Designing Protograph-Based Quasi-Cyclic Spatially Coupled LDPC Codes with Large Girth

Shiyuan Mo, *Student Member, IEEE*, Li Chen, *Senior Member, IEEE*, Daniel J. Costello, Jr., *Life Fellow, IEEE*, David G. M. Mitchell, *Senior Member, IEEE*, Roxana Smarandache, *Senior Member, IEEE*, Jie Qiu

Abstract

Spatially coupled (SC) low-density parity-check (LDPC) codes can achieve capacity approaching performance with low message recovery latency when using sliding window (SW) decoding. An SC-LDPC code constructed from a protograph can be generated by first *coupling* a chain of block protographs and then *lifting* the coupled protograph using permutation matrices. In this paper, we introduce a systematic design to eliminate 4-cycles in a coupled protograph. Further using a quasi-cyclic (QC) lifting, we introduce a procedure for constructing QC-SC-LDPC codes of girth at least eight. This can be interpreted as a multi-stage graph lifting process that yields a greater flexibility in designing QC-SC-LDPC codes with a large girth than previous approaches. Simulation results show the design leads to improved decoding performance, particularly in the error floor, compared to random constructions. Finally, we determine the minimum coupling width required to eliminate 4-cycles in a coupled protograph.

Index Terms

Shiyuan Mo, Li Chen and Jie Qiu are with the School of Electronics and Information Technology at the Sun Yat-sen University, Guangzhou, P. R. China. Daniel J. Costello, Jr. and Roxana Smarandache are with the Department of Electrical Engineering, and Roxana Smarandache is also with the Department of Mathematics, at the University of Notre Dame, Notre Dame, IN, USA. David G. M. Mitchell is with the Klipsch School of Electrical and Computer Engineering at the New Mexico State University, Las Cruces, NM, USA. Email: moshiy@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn, costello.2@nd.edu, dgmm@nmsu.edu, rsmarand@nd.edu.

This work is sponsored by the National Natural Science Foundation of China (NSFC) under project ID 61671486, and in part by the National Science Foundation under grant No. ECCS-1710920. Some parts of the paper were published in the Proceedings of the 2017 IEEE International Symposium on Information Theory.

Girth, protographs, quasi-cyclic LDPC codes, spatially coupled codes, sliding window decoding.

I. Introduction

Since the original work of Thorpe [1], it has been recognized that protographs provide an efficient method of constructing low-density parity-check (LDPC) codes. Analyzing the iterative decoding thresholds and minimum distance growth properties of small protographs facilitates the construction of code ensembles with good asymptotic properties after applying a *graph-lifting* procedure [2]. Protograph-based methods were used to construct good spatially coupled LDPC (SC-LDPC) codes in [3], where an *edge-spreading* procedure is first used to couple together a chain of block code protographs (thus introducing memory to the code), followed by graph lifting using permutation matrices. This two-step code design procedure was shown to result in SC-LDPC code ensembles with thresholds approaching the *maximum a posteriori* (MAP) thresholds of their underlying LDPC block code ensembles, i.e., they exhibit the *threshold saturation* effect [4] [5] [6] [3], and linear growth of minimum distance with block length, i.e., the ensembles are *asymptotically good*. If the permutation matrices used in the lifting procedure are circulants (shifted identity matrices), a quasi-cyclic (QC) ensemble results, a desirable property for practical implementation [7]. One important aspect of finite-length LDPC code design is to maximize the girth of the Tanner graph representation of the parity-check matrix to ensure that the convergence behavior of iterative decoding is not negatively affected by short cycles. For protograph-based constructions of QC-LDPC codes, this can be accomplished by applying the Fossorier condition [8] to the graph lifting.

Several constructions of QC-SC-LDPC codes have been proposed recently in the literature [7] [9]–[15]. Most of these approaches, including [11]–[15], focus on constructions of QC-SC-LDPC codes that are based on a certain underlying block code structure. The goal of these papers is to devise good edge spreading (or coupling) connections given the underlying code. These approaches typically involve minimizing the harmful objects (cycles, absorbing sets) based on the structure of the underlying code and were shown in those papers to result in QC-SC-LDPC codes with improved code performance. However, the computational complexity of searching for a good edge spreading limits the memory (or coupling width) of the resulting QC-SC-LDPC codes to be small. For example, in [11] and [15], optimization techniques were used to minimize the number of 6-cycles in circulant-based SC-LDPC codes. Due to the complexity of the optimization, however, this approach is limited to coupling widths less than or equal to two. Direct designs of time-invariant QC-SC-LDPC codes, based on the polynomial representation of the parity-check matrix of the code, were also investigated in [16]–[20]. The design of more general SC-LDPC codes was also considered in [4] [21] [22], where protographs were constructed with the smallest constraint length needed to avoid 4-cycles. Finally, memory efficient hardware implementations of QC-SC-LDPC codes have been addressed in [23].

A primary motivation of this paper is the heuristic construction of SC-LDPC code designs with large coupling widths, since such designs have been shown to be capable of better performance on a fixed latency basis [24].[1] Motivated by the results of [25] [26], we also take a more general multi-stage lifting approach that can be used to improve the design of the code at each lifting stage, where reduction/elimination of problematic objects can be achieved at the different stages, including at the first graph-lifting, i.e., the protograph design stage. Our idea also depends on the fact that the girth of a lifted graph is lower bounded by the girth of its base graph [12]. Hence, starting from a block code protograph with good asymptotic threshold and distance properties, we design the edge spreading in two stages to maximize the girth and minimize the number of short cycles in the SC protograph. The edge-spreading procedure can be interpreted as decomposing a base matrix $\mathbf{B}$ (corresponding to a block code protograph) into a number of component matrices, which are then used to form an SC base matrix $\mathbf{B}_{\text{SC}}$. In our approach, we identify several sub-blocks of $\mathbf{B}_{\text{SC}}$ that guide the design of the component matrices, leading to an SC protograph with a girth of at least six.

By further performing a circulant-based graph-lifting of $\mathbf{B}_{\text{SC}}$ and applying the Fossorier condition to generate an SC parity-check matrix $\mathbf{H}_{\text{SC}}$, we obtain QC-SC-LDPC codes with a girth of at least eight.[2] Simulation results show that substantial performance gains, particularly in the error floor, are achieved using the two-stage design approach compared to random code constructions. Note that, with an undesigned SC protograph, the Fossorier condition can still be applied to yield an SC-LDPC code of girth eight. However, the existence of many short cycles in the protograph makes the process very complex. The two-stage approach, on the other hand,

[1] The decoding latency depends on the product of the coupling width and the graph lifting factor, so codes with a large coupling width can still have small latency.

[2] In this paper, we have restricted our approach to eliminating cycles of length four and length six in the two design stages; however, we believe that the approach could be suitably generalized to other design criteria.

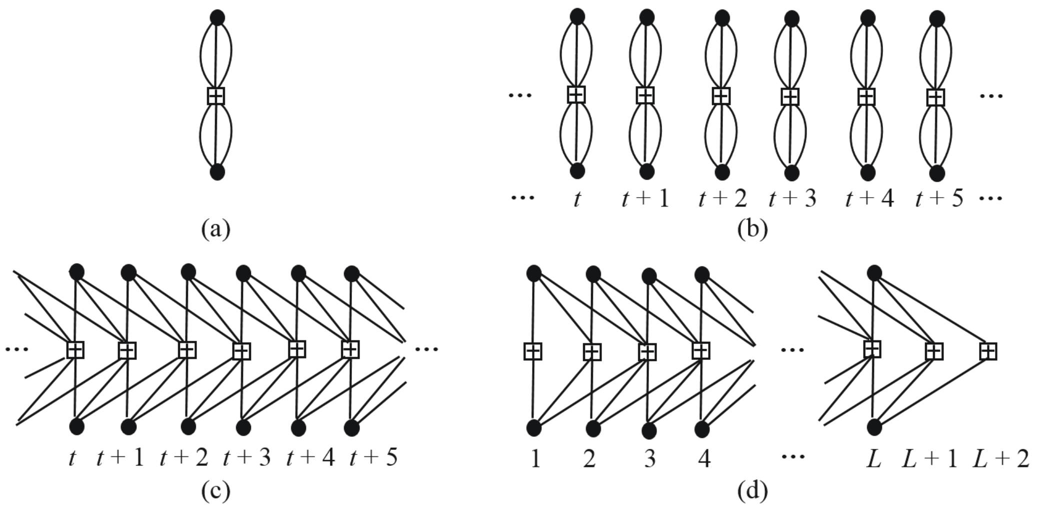
makes it much easier to apply the Fossorier condition, since the SC protograph has already been designed to have girth six. In addition, in contrast to optimization methods [11] [15], our heuristic approach allows us the possibility of using larger coupling widths, which make it easier to guarantee girth six. Although the proposed multi-stage design framework does not carry any guarantee of optimality, it does allow us the flexibility in code design needed to reduce/eliminate harmful objects in the Tanner graph, e.g., cycles, absorbing sets, and so on. Moreover, compared to the time-invariant QC-SC-LDPC code designs in [16]–[20], our approach can produce periodically time-varying QC-SC-LDPC codes (of which time-invariant QC-SC-LDPC codes are a special case), which have the potential of yielding larger minimum (free) distances [3]. Further, if non-circulant or random liftings are desired, our approach still guarantees a code with girth at least six and lends itself to asymptotic analyses of the SC-LDPC code ensembles derived from the designed protographs, for which we then compute belief propagation (BP) iterative decoding thresholds over the binary erasure channel (BEC). Finally, we perform an heuristic search for the minimum coupling width required to eliminate 4-cycles in a coupled protograph.

II. SC-LDPC Codes

The construction of a protograph-based SC-LDPC code can be described as a two-step procedure – first protograph coupling and then graph lifting [3]. A block protograph [1] is a small bipartite graph with n_c check nodes and n_v variable nodes, where $n_c < n_v$. It can be represented by a base matrix

$$\mathbf{B} = [\mathbf{B}(r,s)]_{n_c \times n_v}, \tag{1}$$

where $\mathbf{B}(r,s)$, a non-negative integer, is the row-r column-s entry, $r = 1, \dots, n_c$, $s = 1, \dots, n_v$. The entries determine the number of edges that connect check node r to variable node s in the protograph. For example, Fig. 1(a) shows a block protograph defined by $\mathbf{B} = [3\ 3]$. To construct an SC protograph, we first replicate the block protograph as an infinite chain, as shown in Fig. 1(b), and then spread edges from the variable nodes of the protograph at time instant t by connecting them to check nodes at time instants t to $t + \omega$. Repeating this spreading over all the protographs in the chain yields an SC protograph with *coupling width* ω, as shown in Fig. 1(c) for the case $\omega = 2$. This edge spreading can be interpreted as decomposing $\mathbf{B}$ into $\omega + 1$ component matrices of the same size, i.e., $\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_\omega$, such that



(a)

(b)

(c)

(d)

Fig. 1. (a) Block protograph for $\mathbf{B} = [3\ 3]$, (b) an infinite chain of block protographs, (c) coupling the protographs with $\omega = 2$, and (d) a finite chain of L coupled protographs.

$$\mathbf{B}(r,s) = \sum_{i=0}^{\omega} \mathbf{B}_i(r,s), \tag{2}$$

so the coupled protograph maintains the same check node and variable node degrees as the original block protograph. If the block protograph has a regular structure that exhibits uniform check node and variable node degrees, as in Fig. 1(a), the constructed SC protograph will also be regular. In practice, an SC protograph is terminated after a finite number L of coupled block protographs, where L is called the *coupling length*. The terminated protograph contains Ln_v variable nodes and $(L+\omega)n_c$ check nodes, as shown in Fig. 1(d). The corresponding $(L+w)n_c \times Ln_v$ SC base matrix

$$\mathbf{B}_{\text{SC}}^{(L)} = \begin{bmatrix} \mathbf{B}_0 & & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & & \\ \vdots & \vdots & \ddots & & & \\ \mathbf{B}_\omega & \mathbf{B}_{\omega-1} & \cdots & \mathbf{B}_0 & & \\ & \ddots & & & \ddots & \\ & & \mathbf{B}_\omega & \mathbf{B}_{\omega-1} & \cdots & \mathbf{B}_0 \\ & & & \ddots & & \vdots \\ & & & & \mathbf{B}_\omega & \mathbf{B}_{\omega-1} \\ & & & & & \mathbf{B}_\omega \end{bmatrix} \tag{3}$$

exhibits a diagonal band of nonzero entries. Note that the first and last ωn_c check nodes have reduced degrees, i.e., the terminated protograph has a slight irregularity at both ends, which is an important feature in realizing the saturation threshold effect of SC-LDPC codes [3], [4], [5], [6].

