A Protograph-Based Design of Quasi-Cyclic Spatially Coupled LDPC Codes

Li Chen ‡, Shi yuan Mo ‡, Daniel J. Costello, Jr. ‡, David G. M. Mitchell §, Roxana Smarandache ‡
‡ School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
‡ Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA
§ Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, USA
Emails: chenli55@mail.sysu.edu.cn, moshiy@mail2.sysu.edu.cn, {costello.2, rsmarand}@nd.edu, dgmm@nmsu.edu

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. This paper introduces a systematic design of SC-LDPC codes. The idea depends on the fact that the girth of a lifted graph is lower bounded by the girth of its base graph. Hence, starting from a block protograph with good asymptotic properties, we design the edge spreading in two stages to maximize the girth and minimize the number of short cycles in the SC protograph. Then, in the lifting phase, we use circulants to further improve the girth by applying the Fossorier condition. Besides improving our ability to find protographs with large girth and a small number of short cycles, the two-stage approach makes it easier to apply the Fossorier condition, since the SC protograph has already been designed to achieve these objectives.

The edge-spreading procedure can be interpreted as decomposing a base matrix \( B \) (corresponding to a block protograph) into a number of submatrices, which are used to form an SC base matrix \( B_{SC} \). In our approach, we identify several component blocks of \( B_{SC} \) that guide the design of the submatrices, leading to an SC protograph with good girth properties. By further performing a graph-lifting of \( B_{SC} \) using the Fossorier condition to generate an SC parity-check matrix \( H_{SC} \), we show that it is possible to achieve a girth of at least eight. Simulation results show that substantial performance gains, particularly in the error floor, are achieved using the two-stage design approach compared to random designs.

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 properties of small protographs sheds light on constructing code ensembles with good asymptotic properties by applying a graph-lifting procedure [2]. 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. Another important aspect of code design is to maximize the girth of the Tanner graph. For protograph-based constructions of QC-LDPC codes, this can be accomplished by applying the Fossorier condition [3] to the graph-lifting. The protograph-based method has also been used to construct good spatially coupled LDPC (SC-LDPC) codes [4]. An edge-spreading procedure is first applied to a chain of block protographs in order to introduce memory. This results in a two-stage code design procedure, first the edge spreading and then the graph-lifting, to achieve good asymptotic properties and a large girth, respectively.

Several constructions for \( Q \)-SC-LDPC codes have been proposed in [5]–[8]. In this paper, we take a new protograph-based systematic design approach to insure large girth for QC-SC-LDPC codes. The idea depends on the fact that the girth of a lifted graph is lower bounded by the girth of its base graph. Hence, starting from a block protograph with good asymptotic properties, we design the edge spreading in two
so the coupled protograph maintains the check node and variable node degrees of the original protograph. If the original 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-LDPC code has finite length. It can be obtained from a finite number $L$ of coupled block protographs, where $L$ is called the coupling length. The coupled protograph contains $Ln_v$ variable nodes and $(L + \omega)n_c$ check nodes, as shown in Fig. 1(d), and the corresponding SC base matrix is

$$B_{SC}^{(L)} = \begin{bmatrix} B_0 & B_0 & \ldots & B_0 \\ \vdots & \vdots & \ddots & \vdots \\ B_\omega & B_\omega & \ldots & B_0 \\ \vdots & \vdots & \ddots & \vdots \\ B_\omega & B_\omega & \ldots & B_0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}. \tag{3}$$

Note that the first and last $\omega n_c$ check nodes have reduced degrees, which is an important feature to realize the excellent thresholds of SC-LDPC codes [4], [10].

Let $P_a$ denote an $a \times a$ binary permutation matrix and $I_a$ denote the $a \times a$ identity matrix. Furthermore, let $I_a^{(\theta)}$ denote the shifted identity matrix with each row of $I_a$ cyclically shifted to the left by $\theta$ positions. Finally, let $\Xi_{a \times b}$ denote an $a \times b$ binary matrix with a minimum row weight of one and with as small a maximum column weight as possible.

