otherwise. The inverse of the permutation will be denoted $\pi^{-1}$ and verifies $\pi^{-1}(\pi(i)) = i$. Moreover, $(P_\pi)^{-1} = P_{\pi^{-1}}$ and $P_{\pi^{-1}} P_{\pi} = P_{\pi} P_{\pi^{-1}} = I_n$ with $I_n$ 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 $(n \times n)$ are to be called “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 equivalence 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 ($A = P_{r_1} \times B \times P_{c_1}$ and $B = P_{r_2} \times C \times P_{c_2}$ implies $A = P_{r_1} P_{r_2} \times C \times P_{c_1} P_{c_2}$).

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

To summarize, if $A \equiv_{\mathcal{H}} B$, 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 $\pi_a^r$ be the permutation over $\{0, 1, \ldots, n-1\}$ defined as $\pi_a^r(i) = i + a \mod n$. The permutation matrix $P_{\pi_a^r}$ associated with $\pi_a^r$ is also equal to the circulant matrix $P_{\pi_a} = P_{\pi_a^0}$.

Property 3.2: The product of two circulant matrices $P_{\pi_a}^r$ and $P_{\pi_b}^s$ is equal to the circulant matrix $P_{\pi_a}^r P_{\pi_b}^s = P_{\pi_a+b}^{r+s}$ [25].

Property 3.3 (Row rearrangement): Let $M$ be an $n \times m$ matrix and $\pi_r$ 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 $\pi_r$.

Property 3.4 (Column rearrangement): Let $M$ be a $n \times m$ matrix and $\pi_c$ 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 $\pi_c^{-1}$.

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 $H_r^+$ with $H_r^+ = H(i, j) + r(i)$ $\forall i \in \{1, 2, \ldots, J\}$, $j \in \{1, 2, \ldots, L\}$.

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

$$
C_r^+ = \text{diag}(P_{c(1)}, P_{c(2)}, \ldots, P_{c(L)}),
$$

with $c = (c(1), c(2), \ldots, c(L))$ representing a vector of integers.

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

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

$$
H_{r_1}^+ = \begin{bmatrix}
0 & b_1-a_1 & c_1-a_1 & d_1-a_1 & \cdots \\
0 & b_2-a_2 & c_2-a_2 & d_2-a_2 & \cdots \\
& & & & \\
0 & b_3 & c_3 & d_3 & e_3 & f_3
\end{bmatrix}
$$ (7)
Then, multiplying $H_{\pi}^+ \cdot c^+$ by the $(6N,6N)$ permutation matrix $C_6^+$ with $c = (0, a_1-b_1, a_1-c_1, a_1-d_1, a_2-c_2, a_2-f_2)$, gives $H_{\pi}^+ \cdot c^+ = H_{\pi}^+ \times C_6^+$. According to property 3.6, $H_{\pi}^+ \cdot c^+$ can be represented as:

$$
H_{\pi}^+ \cdot c^+ = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & b_2-a_2 + a_1 - b_1 & c_3-a_1-c_1 & d_3-a_1-d_1 & e_3-a_2-e_2 & f_3-a_2-f_2 \\
\end{bmatrix}
$$

(8)

Finally, multiplying $H_{\pi}^+ \cdot c^+$ on the left by $C_2^+$ with $c = (0, 0, -c_3 - a_1 + c_1)$ gives $H_{\pi}^+ \cdot c^+ \cdot x_2^+ = C_2^+ \times H_{\pi}^+ \cdot c^+$. According to property 3.6, $H_{\pi}^+ \cdot c^+ \cdot x_2^+$ can be represented as

$$
H_{\pi}^+ \cdot c^+ \cdot x_2^+ = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & b & c_3-a_1-c_1 & d_3-a_1-d_1 & e_3-a_2-e_2 & f_3-a_2-f_2 \\
\end{bmatrix}
$$