The parity-check matrix $\mathbf{H}_{\text{SC}}^{(L)}$ of an SC-LDPC code can be obtained by an M-fold matrix expansion from $\mathbf{B}_{\text{SC}}^{(L)}$ that corresponds to an M-fold graph-lifting of the terminated SC protograph [27]. In the lifted graph, each check node and variable node is replaced by M copies of the original node and each edge is replaced by M edges connecting M pairs of check and variable nodes. From $\mathbf{B}_{\text{SC}}^{(L)} = [\mathbf{B}_{\text{SC}}^{(L)}(r,s)]_{(L+\omega)n_c \times Ln_v}$, $\mathbf{H}_{\text{SC}}^{(L)}$ is generated by replacing each nonzero entry in $\mathbf{B}_{\text{SC}}^{(L)}$ by a non-overlapping sum of $\mathbf{B}_{\text{SC}}^{(L)}(r,s)$ $M \times M$ binary permutation matrices $\mathbf{P}_M$ and replacing each zero entry by the $M \times M$ all zero matrix, so that $\mathbf{H}_{\text{SC}}^{(L)}$ also exhibits a diagonal band of nonzero entries. The *constraint length* and *design rate* of the corresponding code are $v = Mn_v(\omega+1)$ and $R_{\text{SC}}^{(L)} = 1 - \frac{(L+\omega)n_c}{Ln_v}$, respectively, and the *asymptotic rate* is given by $\lim_{L\to\infty} R_{\text{SC}}^{(L)} \triangleq R_{\text{SC}}^{(\infty)} = 1 - \frac{n_c}{n_v}$. $\mathbf{H}_{\text{SC}}^{(L)}$ defines a particular SC-LDPC code, whose girth (denoted g) is given by the length of the shortest cycle in the corresponding Tanner graph. We restate here a Lemma from [12]. (Other similar girth characterizations of LDPC codes have been reported in [34] [35] [36].)

Lemma 1. The girth of the Tanner graph of $\mathbf{H}_{\text{SC}}^{(L)}$ is lower bounded by the girth of the protograph of $\mathbf{B}_{\text{SC}}^{(L)}$.

Proof: Let g and g' denote the girths of the Tanner graphs of $\mathbf{B}_{\text{SC}}^{(L)}$ and $\mathbf{H}_{\text{SC}}^{(L)}$, respectively. Assume $g' < g$, i.e., $\mathbf{H}_{\text{SC}}^{(L)}$ contains a cycle of length less than g. Since $\mathbf{H}_{\text{SC}}^{(L)}$ is comprised of permutation matrices, which contain only a single one in any row or column, this implies that there also exists a cycle of length less than g in $\mathbf{B}_{\text{SC}}^{(L)}$, which contradicts the fact that the girth of $\mathbf{B}_{\text{SC}}^{(L)}$ is g.[3] ■

This lemma motivates the design in Section III.

III. Design of $\mathbf{B}_{\text{SC}}^{(L)}$

Based on Lemma 1, the proposed approach aims to first eliminate (or reduce the number of) 4-cycles in $\mathbf{B}_{\text{SC}}^{(L)}$. Then, using a systematic circulant-based lifting, we try to construct matrices

[3] If $\mathbf{B}_{\text{SC}}^{(L)}$ contains integer values greater than one, corresponding to a multi-edge protograph, its Tanner graph has girth $g = 2$, and the lemma follows immediately.

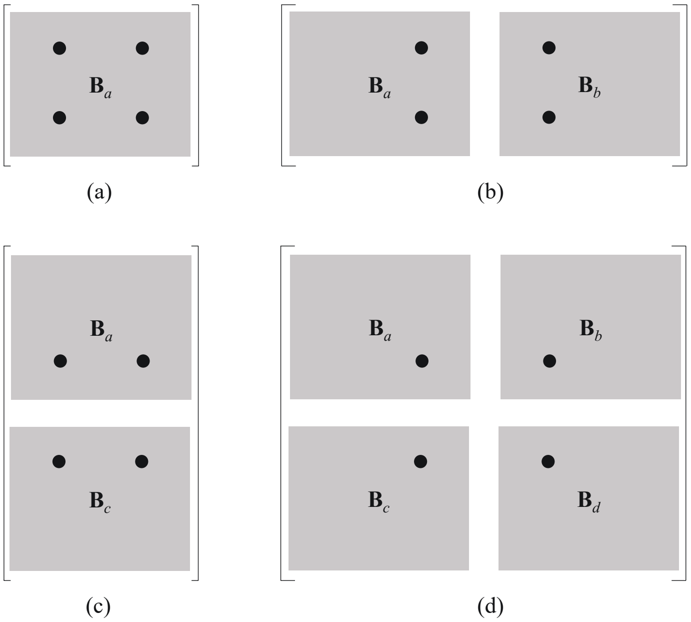


Fig. 2. Nonzero entries (solid circles) that form 4-cycles in $\mathbf{B}_{\text{SC}}^{(L)}$, where $\mathbf{B}_a$, $\mathbf{B}_b$, $\mathbf{B}_c$ and $\mathbf{B}_d$, $a, b, c, d \in \{0, 1, \ldots, \omega\}$, are $n_c \times n_v$ component matrices in $\mathbf{B}_{\text{SC}}^{(L)}$.

$\mathbf{H}_{\text{SC}}^{(L)}$ with girth $g \geq 8$. Due to the diagonal nature of $\mathbf{B}_{\text{SC}}^{(L)}$ (see (3)), a careful examination of its structure is needed in the design.

A. Preliminaries

To illustrate the procedure, we consider the common case when the base matrix is all-ones, i.e., $\mathbf{B} = \mathbf{1}_{n_c \times n_v}$, resulting in an (n_c, n_v)-regular protograph. A 4-cycle in a coupled protograph corresponds to four nonzero entries that form a rectangular array in $\mathbf{B}_{\text{SC}}^{(L)}$. Fig. 2 shows demonstrative sketches of all possible patterns of 4-cycles in $\mathbf{B}_{\text{SC}}^{(L)}$, which leads to the following lemma.

Lemma 2. In $\mathbf{B}_{\text{SC}}^{(L)}$, 4-cycles may be contained in: 1) one component matrix of $\mathbf{B}_{\text{SC}}^{(L)}$ (see Fig. 2(a)); 2) two component matrices that appear in the same row or the same column of $\mathbf{B}_{\text{SC}}^{(L)}$ (see Figs. 2(b) and 2(c)); 3) four component matrices that appear in a rectangular array of $\mathbf{B}_{\text{SC}}^{(L)}$ (see Fig. 2(d)).

We now decompose $\mathbf{B}_{\text{SC}}^{(L)}$ as follows:

- The *representative block* $\mathbf{B}_{\text{R}}$ is defined as

$$\mathbf{B}_{\mathrm{R}} \triangleq \begin{bmatrix} \mathbf{B}_{\omega} & \mathbf{B}_{\omega-1} & \cdots & \mathbf{B}_0 \\ & \mathbf{B}_{\omega} & \cdots & \mathbf{B}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{B}_{\omega} \end{bmatrix}, \tag{4}$$

with size $(\omega+1)n_c \times (\omega+1)n_v$. By comparing (4) to (3), and noting the repeating diagonal structure of $\mathbf{B}_{\text{SC}}^{(L)}$, it can be seen that any combination of one, two, or four component matrices that contain a 4-cycle in $\mathbf{B}_{\text{SC}}^{(L)}$ will appear in $\mathbf{B}_{\mathrm{R}}$. Therefore, if $\mathbf{B}_{\mathrm{R}}$ does not contain 4-cycles, neither will $\mathbf{B}_{\text{SC}}^{(L)}$. To help explain our design, the following two definitions based on $\mathbf{B}_{\mathrm{R}}$ are given:

- A *constituent block* $\mathbf{B}_{\mathrm{C}}$ is defined as

$$\mathbf{B}_{\mathrm{C}} \triangleq \begin{bmatrix} \mathbf{B}_{\beta-1} & \mathbf{B}_{\beta-2} & \cdots & \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_{\beta} & \mathbf{B}_{\beta-1} & \cdots & \mathbf{B}_2 & \mathbf{B}_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{B}_{\omega-1} & \mathbf{B}_{\omega-2} & \cdots & \mathbf{B}_{\alpha-1} & \mathbf{B}_{\alpha-2} \\ \mathbf{B}_{\omega} & \mathbf{B}_{\omega-1} & \cdots & \mathbf{B}_{\alpha} & \mathbf{B}_{\alpha-1} \end{bmatrix}, \tag{5}$$

where $\omega = \alpha+\beta-2$ and $\alpha, \beta > 1$, with size $\alpha n_c \times \beta n_v$. $\mathbf{B}_{\mathrm{C}}$ is obtained by forming a rectangular matrix from $\mathbf{B}_{\mathrm{R}}$ that contains $\mathbf{B}_0$ in the upper right corner and one of the $\mathbf{B}_{\omega}$ component matrices all along the diagonal of $\mathbf{B}_{\mathrm{R}}$ (except those in the upper left and lower right corners) in the lower left corner. Hence, there are $\omega - 1$ choices for $\mathbf{B}_{\mathrm{C}}$. Note that when $\omega = 2$, $\mathbf{B}_{\mathrm{C}}$ is unique, and when $\omega = 1$, $\mathbf{B}_{\mathrm{C}}$ does not exist. Note that each constituent block includes all component matrices $\mathbf{B}_i$. For a given $\mathbf{B}_{\mathrm{C}}$, we define the weight $\mathit{wt}(\mathbf{B}_i)$ of a component matrix $\mathbf{B}_i$ as the number of times it is included in $\mathbf{B}_{\mathrm{C}}$, where $\sum_{i=0}^{\omega} \mathit{wt}(\mathbf{B}_i) = \alpha\beta$;

- *Excluded patterns* $\mathbf{B}_{\mathrm{E}}^{(j)}$ are defined as

$$\mathbf{B}_{\mathrm{E}}^{(1)} = \begin{bmatrix} \mathbf{B}_{\omega} & \mathbf{B}_0 \end{bmatrix}, \ \mathbf{B}_{\mathrm{E}}^{(2)} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_{\omega} \end{bmatrix},$$

$$\mathbf{B}_{\mathrm{E}}^{(j)} = \begin{bmatrix} \mathbf{B}_{a_j} & \mathbf{B}_{b_j} \\ \mathbf{B}_{c_j} & \mathbf{B}_{d_j} \end{bmatrix}, \ j = 3, 4, \ldots, n_{\mathrm{E}}, \tag{6}$$

where $a_j, b_j, c_j, d_j \in \{0, 1, \ldots, \omega\}$ and n_E is the number of excluded patterns. Block $\mathbf{B}_\mathrm{E}^{(1)}$ (resp. $\mathbf{B}_\mathrm{E}^{(2)}$) is the $n_c \times 2n_v$ (resp. $2n_c \times n_v$) single row (resp. column) pattern (pair of component matrices) that appears in $\mathbf{B}_\mathrm{R}$ but cannot appear in $\mathbf{B}_\mathrm{C}$, whereas $\mathbf{B}_\mathrm{E}^{(j)}$, $j = 3, 4, \ldots, n_\mathrm{E}$, are all $2n_c \times 2n_v$ rectangular patterns that appear in $\mathbf{B}_\mathrm{R}$ but not in $\mathbf{B}_\mathrm{C}$. The number of excluded patterns n_E depends on ω and the given $\mathbf{B}_\mathrm{C}$, while the particular set of excluded patterns depends on the given $\mathbf{B}_\mathrm{C}$. Note that when $\omega = 2$, there are only two excluded patterns $\mathbf{B}_\mathrm{E}^{(1)}$ and $\mathbf{B}_\mathrm{E}^{(2)}$, since the one $2n_c \times 2n_v$ rectangular pattern in $\mathbf{B}_\mathrm{R}$ also appears in $\mathbf{B}_\mathrm{C}$. Also, when $\omega = 1$, the only excluded patterns are $\mathbf{B}_\mathrm{E}^{(1)}$ and $\mathbf{B}_\mathrm{E}^{(2)}$.

