Optimized Design of Finite-Length Separable Circulant-Based Spatially-Coupled Codes: An Absorbing Set-Based Analysis

Behzad Amiri, *Student Member, IEEE*, Amirhossein Reisizadehmobarakeh, *Student Member, IEEE*, Homa Esfahanizadeh, *Student Member, IEEE*, Jörg Kliewer, *Senior Member, IEEE*, and Lara Dolecek, *Senior Member, IEEE*

***Abstract*— In this paper, we characterize the finite-length performance of separable circulant-based spatially-coupled (SCB-SC) LDPC codes for transmission over the additive white Gaussian noise channel. For a general class of finite-length graph-based codes, it is known that the existence of small absorbing sets causes a performance degradation in the error floor regime. We first present the mathematical conditions for the existence of absorbing sets in binary SCB-SC codes. This analysis enables us to find the exact number of absorbing sets as a function of the design parameters. In particular, our results show that the choice of the cutting vector affects the number of absorbing sets and, therefore, the error floor performance of the code. For a fixed column weight, we find provably optimal cutting vectors that result in the least number of absorbing sets. Furthermore, we extend our analysis to nonbinary SCB-SC codes, where we show that the choice of the cutting vector is not as critical as in the binary case. We provide an algorithm which provably removes the problematic nonbinary absorbing sets from nonbinary SCB-SC codes by informed selection of edge labels. Our simulation results show the superior error floor performance of our designed binary and nonbinary SCB-SC codes compared with binary unstructured and nonbinary quasi-cyclic SC codes available in the open literature.**

***Index Terms*— Spatially-coupled codes, finite block length, separable circulant-based, array-based codes, absorbing sets, error floor performance.**

I. Introduction

In RECENT years, spatially-coupled low-density parity-check (LDPC) codes have drawn a lot of attention due to their outstanding asymptotic performance. In the asymptotic regime, it is known that long binary spatially-coupled LDPC (SC-LDPC) codes achieve the maximum-likelihood decoding threshold under belief propagation decoding, which is called the *threshold saturation phenomenon* [1]–[8], [31]. This phenomenon was shown in [1] for binary erasure channels and regular code ensemble and was also reported for other channels [2]–[5] and code ensembles [6]–[8]. Asymptotic minimum codeword and trapping set distances of spatially-coupled codes have been studied as well, see [9]. It was shown that the asymptotic minimum distance of spatially-coupled codes grows linearly with the coupling length of these codes. As a result of these favorable properties, the analysis and design of SC-LDPC codes has been actively explored, see, e.g., [7], [9]–[12]. More recently, it was shown that non-binary SC-LDPC codes also achieve maximum-likelihood decoding threshold and have superior performance compared to their binary counterparts [13]–[15]. Further, in [16] a survey about the theory and practice of SC-LDPC codes was presented.

While the majority of recent results has been devoted to establishing the asymptotic performance of spatially-coupled codes, much less has been done for the finite-length analysis and design of these codes, with the exception of the following representative results. The finite-length performance and various constructions of spatially-coupled codes were reported in [17]–[20]. Also, various protograph-based methods for deriving families of spatially-coupled codes with good performance were presented in [17] and [47], and a class of spatially-coupled codes based on regular repeat-accumulate codes was discussed in [18] and [34]. In [19], the finite-length performance analysis for codes constructed based on a chain of spatially-coupled codes was presented. More recently, the authors in [20] provided an analytical framework for predicting the error probability of finite-length spatially-coupled ensembles based on a proposed scaling law. Several works, including [48] and [49], considered a subclass of spatially-coupled codes which are constructed based on quasi-cyclic LDPC codes. In [48], the authors showed that the minimum distance of quasi-cyclic LDPC convolutional codes can be improved by a careful choice of the circulants in the Tanner graph of the code. The construction of a class of quasi-cyclic LDPC convolutional codes and its finite-length performance was investigated in [49]. Finally, the finite-length analysis and the design of non-binary spatially-coupled codes was recently studied in [14] and [21].

Manuscript received November 27, 2015; revised May 11, 2016 and July 16, 2016; accepted July 23, 2016. Date of publication August 2, 2016; date of current version October 14, 2016. This paper was presented in part at 2015 IEEE International Symposium on Information Theory [30]. This work was supported in part by ASTC-IDEMA and NSF grants CCF-1440001 and CCF-1161798. The associate editor coordinating the review of this paper and approving it for publication was A. Thangaraj.

B. Amiri is with Pure Storage Inc., Mountain View, CA 94041 USA (e-mail: amiri@ucla.edu).

A. Reisizadehmobarakeh, H. Esfahanizadeh, and L. Dolecek are with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: reisizadeh@ucla.edu; hesfahanizadeh@ucla.edu; dolecek@ee.ucla.edu).

J. Kliewer is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: jkliewer@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2016.2597287

It is known that under message passing decoding, certain non-codewords result in failures in the iterative decoding of graph-based codes. *Absorbing sets (ASs)* were introduced in [22] as particular combinatorial substructures of the Tanner graph which are responsible for the performance degradation of (block) LDPC codes in the error floor region. An analysis of ASs and the design of separable circulant-based (SCB) LDPC codes and their subclass, array-based (AB) LDPC codes [35], with a reduced number of ASs was presented in [22] and [23].