(9)

where

$$
\begin{cases}
    b = b_2 - a_2 + a_1 - b_1, \\
    d = d_3 - d_1 - c_3 + c_1, \\
    e = c_3 + a_2 - c_2 - c_3 - a_1 + c_1, \\
    f = f_3 + a_2 - f_2 - c_3 - a_1 + c_1.
\end{cases}
$$

Property 3.7 (bdef-pattern): Any matrix $H \in \mathbb{H}_N^N$ is equivalent to a matrix of type (9). Thus, the study of the set $\mathbb{H}_N^N$ can be restricted to the study of matrices defined in (9). To simplify notation, a matrix $H \in \mathbb{H}_N^N$ can be also represented by the 4-tuple $H = (b, d, e, f)$. In this section, as mentioned before, we apply the results of [20] to the case of the prototype matrix $H_3$. 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 $J_d$c to $J_d c - J - L + 1$.

It is worth noticing that matrices of the form $H = (b, d, e, f)$ do not give a unique representative of each equivalence class, i.e., two matrices $H = (b, d, e, f)$ and $H' = (b', d', e', f')$ 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 $P_{J,L}$ of pair of permutations ($\pi_r, \pi_c$) of size equal to $L$ and $J$, respectively, that satisfies

$$
H = P_{\pi_r} \times H \times P_{\pi_c},
$$

(11)

where $P_{\pi_r}$ and $P_{\pi_c}$ are, respectively, the permutation matrices associated with $\pi_r$ and $\pi_c$.

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

$$
P_{\pi_r} \times H_3 \times P_{\pi_c} = \begin{bmatrix}
0 & 1 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
\end{bmatrix}
\times
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & . & . & . \\
1 & 1 & 1 & 1 \\
\end{bmatrix}
\times
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & . & . \\
. & 1 & 1 & 1 \\
. & . & 1 & 1 \\
\end{bmatrix}
$$

then $(P_{\pi_r} \times H_3) \times P_{\pi_c}$ gives

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

(12)

(13)

Property 3.4 implies that the right product with $P_{\pi_c}$ is equivalent to the permutation of the columns according to $\pi_c^{-1}$. 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 $P_{\pi_c}$: 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 $P_{\pi_r} = P_{\pi_r} \otimes I_N$, $P_{\pi_c} = P_{\pi_c} \otimes I_N$ and $H \in \mathbb{H}_N^N$, then $H^{\pi_r,\pi_c} = P_{\pi_r} \times H \times P_{\pi_c}$ is also an element of $\mathbb{H}_N^N$ satisfying $H^{\pi_r,\pi_c} = H$. 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 $H = (b, d, e, f)$ be an element of $\mathbb{H}_3^3$ and $(\pi_r, \pi_c)$ the pair of permutations defined above, then $H^{\pi_r,\pi_c}$ can be represented as

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

(14)

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

For each of the 6 permutations $\pi_r$ on the 3 lines of the prototype matrix $H_3$, there are exactly 8 permutations $\pi_c$ on columns that give structural transformations. For example, if $\pi_r$ 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 $H_3$ (2). By combining these 3 permutations, a total of 8 structural permutations are obtained when $\pi_r$ 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 $b, d, f - e \neq 0$ 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 \((b, d, e, f)\) of \(\mathbb{H}_3\) characterized by \(\Phi(H) = (b, d, f - e)\) is equivalent to \(b\) satisfying \(b \leq N/2\) \(d \leq N/2\) and \(f - e \leq N/2\).

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

A. \(\Phi\)-order constraint: \(\phi(C_{AB}) \leq \phi(C_{CD}) \leq \phi(C_{EF})\).

Lemma 4.4 (Swapping \(C_{AB}\) and \(C_{CD}\)): Any matrix \(H = (b, d, e, f)\) of \(\mathbb{H}_3\) characterized by \(\Phi(H) = (b, d, f - e)\) is equivalent to a matrix \(H_1 \in \mathbb{H}_3\) characterized by \(\Phi(H_1) = (d, b, f - e)\).