The following example illustrates the above definitions.

Example 1. When $\omega = 4$, we have

$$\mathbf{B}_\mathrm{R} = \begin{bmatrix} \mathbf{B}_4 & \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 \\ & \mathbf{B}_4 & \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 \\ & & \mathbf{B}_4 & \mathbf{B}_3 & \mathbf{B}_2 \\ & & & \mathbf{B}_4 & \mathbf{B}_3 \\ & & & & \mathbf{B}_4 \end{bmatrix}.$$

$\mathbf{B}_\mathrm{C}$ can be chosen as the $3n_c \times 3n_v$ pattern

$$\mathbf{B}_\mathrm{C} = \begin{bmatrix} \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 \\ \mathbf{B}_4 & \mathbf{B}_3 & \mathbf{B}_2 \end{bmatrix},$$

where $wt(\mathbf{B}_0) = wt(\mathbf{B}_4) = 1$, $wt(\mathbf{B}_1) = wt(\mathbf{B}_3) = 2$, and $wt(\mathbf{B}_2) = 3$. The excluded patterns are

$$\mathbf{B}_\mathrm{E}^{(1)} = \begin{bmatrix} \mathbf{B}_4 & \mathbf{B}_0 \end{bmatrix},\ \mathbf{B}_\mathrm{E}^{(2)} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_4 \end{bmatrix},\ \mathbf{B}_\mathrm{E}^{(3)} = \begin{bmatrix} \mathbf{B}_3 & \mathbf{B}_0 \\ \mathbf{B}_4 & \mathbf{B}_1 \end{bmatrix},\mathbf{B}_\mathrm{E}^{(4)} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_4 & \mathbf{B}_3 \end{bmatrix},$$

where, following the notation of (6), we have $a_3 = 3$, $b_3 = 0$, $c_3 = 4$, $d_3 = 1$ and $a_4 = 1$, $b_4 = 0$, $c_4 = 4$, $d_4 = 3$.

Note that the constituent block can also be chosen as the $2n_c \times 4n_v$ pattern

$$\mathbf{B}_\mathrm{C} = \begin{bmatrix} \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_4 & \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 \end{bmatrix},$$

or the $4n_c \times 2n_v$ pattern

$$\mathbf{B}_{\text{C}} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_2 & \mathbf{B}_1 \\ \mathbf{B}_3 & \mathbf{B}_2 \\ \mathbf{B}_4 & \mathbf{B}_3 \end{bmatrix}.$$

In each of these cases, $\mathit{wt}(\mathbf{B}_0) = \mathit{wt}(\mathbf{B}_4) = 1$ and $\mathit{wt}(\mathbf{B}_1) = \mathit{wt}(\mathbf{B}_2) = \mathit{wt}(\mathbf{B}_3) = 2$. The $2n_c \times 4n_v$ choice of $\mathbf{B}_{\text{C}}$ results in the excluded patterns

$$\mathbf{B}_{\text{E}}^{(1)} = \begin{bmatrix} \mathbf{B}_4 & \mathbf{B}_0 \end{bmatrix},\ \mathbf{B}_{\text{E}}^{(2)} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_4 \end{bmatrix},\ \mathbf{B}_{\text{E}}^{(3)} = \begin{bmatrix} \mathbf{B}_2 & \mathbf{B}_0 \\ \mathbf{B}_4 & \mathbf{B}_2 \end{bmatrix}, \mathbf{B}_{\text{E}}^{(4)} = \begin{bmatrix} \mathbf{B}_2 & \mathbf{B}_1 \\ \mathbf{B}_4 & \mathbf{B}_3 \end{bmatrix},$$

$$\mathbf{B}_{\text{E}}^{(5)} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_3 & \mathbf{B}_2 \end{bmatrix}, \mathbf{B}_{\text{E}}^{(6)} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_4 & \mathbf{B}_3 \end{bmatrix},$$

while the $4n_c \times 2n_v$ choice of $\mathbf{B}_{\text{C}}$ results in the excluded patterns

$$\mathbf{B}_{\text{E}}^{(1)} = \begin{bmatrix} \mathbf{B}_4 & \mathbf{B}_0 \end{bmatrix},\ \mathbf{B}_{\text{E}}^{(2)} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_4 \end{bmatrix},\ \mathbf{B}_{\text{E}}^{(3)} = \begin{bmatrix} \mathbf{B}_3 & \mathbf{B}_0 \\ \mathbf{B}_4 & \mathbf{B}_1 \end{bmatrix}, \mathbf{B}_{\text{E}}^{(4)} = \begin{bmatrix} \mathbf{B}_3 & \mathbf{B}_1 \\ \mathbf{B}_4 & \mathbf{B}_2 \end{bmatrix},$$

$$\mathbf{B}_{\text{E}}^{(5)} = \begin{bmatrix} \mathbf{B}_2 & \mathbf{B}_0 \\ \mathbf{B}_3 & \mathbf{B}_1 \end{bmatrix}, \mathbf{B}_{\text{E}}^{(6)} = \begin{bmatrix} \mathbf{B}_2 & \mathbf{B}_0 \\ \mathbf{B}_4 & \mathbf{B}_2 \end{bmatrix}. \blacksquare$$

The above definitions lead to the following theorem.

Theorem 3. The coupled protograph of $\mathbf{B}_{\text{SC}}^{(L)}$ does not have any 4-cycles if the chosen $\mathbf{B}_{\text{C}}$ and its associated $\mathbf{B}_{\text{E}}^{(j)}, j = 1, 2, \ldots, n_{\text{E}}$, do not contain any 4-cycles.

Proof: The result follows directly from Lemma 2. For case 1), $\mathbf{B}_{\text{C}}$ includes all possible component matrices $\mathbf{B}_i, i = 0, 1, \ldots, \omega$. For cases 2) and 3), $\mathbf{B}_{\text{C}}$ and $\mathbf{B}_{\text{E}}^{(j)}$, $j = 1, 2, \ldots, n_{\text{E}}$, have been defined such that they contain all possible patterns of component matrices that can result in 4-cycles in $\mathbf{B}_{\text{R}}$, and hence in $\mathbf{B}_{\text{SC}}^{(L)}$. Therefore, if there are no 4-cycles in the chosen $\mathbf{B}_{\text{C}}$ and the associated $\mathbf{B}_{\text{E}}^{(j)}, j = 1, 2, \ldots, n_{\text{E}}$, there are no 4-cycles in $\mathbf{B}_{\text{SC}}^{(L)}$. $\blacksquare$

Based on Theorem 3, we now proceed to design the component matrices $\mathbf{B}_i$ such that neither the chosen $\mathbf{B}_{\text{C}}$ nor its associated $\mathbf{B}_{\text{E}}^{(j)}$ contain any 4-cycles. The proposed design includes two stages: Stage 1 initializes the component matrices based on the $\mathbf{B}_{\text{E}}^{(j)}$; Stage 2 modifies the component matrices based on $\mathbf{B}_{\text{C}}$.

B. Design Stage 1

Given a base matrix $\mathbf{B} = \mathbf{1}_{n_c \times n_v}$ and a representative block $\mathbf{B}_{\mathrm{R}}$ with coupling width ω, a constituent block $\mathbf{B}_{\mathrm{C}}$ is chosen and its associated excluded patterns $\mathbf{B}_{\mathrm{E}}^{(j)}$ are determined. Then the Stage 1 design insures that the component matrices and the excluded patterns do not contain any 4-cycles. Let $\mathbf{I}_{n_c}$ denote the $n_c \times n_c$ identity matrix. Furthermore, let $\Xi_{n_c \times (n_v - n_c)}$ denote an $n_c \times (n_v - n_c)$ binary matrix with a minimum row weight of one and a maximum column weight as small as possible.[4] Stage 1 is summarized as Design Rule 1 below.

Design Rule 1 Initialize the component matrices (Stage 1)

1.1: Let $\mathbf{B}_0 = [\mathbf{I}_{n_c} \quad \Xi_{n_c \times (n_v - n_c)}]$, where $\Xi_{n_c \times (n_v - n_c)}$ is chosen as above such that there is no 4-cycle in $\mathbf{B}_0$.

1.2: Initialize $\mathbf{B}_\omega$ such that it contains no 4-cycle and there are no 4-cycles in $\mathbf{B}_{\mathrm{E}}^{(1)}$ or $\mathbf{B}_{\mathrm{E}}^{(2)}$, the minimum row weight of $\mathbf{B}_\omega$ is two, and, given these constraints, $\mathbf{B}_\omega$ has a maximum column weight as small as possible.

1.3: Initialize $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_{\omega-1}$ such that

1) $\mathbf{B}(r,s) = \sum_{i=0}^{\omega} \mathbf{B}_i(r,s)$, i.e., (2) is satisfied;

2) There is no 4-cycle in any of the component matrices $\mathbf{B}_i$, $i = 1, 2, \ldots, \omega - 1$, or in the excluded patterns $\mathbf{B}_{\mathrm{E}}^{(j)}$ $(j = 3, 4, \ldots, n_{\mathrm{E}})$.

In the above design, note that Step **1.1** insures that the minimum row weight of $\mathbf{B}_0$ is two. Requiring each row of $\mathbf{B}_0$ and $\mathbf{B}_\omega$ to have a minimum row weight of two is desirable since they are the only component matrices in the top and bottom rows of $\mathbf{B}_{\mathrm{SC}}^{(L)}$, respectively, and a row weight of at least two is needed to assist the startup and termination of decoding. Since $\mathbf{B}_0 = [\mathbf{I}_{n_c} \quad \Xi_{n_c \times (n_v - n_c)}]$, we must ensure $\Xi_{n_c \times (n_v - n_c)}$ has no 4-cycles. When $n_v - n_c \geq n_c$, this is equivalent to ensuring the maximum column weight of $\Xi_{n_c \times (n_v - n_c)}$ is one. However, when $n_v - n_c < n_c$, maintaining a minimum row weight of one for $\Xi_{n_c \times (n_v - n_c)}$ will inevitably make the maximum column weight greater than one. In this case, $\Xi_{n_c \times (n_v - n_c)}$ must be designed such that no set of four 1s will appear in a rectangular array. Furthermore, restricting the maximum column weights of $\mathbf{B}_0$ and $\mathbf{B}_\omega$ to be as small as possible simplifies Step **1.3**. The remaining

[4] If $n_v - n_c \geq n_c$ $(R_{\mathrm{SC}}^{(\infty)} \geq \frac{1}{2})$, the maximum column weight of $\Xi_{n_c \times (n_v - n_c)}$ can be as low as one. If $n_v - n_c < n_c$ $(R_{\mathrm{SC}}^{(\infty)} < \frac{1}{2})$, it will have a maximum column weight greater than one.

component matrices $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_{\omega-1}$ are then initialized based on the already chosen $\mathbf{B}_0$ and $\mathbf{B}_\omega$ and avoiding 4-cycles in the excluded patterns $\mathbf{B}_{\mathrm{E}}^{(j)} (j = 3, 4, \ldots, n_{\mathrm{E}})$. The following example illustrates Design Rule 1.