SCB-LDPC codes include a wide variety of regular LDPC codes including array-based LDPC codes as well as many common quasi-cyclic codes [23], [45], [46]. These quasi-cyclic LDPC codes have drawn a lot of interest in both academia and industry due to their high-throughput parallel-layered decoding and excellent performance. In particular, the structure of quasi-cyclic LDPC codes allows multiple processors to perform the decoding in parallel on different parts of the Tanner graph, which results in higher decoding throughput and lower decoding latency [40]–[42]. The quasi-cyclic structure guarantees a certain minimum distance, a minimum girth, and a minimum AS size. Therefore, quasi-cyclic codes typically provide superior performance compared to unstructured LDPC codes [22], [43], [44], [46].

A combination of SC-LDPC and SCB-LDPC structures results in a new family of codes, separable circulant-based spatially-coupled (SCB-SC) codes, which inherit the excellent benefits of SC and SCB codes highlighted above. In addition, SCB-SC codes are suitable for windowed encoding and decoding in a streaming fashion which significantly reduces the latency compared to encoding and decoding of SCB-LDPC block codes. For the case of array-based spatially-coupled (AB-SC) codes, a subclass of SCB-SC codes, the authors demonstrated in [24] that the number of smallest ASs grows linearly with the coupling length. Furthermore, in [24] it was shown that the average number of ASs in AB-SC codes is significantly less than the average number of ASs in their underlying AB-LDPC codes. This observation suggests that AB-SC codes potentially have better error floor performance compared to AB-LDPC codes.

In this work, inspired in part by [24], we generalize the characterization of ASs to binary and non-binary SCB-SC codes for transmission over the additive white Gaussian noise (AWGN) channel. Our contribution in this paper is multifold:

1) We introduce an analytical approach to find the exact number of ASs in SCB-SC codes: the original counting problem is mapped to a problem of finding the number of integer points within an area in 2D space.
2) We find the optimal cutting vector for SCB-SC codes with arbitrary circulant size; optimality is defined in terms of dominant detrimental ASs. Here, dominant ASs are those ASs that occur most frequently at the output of the decoder and 'dominate' the error floor behavior of the code. Analytical and experimental results reveal that the choice of the cutting vector significantly affects the error floor performance of binary SCB-SC codes. Further, the proposed cutting vector optimization approach offers a computational advantage compared to the approach presented in [24], in which for each optimization step the dominant ASs are counted via a hard search on the resulting Tanner graph [25]. In this work we solely consider column-weight 3 since these codes are typically preferred in applications due to lower latency [26], a higher hardware decoder throughput [27, Table 13.1], and smaller required chip area [28] compared to higher column weight codes.[1] Note that the check node degree must increase accordingly with the column weight, leading to a much higher complexity demand.
3) We calculate the average number of ASs in non-binary SCB-SC codes constructed by uninformed (random) assignment of edge labels on top of a binary SCB-SC code. This result reveals that the average number of ASs in non-binary SCB-SC codes is significantly lower than the average number of ASs in binary SCB-SC codes. This explains the superior error floor performance of non-binary SCB-SC codes compared to their binary counterparts.

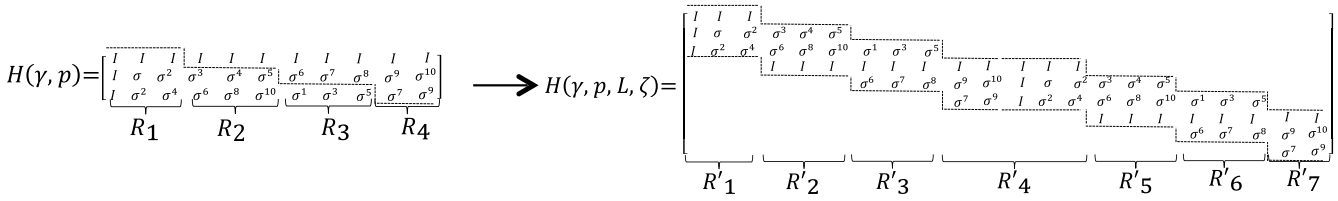
Compared to our preliminary work in [29], this paper extends the discussion on the exact enumeration of ASs to the more general family of SCB-SC codes. Furthermore, by providing the necessary conditions for the existence of non-binary ASs in non-binary SCB-SC codes over $GF(3)$ and $GF(2^m)$, $m > 1$, we also present an analytical approach to provably reduce the number of problematic non-binary ASs in our designed codes.

The rest of this paper is organized as follows. In Section II we briefly review the definition of ASs and the construction of SCB-SC and AB-SC codes. Furthermore, we provide an example which illustrates our motivation for studying ASs of spatially-coupled codes. In Section III we analytically derive the exact enumeration of binary ASs for column-weight-3 SCB-SC codes. Based on these results, in Section IV we find the optimal cutting vector to minimize the number of problematic ASs in binary SCB-SC codes. In Section V we calculate the average number of small ASs for the ensemble of non-binary SCB-SC codes with column-weight 3. Additionally, we present an algorithm to design non-binary SCB-SC codes with improved error floor performance achieved by provable elimination of dominant ASs. Section VI presents our experimental results which show the superior performance of our designed binary and non-binary SCB-SC codes. Section VII delivers the conclusions.