The parity-check matrix $H_{SC}^{(L)}$ of an SC-LDPC code can be obtained by an $M$-fold matrix expansion from $B_{SC}^{(L)}$ that corresponds to an $M$-fold graph-lifting of the coupled protograph [9]. 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. For $B_{SC}^{(L)} = [B_{SC}^{(L)}(r, s)]_{(L+\omega)n_c \times L n_v}$, $H_{SC}^{(L)}$ is generated by replacing each nonzero entry in $B_{SC}^{(L)}$ by a sum of $B_{SC}^{(L)}(r, s)$ permutation matrices $P_M$ and replacing each zero entry by the $M \times M$ all zero matrix. The constraint length and design rate of the code are $M n_v (\omega + 1)$ and $R_{SC}^{(L)} = 1 - \frac{(L+\omega)n_c}{Ln_v}$, respectively, where $\lim_{L \to \infty} R_{SC}^{(L)} = 1 - \frac{n_c}{n_v}$. $H_{SC}^{(L)}$ defines the Tanner graph of a particular SC-LDPC code.

This lifting approach leads to the following characterization for the girth (denoted $g$) of the Tanner graph.

**Lemma 1.** The girth of the Tanner graph of $H_{SC}^{(L)}$ is lower bounded by the girth of the protograph of $B_{SC}^{(L)}$.

This lemma motivates the design in Section III.

### III. Design of QC-SC-LDPC Codes

Based on Lemma 1, the proposed approach aims to first eliminate (or reduce the number of) 4-cycles in $B_{SC}^{(L)}$. Then, using a systematic lifting, we try to construct $H_{SC}^{(L)}$ with $g \geq 8$. Due to the diagonal nature of $B_{SC}^{(L)}$ (see (3)), a careful examination of its structure is needed in the design.

#### A. Preliminaries

We consider the common case when the base matrix is all one, i.e., $B = 1_{n_c \times n_c}$, e.g., $B = 1_{3 \times 6}$. A 4-cycle in a coupled protograph or Tanner graph corresponds to four nonzero entries that form a rectangular array in $B_{SC}^{(L)}$ or $H_{SC}^{(L)}$, respectively. This leads to the following lemma.

**Lemma 2.** In $B_{SC}^{(L)}$, 4-cycles may exist in the following patterns: 1) one of its submatrices; 2) two submatrices that appear in the same row or the same column of $B_{SC}^{(L)}$; 3) four submatrices that appear in a rectangular array of $B_{SC}^{(L)}$.

Lemma 2 defines the various patterns in which 4-cycles can appear in $B_{SC}^{(L)}$. We now decompose $B_{SC}^{(L)}$ as follows:

- The **representative block** $B_R$ is defined as

$$B_R \triangleq \begin{bmatrix} B_\omega & B_\omega & \ldots & B_0 \\ \vdots & \vdots & \ddots & \vdots \\ B_\omega & B_\omega & \ldots & B_0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}, \tag{4}$$

with size $(\omega + 1) n_c \times (\omega + 1) n_v$. Any combination of one, two, or four submatrices in $B_{SC}^{(L)}$ described by conditions 1), 2), or 3) in Lemma 2, respectively, will also appear in $B_R$.

Therefore, if $B_R$ does not contain 4-cycles, neither will $B_{SC}^{(L)}$.

The following two definitions are based on $B_R$.

- A **constituent block** $B_C$ is defined as

$$B_C \triangleq \begin{bmatrix} B_\omega & B_\omega & \ldots & B_0 \\ \vdots & \vdots & \ddots & \vdots \\ B_\omega & B_\omega & \ldots & B_0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}, \tag{5}$$

where $\omega = \alpha + \beta - 2$ and $\alpha, \beta > 1$, with size $\alpha n_c \times \beta n_v$. $B_C$ is obtained by forming a rectangular matrix from $B_R$ that contains $B_0$ in the upper right corner and one of the $B_\omega$ submatrices along the diagonal (excluding the upper left and lower right corners) in the lower left corner. Hence, there are $\omega - 1$ choices for $B_C$, and when $\omega = 2$, $B_C$ is unique. Note that each possible constituent block includes every submatrix. We define the weight $w(\omega)$ of a submatrix $B_i$ as the number of times $B_i$ is included in a particular $B_C$.