Example 2. Given $\mathbf{B} = \mathbf{1}_{3\times 8}$ and $\omega = 4$, $\mathbf{B}_{\mathrm{R}}$, $\mathbf{B}_{\mathrm{C}}$, $\mathbf{B}_{\mathrm{E}}^{(1)}$, $\mathbf{B}_{\mathrm{E}}^{(2)}$, $\mathbf{B}_{\mathrm{E}}^{(3)}$, and $\mathbf{B}_{\mathrm{E}}^{(4)}$ are given in *Example 1*, where $\mathbf{B}_{\mathrm{C}}$ is the $3n_c \times 3n_v$ pattern, i.e., $\alpha = \beta = 3$. We employ Design Rule 1 to initialize the component matrices. First, we let

$$\mathbf{B}_0 = [\mathbf{I}_3 \quad \mathbf{\Xi}_{3\times 5}] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Placing $\mathbf{B}_0$ into $\mathbf{B}_{\mathrm{E}}^{(2)}$, we then initialize

$$\mathbf{B}_4 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

such that neither $\mathbf{B}_{\mathrm{E}}^{(1)}$ nor $\mathbf{B}_{\mathrm{E}}^{(2)}$ contains any 4-cycles. In order to initialize $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$, we place both $\mathbf{B}_0$ and $\mathbf{B}_4$ into the excluded patterns $\mathbf{B}_{\mathrm{E}}^{(3)}$ and $\mathbf{B}_{\mathrm{E}}^{(4)}$. To satisfy (2), as well as to avoid 4-cycles in these two excluded patterns, we design $\mathbf{B}_1$ and $\mathbf{B}_3$ as

$$\mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{B}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and then $\mathbf{B}_2$ is given by

$$\mathbf{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The above design ensures that there are no 4-cycles in all the component matrices and excluded patterns.

Finally, we place the initialized component matrices into $\mathbf{B}_{\mathrm{C}}$ and check if it contains any 4-cycles. If not, the coupled protograph of $\mathbf{B}_{\mathrm{SC}}^{(L)}$ has $g \geq 6$, and the design is complete. Otherwise, we proceed to Stage 2 to eliminate (or reduce the number of) 4-cycles that remain in $\mathbf{B}_{\mathrm{C}}$. In this example, there are two 4-cycles in $\mathbf{B}_\text{C}$, so Stage 2 is needed to further modify the component matrices. ■

C. Design Stage 2

Design Rule 2 Modify the component matrices (Stage 2)

2.1: Identify a 4-cycle in $\mathbf{B}_\text{C}$ with entries

$$\mathbf{B}_\text{C}(x_1, y_1) = 1,\ \mathbf{B}_\text{C}(x_1, y_2) = 1,$$
$$\mathbf{B}_\text{C}(x_2, y_1) = 1,\ \mathbf{B}_\text{C}(x_2, y_2) = 1.$$

2.2: Suppose that the four entries belong to component matrices $\mathbf{B}_{i_1}$, $\mathbf{B}_{i_2}$, $\mathbf{B}_{i_3}$, and $\mathbf{B}_{i_4}$, where $(i_1, i_2, i_3, i_4) \in \{0, 1, \ldots, \omega\}$. Denote these entries as

$$\mathbf{B}_{i_1}(r_1, s_1) = 1,\ \mathbf{B}_{i_2}(r_1, s_2) = 1,$$
$$\mathbf{B}_{i_3}(r_2, s_1) = 1,\ \mathbf{B}_{i_4}(r_2, s_2) = 1.$$

2.3: Among these four entries, identify those that have not previously been *flipped*. Pick one that belongs to a component matrix of the highest weight and denote it $\mathbf{B}_{i'}(r', s')$, where $i' \in \{i_1, i_2, i_3, i_4\}$ and $(r', s') \in \{(r_1, s_1), (r_1, s_2), (r_2, s_1), (r_2, s_2)\}$. *Flip down* the entry $\mathbf{B}_{i'}(r', s')$ such that

$$\mathbf{B}_{i'}(r', s') :\ 1\ \rightarrow\ 0, \tag{7}$$

Also flip down all entries in $\mathbf{B}_\text{E}^{(j)} (j = 1, 2, \ldots, n_\text{E})$ and $\mathbf{B}_\text{C}$ that correspond to entry $\mathbf{B}_{i'}(r', s')$.

2.4: *Flip up* an entry $\mathbf{B}_i(r', s')$ such that

$$\mathbf{B}_i(r', s') :\ 0\ \rightarrow\ 1, \tag{8}$$

where $i \in \{0, 1, \ldots, \omega\}$ and $i \neq i'$, conditioned on

1) The entry has not been previously flipped (down or up);
2) The flipping does not create new 4-cycles in $\mathbf{B}_i$ or in any $\mathbf{B}_\text{E}^{(j)}$ that includes $\mathbf{B}_i$;
3) The number of 4-cycles contained in $\mathbf{B}_\text{C}$ does not increase after the flipping (down and up) process is completed.

Also flip up all entries in $\mathbf{B}_\text{E}^{(j)} (j = 1, 2, \ldots, n_\text{E})$ and $\mathbf{B}_\text{C}$ that correspond to entry $\mathbf{B}_i(r', s')$.

2.5: If the flipping in Step **2.4** succeeds, go to Step **2.6**; else, reflip $\mathbf{B}_{i'}(r', s')$ to its original value, i.e.,

$$\mathbf{B}_{i'}(r', s') :\ 0\ \rightarrow\ 1. \tag{9}$$

Also reflip all entries in $\mathbf{B}_\text{E}^{(j)} (j = 1, 2, \ldots, n_\text{E})$ and $\mathbf{B}_\text{C}$ that correspond to entry $\mathbf{B}_{i'}(r', s')$, and go to Step **2.3**.

2.6: Repeat Steps **2.1** to **2.4** until all 4-cycles are removed or there are no more eligible entries to flip.

Stage 2 modifies the initialized component matrices to remove the remaining 4-cycles in $\mathbf{B}_\text{C}$, if possible. In order to further distinguish between an entry in $\mathbf{B}_\text{C}$ and one in $\mathbf{B}_i$, we use $\mathbf{B}_\text{C}(x, y)$

to denote the row-x column-y entry in $\mathbf{B}_{\text{C}}$, for $x = 1, \ldots, n_c$ and $y = 1, \ldots, \beta n_v$. Stage 2 is characterized by a so-called *check-and-flip process* in which, if a 4-cycle exists in $\mathbf{B}_{\text{C}}$, one of its nonzero entries is flipped down from 1 to 0. Since the flipped entry belongs to a component matrix, maintaining (2) requires that we also flip up an entry from 0 to 1 in one of the other component matrices. But this flipping should not create 4-cycles in this component matrix or in the excluded patterns that contain this component matrix. Moreover, the number of 4-cycles in $\mathbf{B}_{\text{C}}$ should not increase. Stage 2 is summarized as Design Rule 2 below.

In Step **2.2** of the design, i_1, i_2, i_3, and i_4 do not need to be distinct. In Step **2.3**, we prioritize the "flipping down" of a nonzero entry of a component matrix that has maximum weight in $\mathbf{B}_{\text{C}}$. In doing so, we remove the most nonzero entries in $\mathbf{B}_{\text{C}}$, so that more 4-cycles are likely to be removed. It is possible that none of the four entries in an identified 4-cycle allows a complete flipping (both flipping down and flipping up), in which case the remaining 4-cycle is labelled *dormant*. However, a dormant 4-cycle can be targeted for flipping again if some other complete flipping occurs and it still exists. For small coupling widths ω, though, it may not be possible to eliminate all dormant 4-cycles in $\mathbf{B}_{\text{C}}$. Increasing ω allows more freedom in the design, making it easier to eliminate 4-cycles in $\mathbf{B}_{\text{C}}$. However, this also increases the design complexity, since the constituent block $\mathbf{B}_{\text{C}}$ is larger and there are more component matrices $\mathbf{B}_i$ and excluded patterns $\mathbf{B}_{\text{E}}^{(j)}$ to consider, meaning that the flipping up of Step **2.4** must satisfy more conditions. In general, Stage 2 either eliminates all 4-cycles or minimizes the number of 4-cycles in $\mathbf{B}_{\text{C}}$ and, as a result, in $\mathbf{B}_{\text{SC}}^{(L)}$. The following example illustrates Design Rule 2.

Example 3. Given the component matrices $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_4$ that were initialized in *Example 2*, we form the constituent block $\mathbf{B}_{\text{C}}$ shown in Fig. 3(a). We see that there are two 4-cycles, defined by entries $\mathbf{B}_{\text{C}}(1, 7)$, $\mathbf{B}_{\text{C}}(1, 20)$, $\mathbf{B}_{\text{C}}(8, 7)$, and $\mathbf{B}_{\text{C}}(8, 20)$ (or $\mathbf{B}_2(1, 7), \mathbf{B}_0(1, 4), \mathbf{B}_4(2, 7)$, and $\mathbf{B}_2(2, 4)$ in component matrix notation), and $\mathbf{B}_{\text{C}}(1, 17)$, $\mathbf{B}_{\text{C}}(1, 20)$, $\mathbf{B}_{\text{C}}(9, 17)$, and $\mathbf{B}_{\text{C}}(9, 20)$ (or $\mathbf{B}_0(1, 1), \mathbf{B}_0(1, 4), \mathbf{B}_2(3, 1)$, and $\mathbf{B}_2(3, 4)$ in component matrix notation), which are highlighted by the squares.

We now apply Design Rule 2 to remove these 4-cycles. Take the 4-cycle defined by $\mathbf{B}_{\text{C}}(1, 7)$, $\mathbf{B}_{\text{C}}(1, 20)$, $\mathbf{B}_{\text{C}}(8, 7)$, and $\mathbf{B}_{\text{C}}(8, 20)$ as an example, which belong to component matrices $\mathbf{B}_2$, $\mathbf{B}_0$, $\mathbf{B}_4$, and $\mathbf{B}_2$, respectively. Since *wt*$(\mathbf{B}_2) = 3$ and *wt*$(\mathbf{B}_0) =$ *wt*$(\mathbf{B}_4) = 1$, entry $\mathbf{B}_{\text{C}}(1, 7)$ is chosen to be flipped down from 1 to 0, so that $i' = 2$ and $(r', s') = (1, 7)$, and we also flip down the other two $\mathbf{B}_2(1, 7)$ entries that appear in $\mathbf{B}_{\text{C}}$. (Note that $\mathbf{B}_2$ is not contained in any of the excluded

$$\mathbf{B}_C = \begin{bmatrix} \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 \\ \mathbf{B}_4 & \mathbf{B}_3 & \mathbf{B}_2 \end{bmatrix}$$

$$= \left[\begin{array}{cccccccc|cccccccc|cccccccc}
0&0&0&0&0&0&\boxed{1}&1&0&0&0&0&0&0&0&0&\boxed{1}&0&0&\boxed{1}&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1\\
1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
\hline
0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
\hline
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&1\\
1&0&0&0&0&0&\boxed{1}&0&0&0&1&0&0&1&0&0&0&0&0&\boxed{1}&0&0&0&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&\boxed{1}&0&0&\boxed{1}&1&0&0&0
\end{array}\right]$$

(a) The original $\mathbf{B}_C$,

$$= \left[\begin{array}{cccccccc|cccccccc|cccccccc}
0&0&0&0&0&0&\textcircled{0}&1&0&0&0&0&0&0&0&0&\boxed{1}&0&0&\boxed{1}&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1\\
1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
\hline
0&0&0&0&1&0&\triangle 1&0&0&0&0&0&0&0&\textcircled{0}&1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
\hline
0&1&1&0&0&0&0&0&0&0&0&0&1&0&\triangle 1&0&0&0&0&0&0&0&\textcircled{0}&1\\
1&0&0&0&0&0&\boxed{1}&0&0&0&1&0&0&1&0&0&0&0&0&\boxed{1}&0&0&0&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&\boxed{1}&0&0&\boxed{1}&1&0&0&0
\end{array}\right]$$

(b) The modified $\mathbf{B}_C$ after one complete flipping,

$$= \left[\begin{array}{cccccccc|cccccccc|cccccccc}
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
\hline
0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&\boxed{1}&0&0&\boxed{1}&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
\hline
0&\boxed{1}&\boxed{1}&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1\\
1&0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0\\
0&\boxed{1}&0&0&0&\boxed{1}&0&0&1&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0
\end{array}\right]$$

(c) The modified $\mathbf{B}_C$ after 4 flippings.