Proof: According to line 26 of Table VI, \(H = (b, d, e, f)\) is equivalent to the matrix \(H_1 = (d, b, f - e)\) owing to the equivalence relation \(H_1 = P_{\pi_e} \times H \times P_{\pi_c}\) with \((\pi_e, \pi_c) = (3, 1, 2, 5, 6)\). Matrix \(H_1\) is characterized by \(\Phi(H_1) = (d, b, f - e)\), which completes the proof.

Lemma 4.5 (\(\Phi\)-Order): Any matrix \(H\) of \(\mathbb{H}_3\) is equivalent to a matrix \(H_0 = (b, d, e, f)\) with \(\Phi(H_0) = (b, d, f - e)\) satisfying \(b \leq N/2\) \(d \leq N/2\) and \(f - e \leq N/2\).

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

Theorem 4.3 (\(\Phi\)-Order): Any matrix \(H\) of \(\mathbb{H}_3\) is equivalent to a matrix \(H_0 = (b, d, e, f)\) with \(\Phi(H_0) = (b, d, f - e)\) satisfying \(b \leq N/2\) \(d \leq N/2\) and \(f - e \leq N/2\).

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

Theorem 4.8 (Constraint $\Phi$-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 $\phi$ values of $H_0$ in increasing order.

Constraint $\phi(C_{AB}) \leq \phi(C_{CD}) \leq \phi(C_{EF}) \leq N/2$ 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 $\Phi$-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 $\Phi$ 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. $\Phi^0$-order-$N/2$ constraint: $0 < \phi(C_{AB}) \leq \phi(C_{CD}) \leq \phi(C_{EF}) \leq N/2$.

Finally, we can also add an additional constraint to the $\Phi$-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 $\Phi^0$-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 $\Phi^0$-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 $P_k^X$ is the permutation matrix associated with the permutation $\pi_k^X(i) = k \times i \mod N$.

Since $k$ and $N$ are coprime, then there exists an integer $k^{-1}$ so that $k \times k^{-1} = 1 \mod N$ and thus, $(\pi_k^X)^{-1} = \pi_{k^{-1}}^X$ and $(P_k^X)^{-1} = P_{k^{-1}}^X$.

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 $P_k^X \times I_a^N \times P_{k^{-1}}^X$ is equal to the $(ka \mod N)$-circulant matrix $I_{ak}^N$.

Proof: $P_k^X \times I_a^N \times P_{k^{-1}}^X$ is the permutation matrix associated with the permutation $\pi = \pi_k^X(\pi_a^X(\pi_{k^{-1}}^X))$. For all $i \in \mathbb{Z}/N\mathbb{Z}$, we have $\pi_{k^{-1}}^X(i) = k^{-1}i$, thus $\pi_a^X(\pi_{k^{-1}}^X(i)) = k^{-1}i + a$ and therefore, $\pi(i) = \pi_k^X(k^{-1}i + a) = (i + ka) \mod N$. Thus, $\tau$ 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, $P_7^X \times I_{23}^2 \times P_{23}^X = I_{168}^2$.

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): $H_k^X = (I_0^I \otimes P_k^X) \times H \times (I_0^I \otimes P_{k^{-1}}^X)$ is the expanded matrix defined as $H_k^X(i, j) = k \times H(i, j)$.

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

Combined with additive and structural transformations, multiplicative transformation allows us to identify more matrices in the same equivalence 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 $(kb, kd, ke, kf)_N$.

\[ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & d & e \\ 0 & 0 & 0 & f \end{pmatrix} \equiv \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & kb & 0 & 0 \\ 0 & 0 & ke & kf \end{pmatrix} \] (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 $\{D(N)/N\}$ 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 $\rho$, let us define $k_\rho$ as $k_\rho = u + \rho \times N/\gcd(b, N)$. Thus, by construction, $k_\rho \times b = \gcd(b, N) \mod N$. According to Dirichlet’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 $k_\rho$ 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, $k_3 = 13$ 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 $\Phi(H_n) = \langle b_n, d_n, e_n, f_n \rangle$, where $\Phi(H_n) = \langle b_n, d_n, e_n, f_n \rangle$. Thus, using 5.7, the search space is further reduced by up to 48.
(b_n, d_n, f_n - e_n) 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_n, 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 