- **Excluded patterns** $B_E^{(j)}$ are defined as
where \( a_j, b_j, c_j, d_j \in \{0, 1, \ldots, \omega\} \). Block \( B_{E}^{(1)} \) (resp. \( B_{E}^{(2)} \)) is the \( n_c \times 2n_v \) (resp. \( 2n_v \times n_c \)) single row (resp. column) “pattern” (submatrix) that appears in \( B_R \) but not \( B_C \). Similarly, \( B_{E}^{(j)}, j = 3, 4, \ldots, n_E \), are the \( 2n_v \times 2n_v \) rectangular patterns appearing in \( B_R \) but not \( B_C \). The number of excluded patterns \( n_E \) depends on \( \omega \) and the chosen \( B_C \), while the particular set of excluded patterns depends on the chosen \( B_C \). Note that when \( \omega = 2 \), there are only two excluded patterns \( B_{E}^{(1)} \) and \( B_{E}^{(2)} \), since all \( 2n_c \times 2n_v \) rectangular patterns appear in \( B_C \). The following example illustrates the above definitions.

**Example 1.** When \( \omega = 3 \), we have

\[
B_R = \begin{bmatrix}
B_3 & B_2 & B_1 & B_0 \\
B_3 & B_2 & B_1 \\
B_3 & B_2 \\
B_3
\end{bmatrix},
\]

\( B_C \) can be defined as

\[
B_C = \begin{bmatrix}
B_2 & B_1 & B_0 \\
B_3 & B_2 & B_1 \\
B_3 & B_2 \\
B_3
\end{bmatrix},
\]

where \( wt(B_0) = wt(B_1) = 1 \) and \( wt(B_2) = 2 \).

The excluded patterns are

\[
B_{E}^{(1)} = [B_3 \ 0], \quad B_{E}^{(2)} = [B_0], \quad B_{E}^{(3)} = [B_1 \ 0].
\]

Note that \( B_C \) can also be of size \( 3n_c \times 2n_v \), which also results in three excluded patterns.

The above definitions lead to the following theorem.

**Theorem 3.** The coupled protograph of \( B_{SC}^{(L)} \) does not have any 4-cycles if the chosen \( B_C \) and associated \( B_E^{(j)}, j = 1, 2, \ldots, n_E \), do not contain any 4-cycles.

**Proof:** The result follows directly from Lemma 2. For condition 1), \( B_C \) includes all possible submatrices. For conditions 2) and 3), \( B_C \) and \( B_E^{(j)} \) contain all possible patterns of submatrices that can result in 4-cycles in \( B_{SC}^{(L)} \).

\section*{B. Design of \( B_{SC}^{(L)} \) - Stage 1}

**Design Rule 1 Initialize the Submatrices (Stage 1)**

1. Initialize \( B_{o} = [I_{n_c} \ \Xi_{n_c \times (n_v - n_c)}] \).

2. Initialize \( B_{o} \) such that there is no 4-cycle in \( B_{E}^{(2)} \), the minimum row weight of \( B_{o} \) is two, and the maximum column weight of \( B_{o} \) is one.

3. Initialize \( B_{1}, B_{2}, \ldots, B_{v-1} \) such that
   
   1) \( B(r,s) = \sum_{j=0}^{
   \end{array}
   \}
   \] i.e., (2) is satisfied;

2) There is no 4-cycle in any of these submatrices or in the excluded patterns \( B_{E}^{(j)}(j = 3, 4, \ldots, n_E) \).

Based on Theorem 3, we aim to design the submatrices such that neither the chosen \( B_C \) nor its associated \( B_E^{(j)} \) contain any 4-cycles. The proposed design includes two stages: Stage 1 initializes the submatrices based on \( B_E^{(j)} \); Stage 2 modifies the submatrices based on \( B_C \).