II. Preliminaries and Definitions

In this section, we introduce the construction of SCB-SC codes (and AB-SC codes as a subclass) using the edge spreading procedure [23], [24]. We present the error floor performance comparisons for SCB-LDPC and SCB-SC codes. These results serve to illustrate our motivation for analysis of the SCB-SC codes viz. their AS properties. We also briefly revisit the well-known definitions of binary and non-binary ASs [22], [30].

[1] Note that the enumeration method in this paper can also be applied to higher column-weight codes. In that case, the analysis is more involved and tedious since there are significantly more cases to be considered for the enumeration of ASs when the column-weight is larger.



(a) (b)

Fig. 1. (a) Example of an AB-LDPC code with $p = 11$ and $\gamma = 3$, (b) Example of an AB-SC code with $p = 11$, $\gamma = 3$, $L = 2$, and cutting vector $\boldsymbol{\xi} = [3, 6, 9]$. Here, $r_i = |R_i|$ for $i \in \{1, 2, 3, 4\}$ and $r'_i = |R'_i|$ for $i \in \{1, 2, \cdots, 7\}$. Moreover, $r'_1 = r_1$, $r'_2 = r_2$, $r'_3 = r_3$, $r'_4 = r_4 + r_1$, $r'_5 = r_2$, $r'_6 = r_3$, and $r'_7 = r_4$.

A. Separable Circulant-Based and Array-Based Spatially-Coupled LDPC Codes

In this subsection, we first describe the construction of SCB-LDPC codes which are used as underlying codes in the construction of SCB-SC codes. Let p be a prime number indicating the circulant size and the row weight and γ be the column weight. For given one-to-one functions $f(\cdot)$ and $g(\cdot)$, the $\gamma p \times p^2$ parity-check matrix of the underlying SCB-LDPC code is formed as follows [23]:

$$H(\gamma, p) = \begin{bmatrix} \sigma^{f(0)g(0)} & \sigma^{f(0)g(1)} & \sigma^{f(0)g(2)} & \cdots & \sigma^{f(0)g(p-1)} \\ \sigma^{f(1)g(0)} & \sigma^{f(1)g(1)} & \sigma^{f(1)g(2)} & \cdots & \sigma^{f(1)g(p-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma^{f(\gamma-1)g(0)} & \sigma^{f(\gamma-1)g(1)} & \sigma^{f(\gamma-1)g(2)} & \cdots & \sigma^{f(\gamma-1)g(p-1)} \end{bmatrix}, \tag{1}$$

where σ is a $p \times p$ circulant matrix formed by cyclically shifting all rows of the identity matrix one element to the left. The matrix $H(\gamma, p)$ can be viewed as a 2D array of submatrices where each row (column) of matrices denotes a *row (column) group* $i, 0 \le i \le \gamma - 1$ ($j, 0 \le j \le p - 1$). For our discussions, we describe each column of $H(\gamma, p)$ by a pair (j, k) where j is the index of the column group, and k, $0 \le k \le p - 1$, is the index of the column within the column group.

In the following, we construct SCB-SC codes by using an unwrapping procedure defined by a *cutting vector* $\boldsymbol{\xi} = [\xi_0, \cdots, \xi_{\gamma-1}]$ comprising γ step parameters with $0 \le \xi_0 < \xi_1 < \cdots < \xi_{\gamma-1} \le p$. We assume that $\xi_i, i \in \{0, 1, \cdots, \gamma - 1\}$ are distinct. In the general case, these parameters are not necessarily distinct. The cutting vector is used in the construction of SCB-SC codes to form the two matrices H_0 and H_1 as follows. The matrix H_0 of size $\gamma p \times p^2$ is formed by assigning each circulant matrix in row group i and column group j, $j < \xi_i$, of $H(\gamma, p)$ to the equivalent position in H_0. All other remaining elements of H_0 are then set to 0. Furthermore, the matrix H_1 is defined as $H_1 = H(\gamma, p) - H_0$.

Let L be a positive integer called *coupling length* which determines the number of copies of the H_0 and H_1 submatrices, resp., in the parity-check matrix of an SCB-SC code. Given the coupling length L, the parity-check matrix of an SCB-SC code is defined as

$$H(\gamma, p, L, \boldsymbol{\xi}) \triangleq \begin{bmatrix} H_0 & 0 & 0 & \cdots & 0 & 0 \\ H_1 & H_0 & 0 & \cdots & 0 & 0 \\ 0 & H_1 & H_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & H_1 & H_0 \\ 0 & 0 & 0 & \cdots & 0 & H_1 \end{bmatrix}, \tag{2}$$

and has dimension $(L + 1)\gamma p \times Lp^2$. Thus, the total length of the code is Lp^2 bits. The graphical representation of a spatially-coupled code can be viewed as coupling together a series of L identical and disjoint Tanner graphs into a single coupled chain [24] with a design code rate of $R_L = 1 - \frac{(L+1)\gamma p - \gamma + 1}{Lp^2}$. For the considered construction, the *constraint length* ν, which is the maximal width of the non-zero area in each row of $H(\gamma, p, L, \boldsymbol{\xi})$, is equal to p^2. Note that in the above construction the syndrome former memory m is assumed to be 1. In the general case, where $m \ge 1$, we have $H(\gamma, p) = H_0 + H_1 + \cdots + H_m$.