\( H_0 = \langle b_0, d_0, e_0, f_0 \rangle, \ H_1 = \langle b_1, d_1, e_1, f_1 \rangle, \ldots , \ H_N = \langle b_N, d_N, e_N, f_N \rangle \)

satisfying the \( \Phi^0 \)-order-N/2 constraint, with decreasing value of \( b \), i.e. \( b_0 > b_1 > b_2 \cdots \cdots > b_N \geq 1 \) until \( b_N \) reaches the value of a divisor of \( N \) (eventually, \( b_N = 1 \)).

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

Reconstruction: Let us assume that \( H \) is equivalent to a matrix

\( H_n = \langle b_n, d_n, e_n, f_n \rangle \)

satisfying the \( \Phi^0 \)-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 \( H_k \) is equivalent to \( \langle \gcd(b_k, N), kd_n, ke_n, k f_n \rangle \). Note that \( \gcd(b_n, N) \) is less than \( b_n \) since \( b_n \) is not a divisor of \( N \).

Due to the multiplication by \( k \) and the reduction modulo \( N \), \( \langle \gcd(b_n, N), kd_n, ke_n, k f_n \rangle \) may no longer respect the \( \Phi^0 \)-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 \( \Phi^0 \)-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 \( (b, d, e, f) \) respecting the \( \Phi^0 \)-order-N/2 constraint and \( b \in \mathbb{N} \). The value \( \rho(N) \) can be computed as

\[
\rho(N) = \sum_{b \in \mathbb{D}(N)/N} \sum_{d = b}^{N/2} \sum_{d \leq e \leq N/2} 1. \tag{16}
\]

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

\[
\rho(N) = N \sum_{b \in \mathbb{D}(N)/N} \frac{|N/2| - b + 1}{2}. \tag{17}
\]

Note that if \( N \) is a prime number greater than 2, then \( \mathbb{D}(N)/N = \{1\} \) 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 \( (1, 7, 13, 17)_{35} \) satisfying the \( \Phi^0 \)-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 \( \mathbb{P}_H \) of pair of permutations associated with the prototype matrix \( H \), 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 by 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 \( \mathbb{H}_3 \) prototype matrix with a girth ranging from \( g = 8 \) up to \( g = 14 \). Then, we give an explicit rule based on the multiplicity \( M(g) \) 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 \( \mathbb{H}_3 \) protograph.

The problem of optimal lifting by a factor \( N \) of the \( \mathbb{H}_3 \) 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 \( \mathbb{H}_3^{N, \bar{g}} \) 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

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

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 \( M(\tilde{g}_N(H)) \) minimum value, then the value of \( M(\tilde{g}_N(H) + 2) \) 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 \( H_3 \) using the method presented in [29]. Note that the software to perform this enumeration is available online [30]. The number \( N(\ell) \) of cycles for each value of \( \ell \) is given in Table I. Each cycle gives a new constraint on the lifted matrix according to equation (6).

<table>
<thead>
<tr>
<th>( \ell )</th>
<th>( N(\ell) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>12</td>
<td>139</td>
</tr>
<tr>
<td>14</td>
<td>336</td>
</tr>
</tbody>
</table>

**TABLE I**

Number of constraints to be taken into account by the constraint programming software