Fig. 3. The Stage 2 design of Example 3.

patterns for this choice of $\mathbf{B}_\mathrm{C}$.) Fig. 3(b) shows the modified $\mathbf{B}_\mathrm{C}$ in which the three flipped down $\mathbf{B}_2(1,7)$ entries are highlighted by the circles. In order to maintain the edge-spreading condition (see (2)), we must flip up an entry $\mathbf{B}_i(1,7)$, where $i \neq 2$. Since $\mathbf{B}_4(1,7), \mathbf{B}_3(1,7), \mathbf{B}_1(1,7)$, and $\mathbf{B}_0(1,7)$ have not been previously flipped, we choose the two $\mathbf{B}_3(1,7)$ entries, corresponding to $\mathbf{B}_\mathrm{C}(4,7)$ and $\mathbf{B}_\mathrm{C}(7,15)$, to be flipped up from 0 to 1, which is highlighted by the triangles in Fig. 3(b). We then check and find that this flipping up step does not create new 4-cycles in $\mathbf{B}_3$ or in the excluded patterns $\mathbf{B}_\mathrm{E}^{(1)}$, $\mathbf{B}_\mathrm{E}^{(2)}$, $\mathbf{B}_\mathrm{E}^{(3)}$, or $\mathbf{B}_\mathrm{E}^{(4)}$ and that it does not increase the number of 4-cycles in $\mathbf{B}_\mathrm{C}$. As a result, the targeted 4-cycle in $\mathbf{B}_\mathrm{C}$ has been removed. The other 4-cycle in $\mathbf{B}_\mathrm{C}$ can be removed in a similar manner. In all, a total of 4 flippings (flipping down $\mathbf{B}_\mathrm{C}(1,7)$ and $\mathbf{B}_\mathrm{C}(3,1)$, and flipping up $\mathbf{B}_\mathrm{C}(4,7)$ and $\mathbf{B}_\mathrm{C}(6,1)$) are required to remove the two 4-cycles in $\mathbf{B}_\mathrm{C}$. The resulting $\mathbf{B}_\mathrm{C}$ is shown in Fig. 3(c). Its girth is six, which is highlighted in the figure. As a result, we can insure that $\mathbf{B}_\mathrm{SC}^{(L)}$ also has girth $g = 6$. ■

Intuitively, the above design can be seen as guiding the spreading of the $n_c n_v$ 1's in $\mathbf{B}$ over the $\omega + 1$ component matrices in such a way that 4-cycles are eliminated in the coupled protograph. As another example, given the block protograph defined by $\mathbf{B} = \mathbf{1}_{2\times 4}$ and coupling width $\omega = 2$, we can apply the above two stage design to obtain

$$\mathbf{B}_0 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

thus insuring that the coupled protograph of $\mathbf{B}_\mathrm{SC}^{(L)}$ does not contain any 4-cycles. The resulting coupled protograph, which in this case has girth $g = 12$, is shown in Fig. 4.

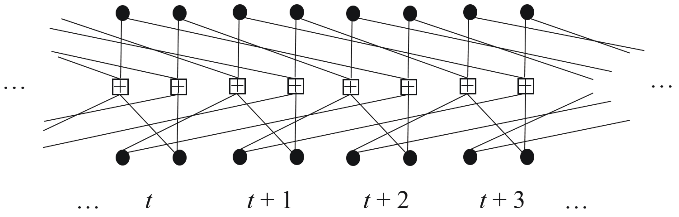


Fig. 4. A coupled protograph with coupling width $\omega = 2$ and girth $g = 12$.

If Stage 2 does not eliminate all the 4-cycles in $\mathbf{B}_\mathrm{C}$[5] for a particular set of initial component

[5] We note here that it is easier to design a $\mathbf{B}_\mathrm{R}$ that does not contain any 4-cycles for larger values of ω, since the individual component matrices are sparser.

matrices $\mathbf{B}_i, i = 0, 1, \ldots, \omega$, designed in Stage 1, or if it results in a $\mathbf{B}_0$ or $\mathbf{B}_\omega$ with minimum row weight less than two (for example, in the coupled protograph of Fig. 4, the minimum row weight of $\mathbf{B}_2$ is one), the design can be repeated with a different set of initial component matrices. Alternatively, one can impose maintaining a minimum row weight of two for $\mathbf{B}_0$ and $\mathbf{B}_\omega$ as a constraint when flipping down in Step **2.3** (although this may limit our ability to find a 4-cycle free $\mathbf{B}_\mathrm{C}$). In the case of unterminated codes (or codes with large L) that are decoded with sliding window (SW) decoding, the row weight of $\mathbf{B}_\omega$ has little effect on performance, and hence the constraint on $\mathbf{B}_\omega$ can be relaxed. Further, for $\omega > 2$, there are multiple choices for $\mathbf{B}_\mathrm{C}$, each of which is associated with a different set of excluded patterns. Our experience has shown that the choice of $\mathbf{B}_\mathrm{C}$ does not affect whether or not 4-cycles can be eliminated (although choosing different sets of initial component matrices can yield different results). This follows from the fact that, when using a different constituent block and its associated excluded patterns, the two-stage design has already checked through all possible patterns that could contain 4-cycles. In other words, different choices of $\mathbf{B}_\mathrm{C}$, along with the associated excluded patterns $\mathbf{B}_\mathrm{E}^{(j)}$, and the subsequent component matrix initialization only affect the scheduling of the flipping, not whether 4-cycles can be eliminated.[6] Finally, we note that, after achieving girth $g = 6$, the design procedure can be continued in an attempt to reduce the number of 6-cycles in the coupled protograph. The way in which finding a 4-cycle free $\mathbf{B}_\mathrm{R}$ depends on the code parameters n_c, n_v, and ω will be discussed in the following subsection.

D. Coupling Width Required to Eliminate 4-Cycles

In order to apply our approach to other SC protographs, it is helpful to identify what is the minimum ω required to achieve girth 6, since we will typically want small ω to minimize latency. Based on the base matrix $\mathbf{B} = \mathbf{1}_{n_c \times n_v}$, the proposed design can be seen as assigning the $n_c n_v$ 1s in $\mathbf{B}$ to the $\omega + 1$ component matrices $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\omega$ in a way such that $\mathbf{B}_\mathrm{R}$ does not contain 4-cycles. Intuitively, a larger ω gives more design freedom, so that the Stage 1 design can already ensure that $\mathbf{B}_\mathrm{R}$ does not contain 4-cycles when ω is sufficiently large. However, the proposed design can result in many possibilities for the designed component matrices. Due to the heuristic nature of the design, it is difficult to theoretically characterize the minimum ω for insuring a

[6] The search complexity can be reduced by choosing $\mathbf{B}_\mathrm{C}$ in (5) such that $\alpha = \beta - 1$, β, or $\beta + 1$, which minimizes the number of excluded patterns (see *Example 1*).

TABLE I
THE MINIMUM w THAT INSURES $\mathbf{B}_{\text{R}}$ DOES NOT CONTAIN 4-CYCLES.

(n_c, n_v)	ω	(n_c, n_v)	ω	(n_c, n_v)	ω	(n_c, n_v)	ω
(2, 3)	1	(3, 4)	2	(4, 5)	2	(5, 6)	4
(2, 4)	2	(3, 5)	2	(4, 6)	3	(5, 7)	4
(2, 5)	2	(3, 6)	3	(4, 7)	4	(5, 8)	5
(2, 6)	3	(3, 7)	3	(4, 8)	5	(5, 9)	6
(2, 7)	3	(3, 8)	4	(4, 9)	5	(5, 10)	8
(2, 8)	4	(3, 9)	4	(4, 10)	6		
(2, 9)	4	(3, 10)	5				
(2, 10)	5	(3, 11)	5				
(2, 11)	5	(3, 12)	6				
(2, 12)	6						

4-cycle free $\mathbf{B}_{\text{R}}$.[7] Alternatively, we have employed an exhaustive search for the minimum ω resulting from the proposed design with given (n_c, n_v) pairs. Our results obtained using Design Rules 1 and 2, thus ensuring the minimum row weight of 2 for the initial $\mathbf{B}_0$ and $\mathbf{B}_\omega$ needed to assist the startup and termination of decoding, are shown in Table I. Our search results echo some existing characterizations in the literature of the minimum coupling width needed to ensure a coupled protograph contains no 4-cycles. For example, our results match well with Lemma 4 of [16], which gives a lower bound on ω needed to ensure a 4-cycle-free SC parity-check matrix for a time-invariant construction of the same type as our more general time-varying construction. Also, when $n_c = 2$, our search results exactly match the coupled protograph design of [33], for which the minimum ω that ensures no 4-cycles is $\frac{n_v}{2}$ when n_v is even and $\frac{n_v-1}{2}$ when n_v is odd.

We observe that, for the same n_c, a larger n_v requires a larger ω to insure a 4-cycle free $\mathbf{B}_{\text{R}}$. For example, when $(n_c, n_v) = (2, 6)$, the minimum ω is 3, while when $(n_c, n_v) = (2, 9)$, the minimum ω is 4. This is because increasing n_v not only leads to more 1s to be assigned to the component matrices, but the asymptotic rate $R_{\text{SC}}^{(\infty)} = 1 - \frac{n_c}{n_v}$ of the designed code also becomes higher, making it more challenging to design a matrix $\mathbf{B}_{\text{R}}$ that does not contain 4-cycles. This can also be seen by comparing designs with the same $n_c n_v$ (the same number of 1s to be assigned), where a smaller ratio of $\frac{n_c}{n_v}$ (a higher asymptotic rate) requires a larger coupling width. For

[7] In recent papers [11] [15], optimization techniques were used to minimize the number of 6-cycles in array-based SC-LDPC codes. Due to the complexity of the optimization, however, this approach is limited to values of $\omega \leq 2$, whereas our heuristic design allows us to reach much larger values of ω.

example, when $(n_c, n_v) = (4, 6)$, the minimum ω is 3, while the minimum values of ω are 4 and 6 for $(n_c, n_v) = (3, 8)$ and $(n_c, n_v) = (2, 12)$, respectively. This is because the component matrices of higher rate codes have relatively more columns than rows, making it easier to form 4-cycles, thus requiring a larger ω to spread the 1s over more component matrices. Finally, for the same asymptotic rate, we see that increasing n_c (and n_v) also leads to an increase in the minimum required ω. For example, $(n_c, n_v) = (2, 4)$ requires $\omega = 2$, while $(n_c, n_v) = (3, 6)$ and $(n_c, n_v) = (4, 8)$ require $\omega = 3$ and $\omega = 5$, respectively. This is because, for the same ratio $\frac{n_c}{n_v}$, a larger n_c (and n_v) means that there are more 1s to be assigned to the component matrices, and consequently a larger ω is required.

IV. BP Decoding Thresholds of the Designed Code Ensembles

The above coupled protograph design ensures that the constructed codes have large girth, thereby yielding good error floor performance. However, the design does not guarantee good waterfall performance. Therefore, we now compare the BP decoding thresholds of several designed SC-LDPC protograph-based code ensembles over the BEC to those of undesigned (random edge spreading) protograph-based ensembles constructed from the same all-one base matrix $\mathbf{1}_{n_c \times n_v}$. Note that the Shannon limit of a BEC is $1 - \epsilon$, where ϵ is the erasure probability.

Fig. 5 shows the BP decoding thresholds of the designed and undesigned protograph-based ensembles with different choices of n_c, n_v, and ω, where the coupling widths ω are chosen such that the designed protographs have girth $g = 6$. It can be seen that, in all cases, the BEC decoding threshold decreases as the coupling length L increases (due to the increasing rate $R_{\text{SC}}^{(L)}$) up to a certain point, after which the BP decoding thresholds of the SC-LDPC code ensembles saturate and approach the MAP decoding thresholds of their underlying LDPC block code ensembles (0.488, 0.497, and 0.499 for (3, 6), (4, 8), and (5, 10)-regular ensembles, respectively). This is due to the structured irregularity at the beginning and end of the coupled graph that results in threshold saturation and is consistent with [3], [4], [5], [6].