Array-based LDPC codes [35] are the simplest SCB codes with functions $f(i) = i$ and $g(j) = j$. We will refer to the class of spatially-coupled codes with underlying array-based matrices as AB-SC codes. Figure 1 shows an example of an AB-SC construction where $\gamma = 3$, $p = 11$, $L = 2$, and $\boldsymbol{\xi} = [3, 6, 9]$. For the sake of our later discussion, we define the *region* $R_n, n \in \{1, \cdots, \gamma + 1\}$ in $H(\gamma, p)$ as the set of column groups with indices between ξ_{n-2} and ξ_{n-1} (cmp. Figure 1(a)). We assume $\xi_{-1} = 0$ and $\xi_\gamma = p$. The number of column groups within the region $R_n, n \in \{1, \cdots, \gamma + 1\}$ is denoted by r_n. The definition of regions can be expanded to SCB-SC codes, where each region is similarly defined as the set of column groups between two consecutive edges of the cutting vector (cmp. Figure 1(b)). Note that the notion of row (column) groups in SCB-LDPC codes can be expanded to SCB-SC codes, where each row (column) in the parity-check matrix of SCB-SC codes belongs to a row (column) group with index $0 \le i \le \gamma - 1$ ($0 \le j \le p - 1$), that is equal to the row (column) group of the corresponding row (column) in the underlaying SCB-LDPC code.

Note that the underlying SCB-LDPC codes include several families of codes in the open literature [35], [45], [46]. For properly chosen f and g, the girth of an SCB code is guaranteed to be at least 6.

TABLE I
SUMMARY OF PARAMETERS AND NOTATION

p	circulant size (and row weight) which is a prime number
γ	column weight
i	index of row group
j	index of column group
R_L	rate $= 1 - \frac{(L+1)\gamma p - \gamma + 1}{Lp^2}$
k	index of the column within the column group
$\boldsymbol{\xi}$	cutting vector with length γ
L	coupling length
ν	constraint length $= p^2$
R_n	column region n
r_n	number of column groups within region R_n
q	Galois field size
a	number of variable nodes in an absorbing set
b	number of unsatisfied check nodes in an absorbing set
w_i	edge label (weight) of edge i
N	block length of SCB-LDPC codes
$\boldsymbol{\xi}^*_{(a,b)}$	optimal cutting vector for (a, b) AS
$f(i)$ and $g(j)$	arbitrary functions used in construction of SCB-LDPC codes

The following remark describes the construction of non-binary SCB-SC codes.

Remark 1: For a given matrix $H(\gamma, p)$ of a binary SCB-LDPC code, the parity-check matrix $H_q(\gamma, p)$ of a non-binary SCB-LDPC code over $GF(q)$ can be constructed by replacing the elements with value '1' of $H(\gamma, p)$ with non-zero elements[2] of $GF(q)$. The parity-check matrix $H_q(\gamma, p, L, \boldsymbol{\xi})$ of a non-binary SCB-SC code is then constructed by the same edge spreading procedure as above.

B. Binary and Non-Binary Absorbing Sets

Consider a subgraph of a Tanner graph over $GF(q)$ induced by a variable nodes (VNs), given by the node set $\mathcal{V}$.

Definition 1 [30]: The set $\mathcal{V}$ is an (a, b) AS over $GF(q)$ if there exists an input $(v_1, v_2, \cdots, v_a) \in GF(q)^a \backslash \{0\}$ for the VNs in $\mathcal{V}$ such that there exist exactly b unsatisfied check nodes (CNs) connected to the VNs in $\mathcal{V}$. Furthermore, for each VN in $\mathcal{V}$, the number of connected satisfied check nodes is larger than the number of connected unsatisfied check nodes.

Note that if $q = 2$ (binary code), the above definition can be simplified as follows.

Definition 2 [22]: The set $\mathcal{V}$ is an (a, b) binary AS if there are b odd degree CNs connected to VNs in $\mathcal{V}$, and each VN is connected to more even degree CNs than odd degree CNs.

We also recall the definition of *elementary ASs*, a subclass of ASs that is particularly useful in characterizing the error floor performance.

Definition 3 [30]: An elementary AS is an AS with each of its neighboring satisfied check nodes having two edges connected to the AS, and each of its neighboring unsatisfied check nodes having exactly one edge connected to the AS.