Then, using the \( \Phi^0 \)-order-\( N/2 \) constraint, we obtained a well-defined constraint optimization problem to find the matrix \( b, d, e, f \) of \( C \) giving a girth greater 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 \( \ell_0 < g_0 \) is given in the last line of Table I. It is noteworthy that some constraints are redundant. For example, the cycle \( C_{AEFEC} \) of length \( \ell = 14 \) gives the same constraint as the cycle \( C_{AECBF} \) of length \( \ell' = 10 \) since \( \Phi(C_{AEFEC}) = \Phi(C_{AECBF}) = f - 2e \). 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 \( H = (b, d, e, f) \) with no particular constraint on \( b, d, e \) and \( f \)). Line 3 gives the number of solution \( n_1 \) when the \( \Phi \)-order -\( N/2 \) is used, while line 4 shows the number of solutions \( n_2 \) when both \( \Phi \)-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 \( \Phi^0 \)-order-\( N/2 \) constraint can thus be expressed with strict inequality as \( 0 < \Phi(C_{AB}) < \Phi(C_{CD}) < \Phi(C_{EF}) < N/2 \). 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 \( \Phi^0 \)-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 \( M(g) \) 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 \( M(14) \) and \( M(16) \).

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 \( n_m = 20 \) 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.
### TABLE III

<table>
<thead>
<tr>
<th>g₀ = 8</th>
<th>n₀</th>
<th>n₁</th>
<th>n₂</th>
<th>R</th>
<th>N/A</th>
<th>N/A</th>
<th>10</th>
<th>16</th>
<th>28</th>
<th>52</th>
<th>100</th>
<th>114</th>
<th>192</th>
<th>600</th>
<th>1680</th>
<th>5184</th>
<th>15840</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>14</td>
<td>18</td>
<td>55</td>
<td>67</td>
<td>150</td>
<td>175</td>
<td>335</td>
<td>378</td>
<td>10800</td>
<td>1730</td>
<td>7130</td>
<td>4480</td>
<td>2648</td>
<td>702</td>
<td>288</td>
</tr>
<tr>
<td>g₀ = 10</td>
<td>n₀</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>17</td>
<td>14</td>
<td>20</td>
<td>20</td>
<td>35</td>
<td>99</td>
<td>39</td>
<td>92</td>
<td>150</td>
<td>175</td>
<td>192</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>14</td>
<td>18</td>
<td>55</td>
<td>67</td>
<td>150</td>
<td>175</td>
<td>335</td>
<td>378</td>
<td>10800</td>
<td>1730</td>
<td>7130</td>
<td>4480</td>
<td>2648</td>
<td>702</td>
<td>288</td>
</tr>
<tr>
<td>g₀ = 12</td>
<td>n₀</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>17</td>
<td>14</td>
<td>20</td>
<td>20</td>
<td>35</td>
<td>99</td>
<td>39</td>
<td>92</td>
<td>150</td>
<td>175</td>
<td>192</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>14</td>
<td>18</td>
<td>55</td>
<td>67</td>
<td>150</td>
<td>175</td>
<td>335</td>
<td>378</td>
<td>10800</td>
<td>1730</td>
<td>7130</td>
<td>4480</td>
<td>2648</td>
<td>702</td>
<td>288</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>#</th>
<th>(b, d, e, f)</th>
<th>M(14)</th>
<th>M(16)</th>
<th>Lₘ</th>
<th>Lₙ</th>
<th>Lₚ</th>
<th>Lₚ</th>
<th>Lₚ</th>
<th>Lₚ</th>
<th>Lₚ</th>
<th>Lₚ</th>
<th>Lₚ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1, 3, 7, 19)</td>
<td>180</td>
<td>825</td>
<td>38</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
</tr>
<tr>
<td>2</td>
<td>(1, 5, 7, 16)</td>
<td>180</td>
<td>840</td>
<td>38</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
<tr>
<td>3</td>
<td>(1, 6, 25, 8)</td>
<td>180</td>
<td>810</td>
<td>38</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>(1, 9, 2, 13)</td>
<td>90</td>
<td>1080</td>
<td>38</td>
<td>108</td>
<td>108</td>
<td>108</td>
<td>108</td>
<td>108</td>
<td>108</td>
<td>108</td>
<td>108</td>
</tr>
<tr>
<td>5</td>
<td>(3, 5, 13)</td>
<td>240</td>
<td>750</td>
<td>38</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
</tr>
</tbody>
</table>

**TABLE II**

Comparison of execution time and number of solutions with and without Φ-order-N/2 constraint for g₀ = 8, 10, 12, 14 and 16.