We observe that the designed and undesigned ensembles have similar BP decoding thresholds, especially when L is large. This is due to the fact that our design does not change the row weight of a full diagonal band (that contains $\mathbf{B}_\omega$, $\mathbf{B}_{\omega-1}$, ..., $\mathbf{B}_0$) or the column weight of the SC base matrix, and thus it has the same degree profile as an undesigned protograph. Comparing the ensembles with different column (and row) weights, we see that, for the same asymptotic rate

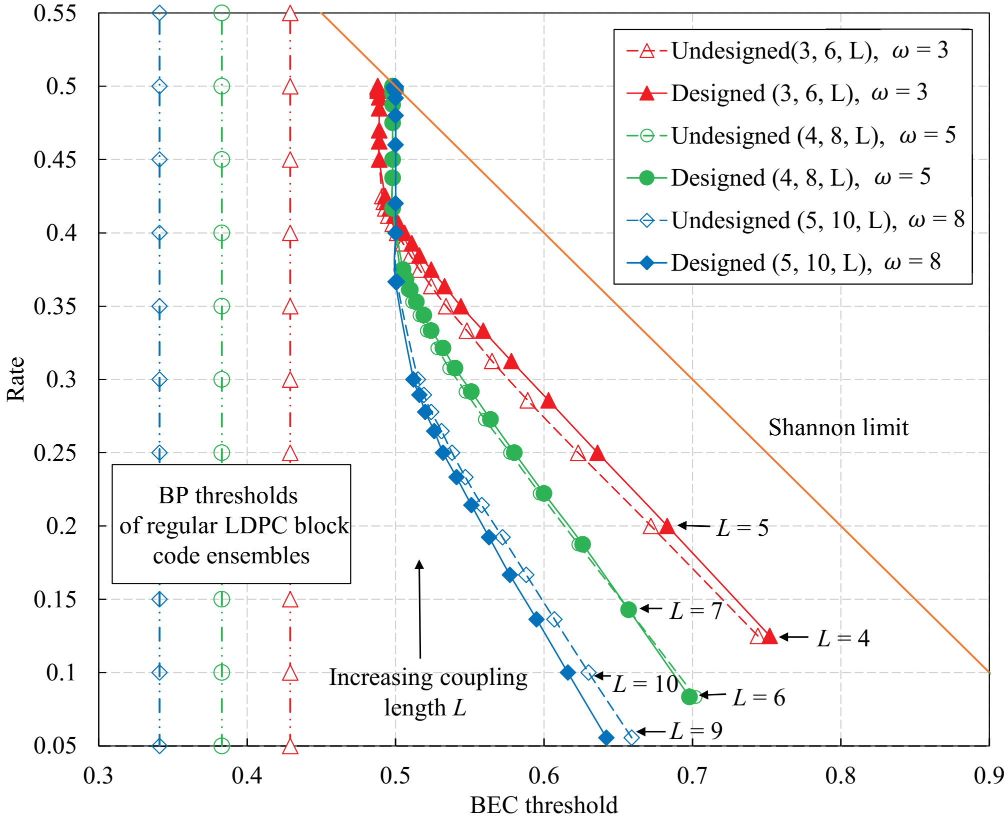


Fig. 5. BP decoding thresholds of designed and undesigned SC-LDPC code protograph-based ensembles.

$R_{\text{SC}}^{(\infty)} = 0.5$, increasing the column (and row) weight leads to a decrease in the BP decoding threshold for small values of L, where the termination causes significant rate loss. But larger column (and row) weights lead to better BP decoding thresholds as L becomes larger and tends to infinity, where the rate loss approaches zero. This follows from the fact that block code ensembles with larger column (and row) weights have better minimum distance properties, and thus their MAP decoding thresholds are better. In contrast, for regular LDPC block code ensembles, where threshold saturation does not occur, higher column (and row) weights result in worse BP decoding thresholds. In summary we see that the BP thresholds of the designed ensembles closely track those of undesigned ensembles and approach the MAP thresholds of the underlying block code ensembles for large L, thus insuring that the designed codes also achieve good waterfall performance.

$(r_1, s_1) : \theta_1$ ○	$(r_1, s_2) : \theta_2$ ●	$(r_1, s_3) : \theta_3$ ●
$(r_2, s_1) : \theta_4$ ●	$(r_2, s_2) : \theta_5$ ●	$(r_2, s_3) : \theta_6$ ○
$(r_3, s_1) : \theta_7$ ●	$(r_3, s_2) : \theta_8$ ○	$(r_3, s_3) : \theta_9$ ●

Fig. 6. A 3×3 grid of nonzero entries corresponding to a 6-cycle in $\mathbf{B}_{\text{SC}}^{(L)}$, where $r_1 < r_2 < r_3$ and $s_1 < s_2 < s_3$.

V. QC Lifting Based on $\mathbf{B}_{\text{SC}}^{(L)}$

Given a designed $\mathbf{B}_{\text{SC}}^{(L)}$, we can employ a systematic lifting using circulants in an attempt to further reduce the multiplicity of short cycles and increase the girth. In this paper, we pay particular attention to the removal of all 6-cycles (and any remaining 4-cycles) so that the resulting $\mathbf{H}_{\text{SC}}^{(L)}$ is QC and has $g \geq 8$, although the approach could be extended in a straightforward way to target higher girth. Note that, without loss of generality, any 6-cycle can be represented by a 3×3 grid of nonzero entries in $\mathbf{B}_{\text{SC}}^{(L)}$, as illustrated in Fig. 6 for the 6-cycle highlighted in Fig. 3(c). In general, circulants can be chosen using the Fossorier condition (Theorem 2.1 of [8]) to avoid a $2k$-cycle, $k = 2, 3, \ldots$, in a parity-check matrix. For example, if the six nonzero entries that constitute the 6-cycle in Fig. 6 (indicated by the solid circles) are lifted with different circulants $\mathbf{I}_M^{(\theta)}$, where $\mathbf{I}_M^{(\theta)}$ denotes the shifted identity matrix with each row of the $M \times M$ identity matrix $\mathbf{I}_M$ cyclically shifted to the left by θ positions, and the shifting factors satisfy

$$(\theta_3 - \theta_9) \neq (\theta_2 - \theta_5) + (\theta_4 - \theta_7) \mod M, \tag{10}$$

then there are no 6-cycles in the lifted subgraph corresponding to $\mathbf{H}_{\text{SC}}^{(L)}$ associated with this 6-cycle in $\mathbf{B}_{\text{SC}}^{(L)}$. In this case, we say that the 6-cycle in the protograph has been "removed" by lifting. In general, a QC lifting based on $\mathbf{B}_{\text{SC}}^{(L)}$ results in non-periodically time-varying SC-LDPC codes, but for ease of implementation it is desirable to construct periodically time-varying, or even time-invariant codes [31].[8] We treat these two cases separately below.

[8] Note that, if we use the same set of circulants for lifting every column of $\mathbf{B}_{\text{SC}}^{(L)}$, we will obtain a time-invariant QC-SC-LDPC code. However, the time-varying designs give us added flexibility, making it easier to achieve girth 8 for a given M. Also, if M is too small, it may not be possible to achieve girth 8 at all with the time-invariant constraint.

A. The Non-Periodic Case

We start by identifying all the nonzero entries that participate in 6-cycles of $\mathbf{B}_{\text{SC}}^{(L)}$. The identified 6-cycles are then removed sequentially by selecting circulants according to the Fossorier condition. (The remaining nonzero entries can be lifted using randomly generated circulants.) However, since nonzero entries can participate in multiple 6-cycles, care must be taken to insure that cycle removal does not create new short cycles elsewhere in the graph. In our approach, the entries and shifting factors are recorded after removing each cycle. Before a new 6-cycle is targeted for lifting, we first check to see if any of its nonzero entries have been previously lifted. If so, they are left unchanged, and the shifting factors of the other nonzero entries in the cycle are chosen such that the Fossorier condition is satisfied (if possible). For sufficiently large M, we have found that there is enough freedom in choosing the shifting factors to construct non-periodically time-varying QC-SC-LDPC codes with $g \geq 8$.

B. The Periodic Case

To construct periodically time-varying QC-SC-LDPC codes with period $\omega+1$, we can proceed by choosing the shifting factors for only the nonzero entries in the first $\omega+1$ columns of $\mathbf{B}_{\text{SC}}^{(L)}$ that participate in 6-cycles. In the following columns, the shifting factors in every set of $\omega+1$ columns of $\mathbf{B}_{\text{SC}}^{(L)}$ will be a replication of those in the first $\omega+1$ columns. Note that this gives us a more efficient lifting than in the non-periodic case, since we do not have to check the entire coupled protograph of length L for 6-cycle removal, but only the first $\omega+1$ columns of $\mathbf{B}_{\text{SC}}^{(L)}$.

$\mathbf{B}_{\text{SC}}^{(L)}$ can be seen as consisting of L columns of component matrices $\mathbf{B}_0$, $\mathbf{B}_1$, …, $\mathbf{B}_\omega$, as shown in Fig. 7, where we index the columns as $\tau = 1, 2, \ldots, \omega, \omega+1, \ldots, L$. For $n_c > 2$, 6-cycles can be contained in component matrices that occupy the same row (or column) of $\mathbf{B}_{\text{SC}}^{(L)}$. However, from Fig. 7 we can see that 6-cycles will not be contained in any two columns of component matrices, indexed by τ_1 and τ_2, respectively, if $|\tau_1 - \tau_2| > \omega$, since in this case the two columns will not have any component matrices that occupy the same row. This observation enables us to design the shifting factors for the first $\omega+1$ columns of $\mathbf{B}_{\text{SC}}^{(L)}$ as follows. We first design the column 1 shifting factors such that all its 6-cycles are removed. The designed shifting factors of column 1 are then replicated in column $\omega+2$. Next, we design column 2 such that all 6-cycles that exist in column 2 alone, jointly between columns 1 and 2, and jointly between columns 2 and $\omega+2$ are removed. Then the designed shifting factors of column 2 are replicated in column

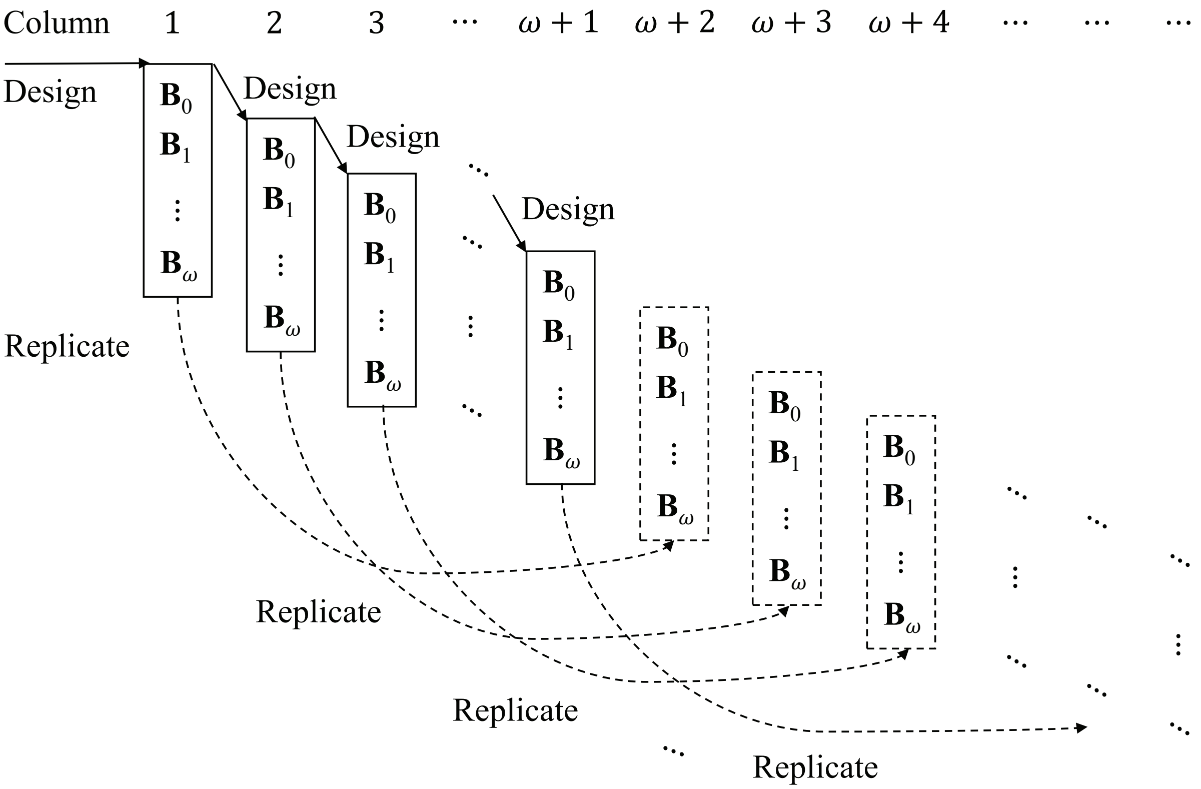


Fig. 7. Periodic design of QC-SC-LDPC codes.