In this paper, we consider non-binary SCB-SC codes defined over finite field sizes that are powers of 2 since such values offer convenient mapping of symbols to bits. We also study SCB-SC codes defined over $GF(3)$ since such codes can benefit from the additional degree of freedom offered by non-binary edge labels, while still having only moderate belief propagation decoding complexity compared to codes defined over higher field sizes. In the study of such non-binary SCB-SC codes, the following result will be useful.

Lemma 1 (see also [30]): A subset of VNs $\mathcal{V}$ is an elementary AS over $GF(q)$ if and only if:

1) *(Topological condition). For the induced subgraph corresponding to $\mathcal{V}$ and its neighboring CNs, unlabeling of all edges (converting all edge labels to one) results in a binary elementary AS.*
2) *(Weight condition). For every cycle of length $2d$, the labels of the edges w_i, $i \in \{1, \ldots, 2d\}$, satisfy:*

$$\begin{cases} \prod_{k=1}^{d} w_{2k-1} \equiv \prod_{k=1}^{d} w_{2k} \mod q, \\ \qquad \text{if } q \text{ is a positive power of 2,} \\ \prod_{k=1}^{d} w_{2k-1} \equiv (-1)^d \prod_{k=1}^{d} w_{2k} \mod q, \\ \qquad \text{if } q \text{ is an odd prime number.} \end{cases} \tag{3}$$

For the convenience of the reader, the parameters employed in the construction of SCB-SC codes and the characterization of ASs are summarized in Table I.

C. Performance Comparison for SCB-LDPC and SCB-SC Codes: A Summary

In the following, we study the error floor performance of SCB-SC codes. For a prescribed channel SNR[3] the error profile spectrum of a code represents the listing of all error types[4] observed at the output of the decoder and their number of occurrences over the course of a representative number of simulation runs. We unveil the properties of the error profile

[2]Note that in this definition, no constraint is imposed on the selection of the non-binary elements; these edge labels can be chosen completely at random. In Section V it is shown that an informed selection of the non-binary labels can result in an improved performance of the code.

[3]Note that throughout the paper, SNR is defined as $10 \log_{10} E_b/N_0$, where E_b and $\frac{N_0}{2}$ are energy per bit and noise power spectral density, respectively.

[4]Different error types include AS errors in which the decoder converges to an AS, non-AS errors in which the decoder converges to a non-AS object, and non-converging errors in which the decoder does not converge to a specific object before reaching the maximum number of iterations.

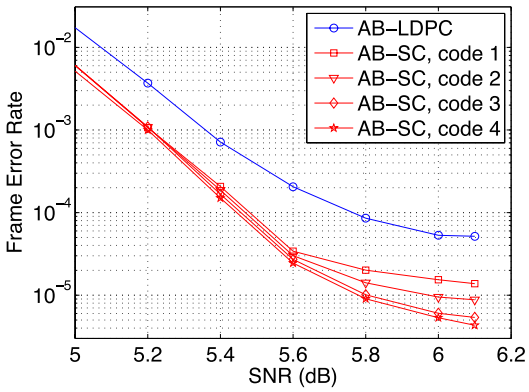


Fig. 2. Performance comparison for AB-LDPC and AB-SC codes with various random cutting vectors.

spectrum of SCB-SC codes and exploit these properties to optimize the code design. In particular, we will show that:

a) SCB-SC codes have better performance in the low error-rate (error floor) region compared to SCB-LDPC codes. Moreover, the error profile spectra are different for SCB-LDPC and SCB-SC codes. This observation motivates us to characterize ASs in SCB-SC codes and present an exact enumeration of small ASs in these codes. Based on these results, we show that, due to partitioning of the underlying parity check matrix into H_0 and H_1 components, some ASs are broken and thus do not exist in the Tanner graph of the resulting SCB-SC code. The reduction in the number of small problematic ASs in moving from block to SC designs in part explains the superior error floor performance of SCB-SC codes compared to SCB-LDPC codes.

b) The choice of the cutting vector significantly affects the performance of binary SCB-SC codes in the low error-rate region. Through our analysis we find the exact number of small ASs as a function of the cutting vector. This analysis enables us to optimize the choice of the cutting vector to design SCB-SC codes with the minimum number of problematic ASs. Before we lay out the theoretical framework for this analysis, we motivate the work with the following example.

Example 1: Figure 2 *shows simulation results for an AB-LDPC code (an SCB-SC code with* $f(i) = i$ *and* $g(j) = j$*) with block length* 4489 *bits, circulant size* 67*, column-weight* 3*, rate* $\approx$ 0.3*, and an AWGN channel, decoded using sliding window decoding.*[5] *We also present the performance curves for comparable AB-SC codes with the same constraint length* $\nu = 4489$*,* $L = 50$*, and column weight* 3*. Four distinct cutting vectors,* $\boldsymbol{\xi} = [10, 18, 56]$ *for code 1,* $\boldsymbol{\xi} = [22, 28, 55]$ *for code 2,* $\boldsymbol{\xi} = [8, 31, 40]$ *for code 3,* $\boldsymbol{\xi} = [15, 31, 47]$ *for code 4, are chosen at random for the construction of each AB-SC code shown in Figure* 2*. Table II includes the error profiles, i.e., the number of specific ASs, for the curves in Figure* 2 *at SNR* $= 6.1$ *dB. Note that the total number of simulation runs at SNR* $= 6.1$ *dB for the AB-LDPC code is* $\approx 2.4 \times 10^6$ *and for the AB-SC codes is* $\approx 1.2 \times 10^7$*. Figure* 2 *shows that AB-SC codes have a performance improvement of around one order of magnitude compared to AB-LDPC codes. We also observe that different cutting vectors result in a change in the error floor performance of AB-SC codes. This observation suggests that the number of problematic ASs in the Tanner graph of AB-SC codes changes with the choice of the cutting vector. Table II also shows that the distribution of AS errors is significantly different for AB-LDPC and AB-SC codes. Our forthcoming analysis will mathematically justify the above observations.*