Given a base matrix \( B = I_{n_c \times n_c} \) and coupling width \( \omega \), the Stage 1 design is given in Design Rule 1. It insures the submatrices and the excluded patterns do not contain any 4-cycles. Step 1.1 insures the minimum row weight of \( B_0 \) is two and its maximum column weight is one if \( n_v - n_c \geq n_c \). If \( n_v - n_c \leq n_c \), some columns of \( B_0 \) will have weight greater than one. The design of \( B_{o} \) in Step 1.2 also insures that \( B_E^{(1)} \) does not contain any 4-cycles. Requiring \( B_0 \) and \( B_\omega \) to have a minimum row weight of two is due to the fact that they are the only submatrices in the top and bottom rows of \( B_{SC}^{(L)} \), respectively, and a row weight of at least two is needed to assist the startup and termination of sliding window (SW) decoding [10], [11]. Also, if possible, restricting the maximum column weight of \( B_{o} \) to one simplifies the design of Step 1.3 for the cases where \( B_\omega \) is included in \( B_E^{(j)}(j = 3, 4, \ldots, n_E) \). The remaining submatrices \( B_1, B_2, \ldots, B_{\omega-1} \) are then initialized based on \( B_{E}^{(j)}(j = 3, 4, \ldots, n_E) \), \( B_0 \), and \( B_{\omega} \). The following example illustrates the procedure.

**Example 2.** Given \( B = I_{3 \times 6} \) and \( \omega = 3 \), \( B_R, B_C, B_E^{(1)}, B_E^{(2)}, \) and \( B_E^{(3)} \) are given in Example 1. We can employ Design Rule 1 to initialize the submatrices. First, we can let

\[
B_0 = [I_3 \ \Xi_{3 \times 3}] = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}.
\]

Placing \( B_0 \) into \( B_{E}^{(2)} \), we can initialize

\[
B_3 = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix},
\]

so that neither \( B_E^{(1)} \) nor \( B_E^{(2)} \) contain any 4-cycles. In order to initialize \( B_1 \) and \( B_2 \), we place both \( B_0 \) and \( B_3 \) into \( B_{E}^{(3)} \). To satisfy (2), we must place zeros into certain positions in \( B_1 \) and \( B_2 \), leaving 12 unspecified positions in \( B_{E}^{(3)} \). We can then initialize \( B_1 \) and \( B_2 \) as

\[
B_1 = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}, \quad B_2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},
\]

where this choice insures \( B_{E}^{(3)} \) does not contain any 4-cycles.

Finally, we place the initialized submatrices into \( B_C \) to check if it contains any 4-cycles. If not, the coupled protograph of \( B_{SC}^{(L)} \) 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 \( B_C \).

\section*{C. Design of \( B_{SC}^{(L)} \) - Stage 2}

This stage modifies the initialized submatrices to remove the remaining 4-cycles in \( B_C \). In order to distinguish between an entry in \( B_C \) and one in \( B_1 \), we use \( B_C(x, y) \) to denote its row-\( x \) column-\( y \) entry, for \( 1 \leq x \leq \alpha n_c \) and \( 1 \leq y \leq \beta n_v \). The Stage 2 design is given in Design Rule 2.
Design Rule 2 Modify the submatrices (Stage 2)

2.1: Identify a 4-cycle in $B_C$ with entries

$$B_C(x_1, y_1) = 1, \ B_C(x_1, y_2) = 1,$$

$$B_C(x_2, y_1) = 1, \ B_C(x_2, y_2) = 1.$$ 

2.2: Say the four entries belong to submatrices $B_{i_1}$, $B_{i_2}$, $B_{i_3}$, and $B_{i_4}$, where $(i_1, i_2, i_3, i_4) \in \{0, 1, \ldots, \omega\}$. Denote these entries as