$\omega+3$. Since this insures that the joint 6-cycles between columns 1 and 2 are removed and the shifting factors of column $\omega+2$ are replicas of those in column 1, there must also be no joint 6-cycles between columns $\omega+2$ and $\omega+3$. This process then continues until we design column $\omega+1$ such that all 6-cycles that exist within column $\omega+1$ alone and jointly between column $\omega+1$ and the previously designed columns (1, 2, …, ω, $\omega+2$, $\omega+3$, …, $2\omega+1$) are removed. Following the design of column $\omega+1$, all the shifting factors for the nonzero entries of all the component matrices in the first $\omega+1$ columns that participate in 6-cycles have been chosen, i.e., no 6-cycles exist in the first $\omega+1$ columns of $\mathbf{B}_{\mathrm{SC}}^{(L)}$. This lifting design can then be replicated for every following set of $\omega+1$ columns of $\mathbf{B}_{\mathrm{SC}}^{(L)}$, so that the designed parity-check matrix $\mathbf{H}_{\mathrm{SC}}^{(L)}$ has period $\omega+1$ and girth $g \geq 8$. Again, a large lifting factor M gives more freedom in choosing the shifting factors to insure $g \geq 8$.

As an enhancement to the above procedure, in both the non-periodic and periodic cases, the 6-cycle profile of the SC protograph could be generated. We could then determine the shifting factors for the nonzero entries of $\mathbf{B}_{\text{SC}}^{(L)}$ that are involved in the most cycles, followed by the others in decreasing order. This would improve our ability to eliminate 6-cycles, at a cost of the increased complexity of ordering the nonzero entries of $\mathbf{B}_{\text{SC}}^{(L)}$ according to their cycle involvement.

The complexity of our proposed multi-stage code design approach depends on designing the SC base matrix $\mathbf{B}_{\text{SC}}$ to avoid 4-cycles and then graph-lifting based on $\mathbf{B}_{\text{SC}}$. The complexity of designing $\mathbf{B}_{\text{SC}}$ depends on the number of 4-cycles that remain in the constituent block $\mathbf{B}_{\text{C}}$ after initializing the component matrices in Stage 1. Similarly, the complexity of the lifting based on $\mathbf{B}_{\text{SC}}$ depends on the number of 6-cycles (and any remaining 4-cycles) in $\mathbf{B}_{\text{SC}}$. (In the case of periodic QC lifting, the cycle counting is limited to the first $\omega + 1$ columns only of $\mathbf{B}_{\text{SC}}$, which considerably simplifies the problem.) Therefore, obtaining the cycle profile of $\mathbf{B}_{\text{C}}$ following the Stage 1 design and the cycle profile of $\mathbf{B}_{\text{SC}}$ following the Stage 2 design is crucial to characterizing the design complexity. However, due to the heuristic nature of the design, an explicit characterization of the complexity is not feasible.

VI. Numerical Results

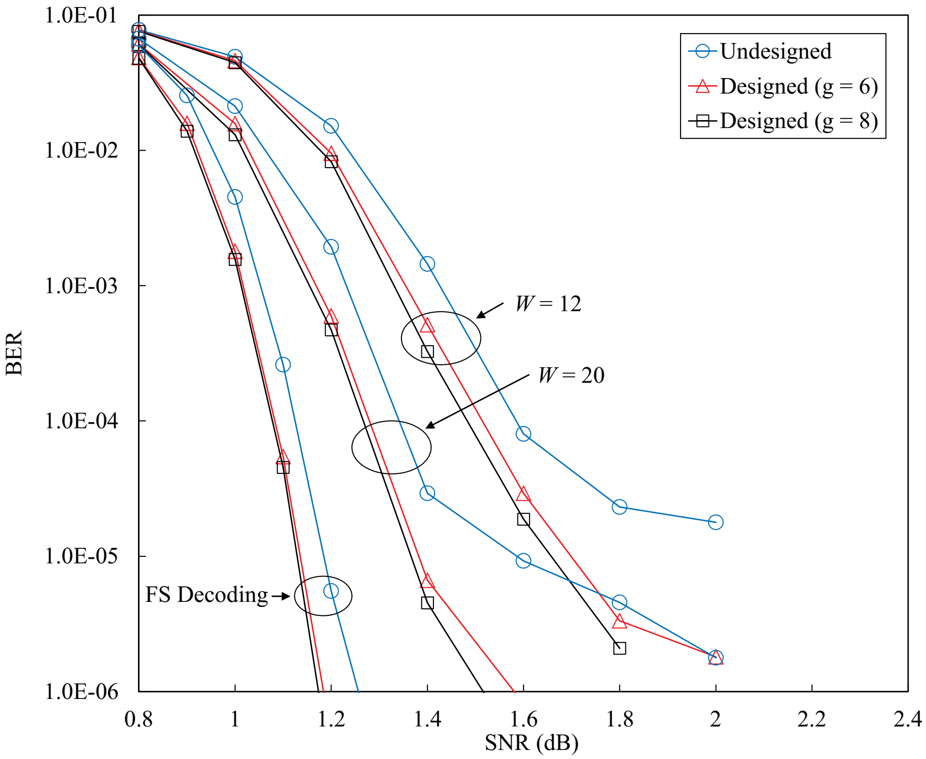


Fig. 8. Performance of designed and undesigned SC-LDPC codes with $(n_c, n_v) = (3, 6)$.

In this section, we present the simulated performance of our designed spatially coupled base matrices $\mathbf{B}_{\text{SC}}^{(L)}$ with both random and (non-periodically time-varying) QC liftings, where the resulting designed SC-LDPC codes have girths of six and eight, respectively. For comparison, we also consider undesigned SC-LDPC codes with randomly chosen base matrices $\mathbf{B}_{\text{SC}}^{(L)}$ and random liftings, which typically have a girth of only four for the selected lengths. The simulations were performed over the additive white Gaussian noise (AWGN) channel using BPSK modulation. (Note that the BP decoding threshold analysis of the designed and undesigned code ensembles in Section IV was conducted over the BEC, since this is a quick and convenient platform for testing and refining our design approach. Its primary conclusions also hold for the AWGN channel, which is a more realistic channel model for simulating the performance of practical communication systems.) *Sliding window* (SW) decoding [4], [22] was used, where a window covers W consecutive block protographs in the coupled graph and the *window size* W (in blocks) satisfies $\omega+1 \leq W \leq L$. Decoding was performed based on the partial Tanner graph framed by the window, where MWn_c check nodes and MWn_v variable nodes are included in a decoding window. In each window position, a block of Mn_v *target symbols*, corresponding to the first block of Mn_v variable nodes in the window, is decoded, and then the window shifts by one block. Sliding along the diagonal band of $\mathbf{H}_{\text{SC}}^{(L)}$, SW decoding estimates codeword symbols block-by-block, resulting in a decoding latency of only W blocks. The maximum number of iterations per window position was 100, and the soft bit-error-rate (BER) stopping rule [29] was employed, with a threshold BER of 10^{-6}. Standard flooding schedule (FS) decoding across the entire terminated graph was also performed for comparison. For FS decoding, a maximum number of 1000 iterations was allowed, and a stopping rule based on the parity-check matrix $\mathbf{H}_{\text{SC}}^{(L)}$ was employed.

Fig. 8 shows the simulated performance of designed and undesigned SC-LDPC codes with $(n_c, n_v) = (3, 6)$ and $\omega = 3$. The undesigned base matrix $\mathbf{B}_{\text{SC}}^{(L)}$ was randomly chosen, with the constraints that $\mathbf{B}_0$ and $\mathbf{B}_3$ have a minimum row weight of two and (2) must be satisfied. The component matrices of the $g = 6$ designed $\mathbf{B}_{\text{SC}}^{(L)}$ were

$$\mathbf{B}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

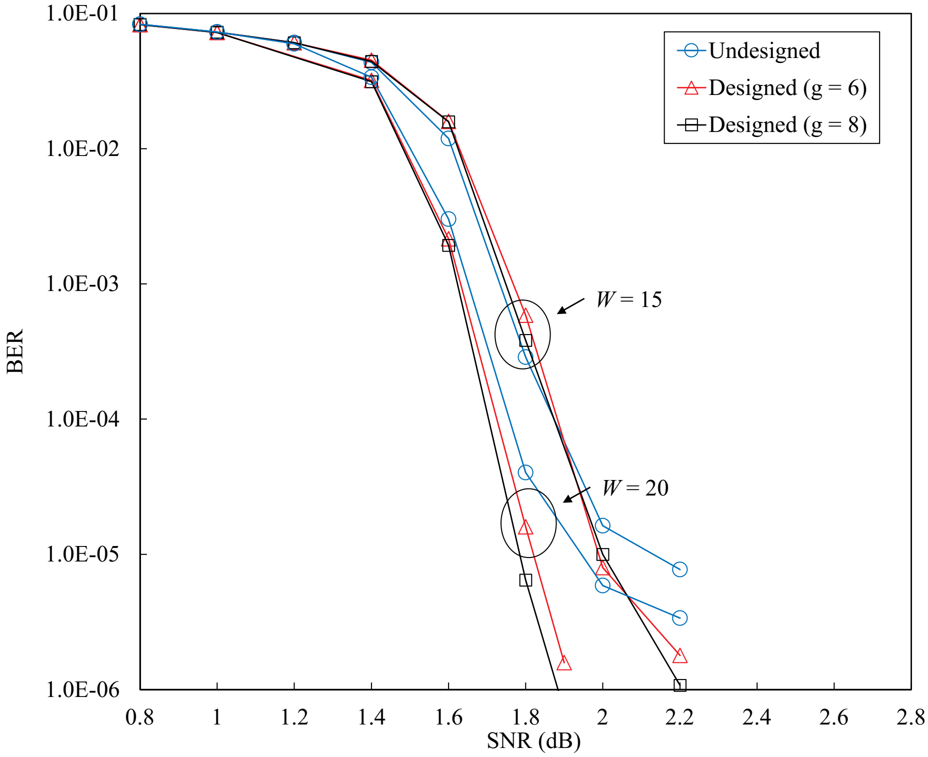


Fig. 9. Performance of designed and undesigned SC-LDPC codes with $(n_c, n_v) = (3, 8)$.

$$\mathbf{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$

and the $g = 8$ designed $\mathbf{H}_{\text{SC}}^{(L)}$ was obtained from $\mathbf{B}_{\text{SC}}^{(L)}$ by doing a (non-periodic) QC lifting using the Fossorier condition. In all cases, $L = 100$, $M = 100$, and $R_{\text{SC}}^{(L)} = 0.485$. The results show that the designed codes substantially outperform the undesigned codes, particularly in the error floor, and that increasing the girth by performing a QC lifting further improves the error floor performance. Also, with FS decoding, the designed and undesigned codes exhibit similar waterfall performance, consistent with the BP decoding threshold analysis of Section III.D.

Fig. 9 shows the performance of designed and undesigned SC-LDPC codes with $(n_c, n_v) = (3, 8)$ and $\omega = 4$. For the undesigned code, the 3×8 binary component matrices were chosen such that $\mathbf{B}_0$ and $\mathbf{B}_4$ have a minimum row weight of two and the edge-spreading condition (see (2)) must be satisfied. L and M were again chosen to be 100, so in this case $R_{\text{SC}}^{(L)} = 0.61$. The results again show the designed codes outperform the undesigned codes, most noticeably in the error floor. We also again see that, since $g = 8$, a QC lifting further improves the error floor performance of the designed codes compared to a random lifting. Finally we note that, with SW decoding, larger window sizes W result in better performance, consistent with the tradeoff between W (latency) and performance reported in [32].