Remark 2: Note that throughout this paper, the simulation results of SCB-SC codes are obtained using sliding window decoding [36], [39]. We follow the approach in the literature (e.g., [15], [38], and [16]) to compare the performance of spatially-coupled codes with constraint length ν *with the performance of block LDPC codes with block length* $N = \nu$*. In this case* ($N = \nu$)*, the normalized decoding hardware processor complexity and latency for both spatially-coupled and block codes are equal.*

III. Exact Enumeration of Binary Absorbing Sets in Separable Circulant-Based Spatially-Coupled Codes With Column Weight Three

In this section, we first focus our analysis on AB-SC codes which are a sub-class of SCB-SC codes. We show that the structure of AB-SC codes imposes additional constraints (relative to the block code case) on two variable nodes sharing a check node in an AS. We introduce an approach to calculate the exact number of ASs for binary AB-SC codes and show that the enumeration problem can be expressed as the problem of counting integer pairs within a two-dimensional geometric region specified by the set of column groups R'_n, $n \in \{1, \ldots, \gamma L + 1\}$ (see Fig. 1(b)). We show that the same introduced approach is applicable to the more general case of SCB-SC codes. In particular, the analysis in this section indicates that the number of dominant ASs is reduced for SCB-SC codes compared to SCB-LDPC codes. Although our procedure is in principle applicable to any column weight, for the ease of discussion we limit our analysis to column-weight-3 codes, in which $(3, 3)$ and $(4, 2)$ are the dominant ASs.

We first revisit the *bit, check, and pattern consistency conditions* for AB-LDPC codes from [22].

Lemma 2:

1) ***Bit consistency:*** *The neighboring CNs of a VN must have distinct row-group indices* $i_1 \neq i_2$*.*
2) ***Check consistency:*** *The neighboring VNs of a CN must have distinct column-group indices* $j_1 \neq j_2$*.*
3) ***Pattern consistency:*** *If two VNs corresponding to columns* (j_1, k_1) *and* (j_2, k_2) *share a CN* c_1 *in row group* i_1*, then* $k_1 + i_1 j_1 \equiv k_2 + i_1 j_2 \mod p$*.*

Remark 3: In a spatially-coupled structure, each CN is only connected to VNs within a window of at most γ *consecutive regions,*[6] *where* γ *is the column weight of the code. As an example, each CN in the parity-check matrix shown in Figure 1(b) is connected to the VNs in a window of at most*

[5]Note that throughout this paper, the window size of sliding window decoder is $4 \cdot \nu$, with ν being the constraint length. Also, a maximum number of 50 iterations is used.

[6]Note that the width of the window is shorter for the first and last $\gamma - 1$ columns in $H(\gamma, p, L, \boldsymbol{\xi})$.

TABLE II
ERROR PROFILE (THE NUMBER OF SPECIFIC ASS) FOR THE CODES SHOWN IN FIGURE 2 AT SNR = 6.1 dB

Error Type	$(3,3)$	$(4,2)$	$(4,4)$	$(5,1)$	$(5,3)$	$(5,5)$	other
AB-LDPC	55	23	13	2	3	8	17
AB-SC, code 1	71	6	8	0	4	3	21
AB-SC, code 2	42	4	7	0	2	3	15
AB-SC, code 3	29	4	3	0	1	2	16
AB-SC, code 4	23	2	2	0	1	1	13

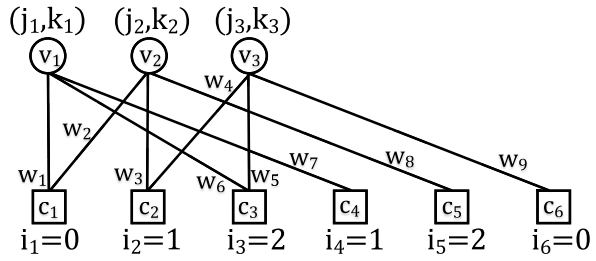


Fig. 3. Structure of a $(3,3)$ AS over $GF(q)$. Note that in the binary case, w_1 through w_9 are equal to '1'.

$\gamma = 3$ *regions. In addition to the pattern consistency condition, this coupling-induced property imposes extra constraints on the valid choices for column groups of two VNs sharing a CN.*

The following example clarifies Remark 3.