---

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 × 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 7 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
   
   \[
   \mathcal{H}^{40}_{(3,11)} = \begin{bmatrix}
   0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
   0 & 1 & 3 & 5 & 8 & 12 & 16 & 19 & 21 & 23 & 24 \\
   0 & 2 & 7 & 18 & 33 & 35 & 39 & 40 & 41 & 42 \\
   \end{bmatrix}
   \]

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 =

---

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

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

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

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

$$H_{(3,10)}^{583} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 5 & 12 & 32 & 55 & 160 & 237 & 261 & 305 \\ 0 & 3 & 13 & 29 & 74 & 246 & 111 & 170 & 350 & 132 \end{bmatrix}$$  \tag{24}

and

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

TABLE IV

<table>
<thead>
<tr>
<th>Year</th>
<th>Lower bound</th>
<th>$L = 4$</th>
<th>$L = 5$</th>
<th>$L = 6$</th>
<th>$L = 7$</th>
<th>$L = 8$</th>
<th>$L = 9$</th>
<th>$L = 10$</th>
<th>$L = 11$</th>
<th>$L = 12$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>[20]</td>
<td>9</td>
<td>14</td>
<td>18</td>
<td>21</td>
<td>26</td>
<td>33</td>
<td>39</td>
<td>46</td>
<td>54</td>
</tr>
<tr>
<td>2006</td>
<td>[19]</td>
<td>9</td>
<td>13</td>
<td>18</td>
<td>22</td>
<td>27</td>
<td>34</td>
<td>40</td>
<td>49</td>
<td>55</td>
</tr>
<tr>
<td>2017</td>
<td>[19]</td>
<td>9</td>
<td>13</td>
<td>18</td>
<td>21</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>41</td>
<td>45</td>
</tr>
</tbody>
</table>

TABLE V

Randomly found solution for a complete protomatrix, with $J = 3$ lines and $L$ columns

ACKNOWLEDGMENT

The authors would like to thank several people that have also help for the realization of this paper: Ahmed Abdouleh 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 Tsidighi 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 resources funded by the Région Bretagne and the ANR through the CPER Sophie.

APPENDIX

REFERENCES

https://www.dvb.org/standards/dvb-s2

http://ieeexplore.ieee.org/browse/standards/get-programpage/series?id=68


TABLE VI

<table>
<thead>
<tr>
<th>#</th>
<th>( \pi_r )</th>
<th>( \pi_c )</th>
<th>((b, d, e, f)_N = \equiv (1, 1, 13, 17)_N )</th>
<th>( b ) divisor of ( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((+b, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>2</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((+b, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>3</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>4</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>5</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((b + d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>6</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((b + d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>7</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((-b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>8</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((-b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>9</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((-b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>10</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((-b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>11</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((-b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
<tr>
<td>12</td>
<td>[1, 2, 4]</td>
<td>[1, 2, 3, 4, 5, 6]</td>
<td>((-b - d, +d, +e, +f)_N )</td>
<td>((1, 7, 13, 17)_N )</td>
</tr>
</tbody>
</table>

TABLE VII

<table>
<thead>
<tr>
<th>#</th>
<th>( \pi_r )</th>
<th>( \pi_c )</th>
<th>((b, d, e, f)_N = \equiv (1, 1, 13, 17)_N )</th>
<th>( b ) divisor of ( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>26</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>27</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>28</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>29</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>30</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>31</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>32</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>33</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>34</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
<tr>
<td>35</td>
<td>[1, 3, 2]</td>
<td>[3, 4, 1, 2, 5, 6]</td>
<td>((+d, +b, +e, +f)_N )</td>
<td>((1, 7, 22, 18)_N )</td>
</tr>
</tbody>
</table>