Finally, in order to demonstrate the advantage of our designed QC-SC-LDPC codes, Fig. 10 compares their performance to the SC-LDPC code designed using the optimal overlap (OO) partitioning and circulant power optimizer (CPO) approach of [12], where FS decoding was used in both cases. Both codes were designed from a block protograph with $(n_c, n_v) = (3, 7)$, using the same lifting factor $M = 7$ and coupling length $L = 60$. They both have the same codeword length of 2940 bits and code rate of 0.56. However, the coupling width of our designed code is 3, whereas the OO-CPO code has coupling width only 1. This means that our code enjoys a larger constraint length than the OO-CPO code, viz., 196 vs. 98. It can be seen that our designed code outperforms the OO-CPO code by 1.3dB at a BER of 10^{-8}, which is mostly due to the fact that we have a larger constraint length and enjoy a larger guaranteed girth, eight vs. six for the OO-CPO code. Note that, if we further lower the lifting factor from seven to six for our designed code, its constraint length will be reduced to 168, and we see that it still maintains a substantial performance advantage over the OO-CPO code. However, if we further lower the lifting factor to match the constraint length of the OO-CPO code, we can no longer achieve girth eight in the lifted graph.

VII. Conclusions

In this paper, we introduced a two-stage design procedure for constructing spatially coupled protographs with girth at least 6. The first stage produces an initial set of component matrices that satisfy the edge-spreading condition such that the coupled protograph contains a small number of 4-cycles; while the second stage employs a systematic approach to eliminate the remaining 4-cycles, thus guaranteeing a girth of at least 6. This was followed by performing a QC (circulant-based) lifting of the coupled protograph that satisfies the Fossorier condition in order to obtain girth 8. Both non-periodic and periodic liftings were proposed, and simulations were used to demonstrate that the new designed codes exhibit substantial error floor improvement compared to random constructions. Compared to previous approaches that focused on optimization techniques and time-invariant constructions, the two-stage design approach allows us to consider larger

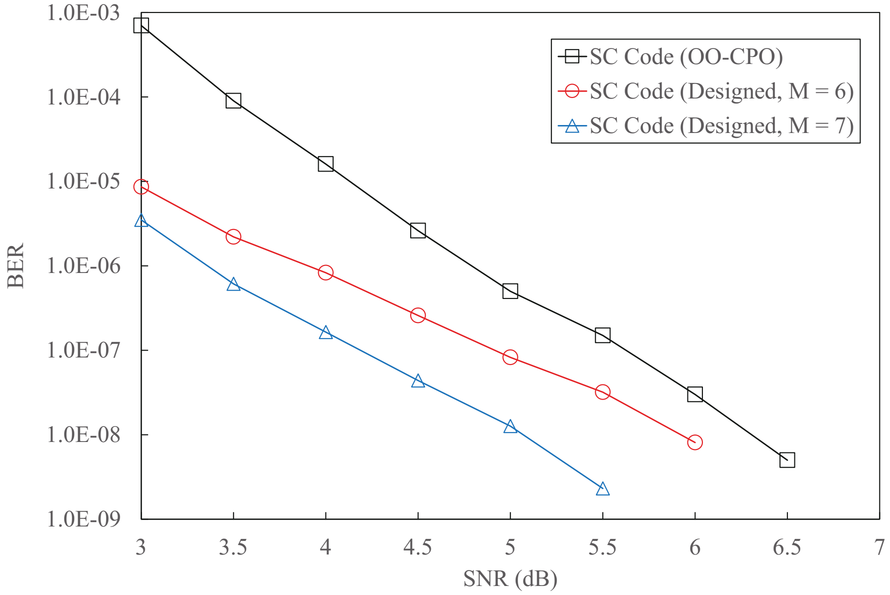


Fig. 10. Performance comparison of our designed codes with the OO-CPO SC-LDPC code with $(n_c, n_v) = (3, 7)$.

coupling widths as well as time-varying code constructions. Finally, empirical results on the minimum coupling width ω needed to ensure girth $g = 6$ for (n_c, n_v)-regular coupled protographs was presented.

References

[1] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” Jet Propulsion Laboratory, Pasadena, CA, INP Progress Report 42-154, Aug. 2003.

[2] D. Divsalar, S. Dolinar. C. Jones, and K. Andrews, “Capacity-approaching protograph codes,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876-888, Aug. 2009.

[3] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., “Spatially coupled LDPC codes constructed from protographs,” *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866-4889, Sept. 2015.

[4] M. Lentmaier, A. Sridharan, D. J. Costello, Jr. and K. Sh. Zigangirov, “Iterative decoding threshold analysis for LDPC convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274-5289, Oct. 2010.

[5] S. Kudekar, T. J. Richardson, and R. L. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC, *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803C834, Feb. 2011.

[6] S. Kudekar, T. J. Richardson, and R. L. Urbanke, “Spatially coupled ensembles universally achieve capacity under belief propagation, *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761C7813, Dec. 2013.

[7] J. Li, S. Lin, K. Abdel-Ghaffar, W. Ryan, and D. J. Costello, Jr., *LDPC Code Designs, Constructions and Unification*, Cambridge Press, 2016.

[8] M. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788-1793, Aug. 2004.

[9] K. Liu, M. El-Khamy, and J. Lee, "Finite-length algebraic spatially coupled quasi-cyclic LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 329-344, Feb. 2016.

[10] M. Zhang, Z. Wang, Q. Huang, and S. Wang, "Time-invariant quasi-cyclic spatially coupled LDPC codes based on packings," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4936-4945, Dec. 2016.

[11] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "A finite-length construction of high performance spatially-coupled codes via optimized partitioning and lifting," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 3-16, Jan. 2019.

[12] D. G. M. Mitchell, L. Dolecek, and D. J. Costello, Jr., "Breaking absorbing sets in array-based spatially coupled LDPC codes, *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, U.S.A., Jun. 2014, pp. 886-890.

[13] B. Amiri and A. Reisizadehmobarakeh and H. Esfahanizadeh and J. Kliewer and L. Dolecek, "Optimized design of finite-length separable circulant-based spatially-coupled codes: an absorbing set-based analysis," *IEEE Trans. Commun.,* vol. 64, no. 10, pp. 4029–4043, Oct. 2016.

[14] A. Beemer, S. Habib and C. Kelley, and J. Kliewer, "A generalized algebraic approach to optimizing SC-LDPC codes," *Proc. 55th Allerton Conf. Commun., Contr. Comput.,* pp. 672–679, Oct. 2017.

[15] D. G. M. Mitchell and E. Rosnes, "Edge spreading design of high rate array-based SC-LDPC codes, *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, June 2017, pp. 2950-2954.

[16] M. Battaglioni, A. Tasdighi, G. Cancellieri, F. Chiaraluce, and M. Baldi, "Design and analysis of time-invariant SC-LDPC convolutional codes with small constraint length," *IEEE Trans. Commun.,* vol. 33, no. 3, pp. 918-931, Mar. 2018.

[17] M. Battaglioni, A. Tasdighi, M. Baldi, M. Tadayon, F. Chiaraluce, "Compact QC-LDPC block and SC-LDPC convolutional codes for low-latency communications," *Proc. IEEE Int. Symp. Pers., Indoor and Mob. Radio Commun. (PIMRC),* Bologna, Italy, Sept. 2018.

[18] M. Tadayon, A. Tasdighi, M. Battaglioni, M. Baldi, F. Chiaraluce, "Efficient search of compact QC-LDPC and SC-LDPC convolutional codes with large girth," *IEEE Comm. Lett.,* vol. 22, no. 6, pp. 1156V1159, Jun. 2018.

[19] M. Sadeghi, "Optimal search for girth-8 quasi cyclic and spatially coupled multiple-edge LDPC codes," *IEEE Comm. Lett.,* vol. 23, no. 9, pp. 1466-1469, Sept. 2019.

[20] S. Naseri, A. Banihashemi, "Spatially coupled LDPC codes with small constraint length and low error floor," *IEEE Comm. Lett.,* vol. 24, no. 2, pp. 254-258, Feb. 2020.

[21] J. Cho and L. Schmalen, "Construction of protographs for large-girth structured LDPC convolutional codes," *Proc. IEEE Int. Conf. Commun. (ICC)*, London, UK, Jun. 2015.

[22] A. Iyengar, M. Papaleo, P. Siegel, J. Wolf, A. Vanelli-Coralli, and G. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol.58, no. 4, pp. 2303-2320, Apr. 2012.

[23] V. Chandrasetty, S. Johnson, G. Lechner, "Memory efficient quasi-cyclic spatially coupled low-density parity-check and repeat accumulate codes," *IET Commun.,* vol. 8, no. 17, pp. 3179-3188, 2014.

[24] D. G. M. Mitchell, A. E. Pusane, M. Lentmaier, and D. J. Costello, Jr., "On the block error rate performance of spatially coupled LDPC codes for streaming applications," *Proc. IEEE Inf. Theory Workshop (ITW),* Cambridge, UK, Sept. 2016.

[25] Y. Wang, S. Draper, and J. Yedida, "Hierarchical and high-girth QC LDPC codes," *IEEE Trans. Inf. Theory,* vol. 59, no. 7, pp. 4553–4583, Jul. 2013.

[26] D. G. M. Mitchell, R. Smarandache, and D. J. Costello, Jr., "Quasi-cyclic LDPC codes based on pre-lifted protographs," *IEEE Trans. Inf. Theory,* vol. 60, no. 10, pp. 5856–5874, Oct. 2014.

[27] A. Pusane, R. Smarandache, P. Vontobel, and D. J. Costello, Jr., “Deriving good LDPC convolutional codes from LDPC block codes,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 835-857, Feb. 2011.

[28] L. Chen, S. Mo, D. J. Costello, Jr., D. G. M. Mitchell, and R. Smarandache, “A protograph-based design of quasi-cyclic spatially coupled LDPC codes,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1683-1687.

[29] N. ul Hassan, M. Lentmaier and G. Fettweis, “Comparision of LDPC block and LDPC convolutional codes based on their decoding latency,” in *Proc. Int. Symp. Turbo Codes and Iter. Inf. (ISTC)*, Gothenburg, Sweden, Aug. 2012.

[30] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., “Spatially coupled LDPC codes constructed from protographs,” *IEEE Trans. Inf. Theory*, vol.61, no. 9, pp. 4866-4889, Jul. 2015.

[31] A. Jimenez Felstrom and K. S. Zigangirov, “Time-varying periodic convolutional codes with low-density parity-check matrix,” *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181-2191, Sept. 1999.

[32] K. Huang, D. G. M. Mitchell, L. Wei, X. Ma, and D. J. Costello, Jr., “Performance comparison of LDPC block and spatially coupled codes over GF(q),” *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 592-604, Mar. 2015.

[33] M. Lentmaier and G. Fettweis, “On the thresholds of generalized LDPC convolutional codes based on protographs,” *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, U.S.A., Jun. 2010, pp. 709-713.

[34] S. Kim, J. No, H. Chung, and D. Shin, “Quasi-cyclic low-density parity-check codes with girth larger than 12,” *IEEE Trans. Inf. Theory,* vol. 53, no. 8, pp. 2885-2891, Aug. 2007.

[35] I. Bocharova, F. Hug, R. Johannesson, B. Kudryashov, and R. Satyukov, “Searching for voltage graph-based LDPC tailbiting codes with large girth,” *IEEE Trans. Inf. Theory,* vol. 58, no. 4, pp. 2265-2279, Apr. 2012.

[36] H. Park, S. Hong, J. No, D. Shin, “Design of multiple-edge protographs for QC LDPC codes avoiding short inevitable cycles,” *IEEE Trans. Inf. Theory,* vol. 59, no. 7, pp. 4598-4614, Jul. 2013.

[37] A. Hareedy and R. Calderbank, “A new family of constrained codes with applications in data storage,” *Proc. IEEE Inf. Theory Workshop,* Visby, Sweden, Aug. 2019.