Example 2: Consider the $H(3, 11, 2, [3, 6, 9])$ *parity-check matrix in Figure 1(b). If two VNs share a CN in the second row group* ($i = 1$)*, then the regions which the two VNs belong to must be within the set* $\{R'_1, R'_2\}$*. Assuming that* v_1 *belongs to* R'_1 *with width* $r_1 = 3$ *and that* v_2 *belongs to* R'_2 *with width* $r_2 = 3$*, the following condition must be satisfied:*

$$k_1 + j_1 \equiv k_2 + j_2 \mod p,$$
$$0 \le j_1 \le r_1 - 1, \quad r_1 \le j_2 \le r_1 + r_2 - 1.$$

The following lemma presents the size of dominant ASs in SCB-SC codes with column weight 3.

Lemma 3 [24]: The smallest possible ASs in the Tanner graph of $H(3, p, L, \boldsymbol{\xi})$ *are* $(3,3)$ *and* $(4,2)$ *ASs.*

Since small ASs typically contribute the most to the error floor of LDPC codes, we focus our analysis on $(3,3)$ and $(4,2)$ ASs, as illustrated in Table II. In [24], the authors show that the number of dominant ASs in column-weight-3 AB-SC codes grows linearly with the coupling length L. Through our analysis, we provide the exact number of $(3,3)$ and $(4,2)$ ASs in the Tanner graph of $H(3, p, L, \boldsymbol{\xi})$ as a function of circulant size p, coupling length L, and the cutting vector $\boldsymbol{\xi}$.

1) Analysis of (3, 3) ASs: We consider the $(3,3)$ AS structure shown in Figure 3. Without loss of generality, we assume that variable nodes v_1 and v_2 share a check node in row group $i_1 = 0$, v_2 and v_3 share a check node in row group $i_2 = 1$, and v_1 and v_3 share a check node in row group $i_3 = 2$. Thus, the pattern consistency constraints lead to:

$$k_1 + 2j_1 \equiv k_3 + 2j_3, \quad k_1 = k_2, \quad k_2 + j_2 \equiv k_3 + j_3, \quad (4)$$

where j_1, j_2, j_3, k_1, k_2, and k_3 are in $\{0, 1, \cdots, p-1\}$ and all equations are modulo p. The above equations results in the following equation, involving only the indices of the column groups:

$$j_2 \equiv 2j_1 - j_3 \mod p. \quad (5)$$

It was shown in [22] that by fixing the values of j_1, j_3, and k_1 in the above equations, the values of all other variables can be uniquely determined. In the case of AB-LDPC codes, j_1 and j_3 can take any pair of distinct values from 0 to $p-1$. Index k_1 also can take any integer value from 0 to $p-1$. Thus, there exist $p^2(p-1)$ ASs of size $(3,3)$ in column-weight-3 AB-LDPC codes. Through our analysis below, we show that not all pairs of (j_1, j_3) are valid in the case of AB-SC codes; this constraint results in a fewer number of $(3,3)$ ASs. This reduction in the number of $(3,3)$ AS in AB-SC codes compared to AB-LDPC codes in part explains our initial observation in Example 1, where AB-SC codes showed a superior error floor performance compared to AB-LDPC codes.

Lemma 4: The three VNs in a $(3,3)$ *ASs span at most three consecutive regions in column-weight-3 AB-SC codes.*

Proof: Based on Remark 3 and the fact that each pair of VNs in a $(3,3)$ AS are connected through a satisfied CN (Figure 3), the three VNs span at most three consecutive regions. ■

Lemma 4 enables us to categorize all $(3,3)$ ASs in $H(3, p, L, \boldsymbol{\xi})$ into four exhaustive mutually-exclusive cases[7]:

- **Case 1**: All three VNs are in the same region.
- **Case 2**: Two VNs are in the same region, the third VN is in the next region.
- **Case 3**: Two VNs are in the same region, the third VN is in the previous region.
- **Case 4**: The three VNs each belong to a different region and the three regions are consecutive.

a) Number of ASs in Case 1: In this case, we put a window over each region (R'_1 to R'_{3L+1}) and count the number of $(3,3)$ ASs within that window.

Lemma 5: The total number of $(3,3)$ *ASs in Case 1, denoted by* $F_1(p, L, r_1, r_2, r_3, r_4)$*, is:*

$$F_1(p, L, r_1, r_2, r_3, r_4) = F_1^{R_1}(p, r_1) + L \cdot F_1^{R_1}(p, r_2) + (L-1) \cdot F_1^{R_1}(p, r_1 + r_4) + L \cdot F_1^{R_1}(p, r_3) + F_1^{R_1}(p, r_4), \quad (6)$$

where $F_1^{R_n}(p, r_n)$ *is the number of* $(3,3)$ *ASs within region* R_n *of width* r_n*, for a given circulant size* p*.*

[7] It is trivial to show that due to the check consistency property of AB-LDPC codes [22], it is not possible to have two VNs in the same region R_n and the third VN in region R_{n-2} or R_{n+2}.