$$B_{i_1}(r_1, s_1) = 1, \ B_{i_2}(r_1, s_2) = 1,$$

$$B_{i_3}(r_2, s_1) = 1, \ B_{i_4}(r_2, s_2) = 1.$$ 

2.3: Among these four entries, identify those that have not previously been flipped. Pick one that belongs to a submatrix of highest weight and denote it $B_{i'}(r', s')$, where $i' \in \{i_1, i_2, i_3, i_4\}$ and $(r', s') \in \{(r_1, s_1), (r_1, s_2), (r_2, s_1), (r_2, s_2)\}$. Further, denote the appearance of $B_{i'}(r', s')$ in $B_C$ as $B_C(x', y')$, where $(x', y') \in \{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}$. Flip down both $B_{i'}(r', s')$ and $B_C(x', y')$ such that

$$B_{i'}(r', s') : 1 \to 0,$$

$$B_C(x', y') : 1 \to 0.$$ 

Also flip down all other entries in $B_{i'}^{(j)} (j = 1, 2, \ldots, n_E)$ and $B_C$ that correspond to entry $B_{i'}(r', s')$.

2.4: Flip up $B_{i'}(r', s')$ in one of the other submatrices $B_{i'}^{(j)}$ where $i = 0, 1, \ldots, \omega$ and $i' \neq i'$, conditioned on

1. The entry has not been previously flipped;
2. The flipping does not create new 4-cycles in $B_i$ or in any $B_{i'}^{(j)}$ that includes $B_i$;
3. The number of 4-cycles contained in $B_C$ does not increase after a complete flipping (down and up) process.

Also flip up all other entries in $B_{i'}^{(j)} (j = 1, 2, \ldots, n_E)$ and $B_C$ that correspond to entry $B_{i'}(r', s')$.

2.5: If the flipping of Step 2.4 succeeds, go to Step 2.6; else, reflip $B_C(x', y')$ and $B_{i'}(r', s')$ to their original values, i.e.,

$$B_C(x', y') : 0 \to 1,$$

$$B_{i'}(r', s') : 0 \to 1.$$ 

Also reflip all other entries in $B_{i'}^{(j)} (j = 1, 2, \ldots, n_E)$ and $B_C$ that correspond to entry $B_{i'}(r', s')$, 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 bits to flip.

In Step 2.2, $i_1$, $i_2$, $i_3$ and $i_4$ do not need to be distinct. In Step 2.3, we prioritize the “flipping down” of a nonzero entry of an identified submatrix that has the highest weight in $B_C$. In doing so, we remove the most nonzero entries in $B_C$, so that more 4-cycles are likely to be removed. However, it is possible that none of the four entries allows a complete flipping, in which case the remaining 4-cycle is labelled dormant. It will be targeted again if some other complete flipping occurs and this dormant 4-cycle still exists. But for small coupling widths, we remove the most nonzero entries in $B_C$ be targeted again if some other complete flipping occurs and this remaining 4-cycle is labelled dormant. It will be targeted again if some other complete flipping occurs and this dormant 4-cycle still exists.
results show that the designed codes substantially outperform the undesigned code, particularly in the error floor, and that performing a QC lifting further improves the performance. Fig. 4 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 \(3 \times 8\) binary submatrices were chosen such that \(B_0\) and \(B_1\) have a minimum row weight of two, while \(B_2, B_3, B_4\) were chosen randomly to satisfy (2), and in this case \(R_{SC}^{(L)} = 0.575\). Again, the results show the designed codes outperform the undesigned code. When \(W = 10\), the designed codes outperform only in the error floor. This is because the window covers only two constraint lengths, which is too small for SW decoding to be effective.

V. Conclusions

This paper presented a protograph-based systematic design of QC-SC-LDPC codes, resulting in a girth \(g \geq 8\) for the lifted codes. The SC base matrix is decomposed into a set of blocks to guide the design of edge spreading. Excluded patterns were utilized to initialize the submatrices, while the constituent block was utilized to modify them. Unless the coupling width \(\omega\) is too small, an SC base matrix with \(g \geq 6\) can be obtained. For large enough lifting factor \(M\), and a systematic QC lifting, an SC parity-check matrix with \(g \geq 8\) can be obtained. Simulation results show our design leads to significantly improved performance, particularly in the error floor.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China (NSFC) under grants 61372079 and 61671486.

References