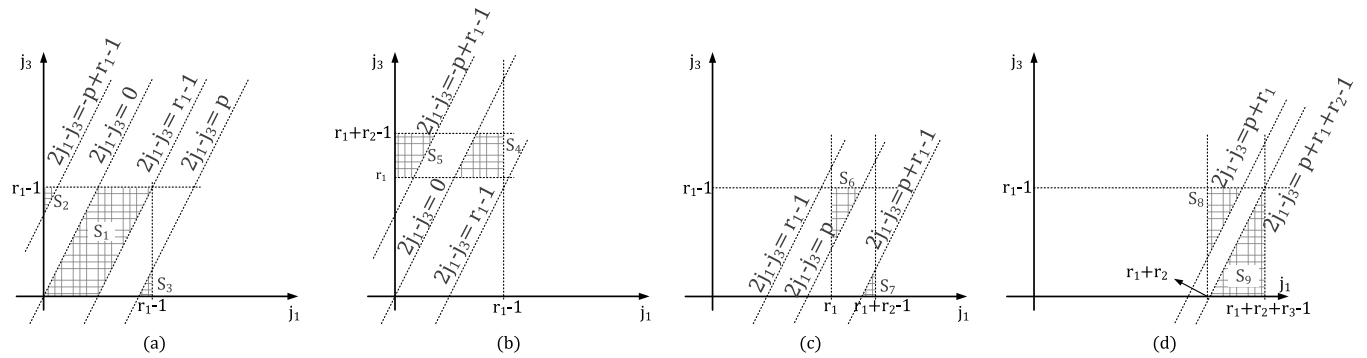


Fig. 4. (a) Example of Case 1. All VNs in region R'_1. (b) Example of Case 2. VNs 1 and 2 are in region R'_1 and VN 3 is in region R'_2. (c) Example of Case 3. VN 3 is in region R'_1 and VNs 1 and 2 are in region R'_2. (d) Example of Case 4. VN 3 is in region R'_1, VN 2 is in region R'_2 and VN 1 is in region R'_3.

Proof: The total number of $(3, 3)$ ASs in Case 1 is equal to the summation of the counted ASs within regions R'_1 through R'_{3L+1}, i.e.,

$$F_1(p, L, r_1, r_2, r_3, r_4) = \sum_{n=1}^{3L+1} F_1^{R'_n}(p, r'_n). \tag{7}$$

Here and elsewhere, r'_n denotes the number of column groups in R'_n. One can show that

$$F_1^{R_1}(p, m) = F_1^{R'_n}(p, m), \quad n \in \{1, \cdots, 3L+1\}. \tag{8}$$

By substituting each term in (7) with the LHS of (8), and by the fact that for any $k \in \mathbb{N}, 1 \le k \le 3L+1$,

$$\begin{aligned} r'_1 = r_1, \quad \{k : (k \mod 3) = 2\} &\to r'_k = r_2, \\ r'_{3L+1} = r_4, \quad \{k : (k \mod 3) = 3\} &\to r'_k = r_3, \\ \{k : (k \mod 3) = 1, k \ne 1, k \ne 3L+1\} &\to r'_k = r_1 + r_4, \end{aligned}$$

(6) can be obtained. ∎

As an example, we consider that all the VNs are in region R'_1. The problem of finding valid column groups j_1 and j_3 can be graphically interpreted as the problem of counting the integer pairs (j_1, j_3) within the areas S_1, S_2 and S_3 in Figure 4(a). Note that based on the values of p and r_1, areas S_2 and S_3 can be either the empty set $\varnothing$ or a triangle. The number of (integer) points existing in S_1, S_2, and S_3, denoted by $N_{S_1}(r_1)$, $N_{S_2}(p, r_1)$, and $N_{S_3}(p, r_1)$, respectively, are found to be (for more details see [32]):

$$N_{S_1}(r_1) = \begin{cases} \dfrac{(r_1-1)^2}{2} & \text{if } r_1 \text{ is odd,} \\ \dfrac{r_1(r_1-2)}{2} & \text{if } r_1 \text{ is even,} \end{cases}$$

and

$$N_{S_2}(p, r_1) = N_{S_3}(p, r_1) = \begin{cases} \frac{(2r_1-p)^2-1}{4} & \text{if } 2r_1 \ge p+2, \\ 0 & \text{if } 2r_1 < p+2. \end{cases}$$

Therefore, the total number of ASs within region R_1 is

$$F_1^{R_1}(p, r_1) = p \cdot \left(N_{S_1}(r_1) + N_{S_2}(p, r_1) + N_{S_3}(p, r_1)\right). \tag{9}$$

Note that the multiplication by p in (9) is due to p choices for k_1. As an example, if $p = 11$ and $r_1 = 8$, the number of $(3, 3)$ ASs with all their three variable nodes in region 1 is equal to $11 \times (24 + 6 + 6) = 396$.

b) Number of ASs in Case 2: Here, we put a window over each two consecutive regions ($\{R'_1, R'_2\}$ through $\{R'_{3L}, R'_{3L+1}\}$).

Lemma 6: The total number of $(3, 3)$ *ASs in Case 2 is obtained as follows:*

$$\begin{aligned} F_2(p, L, r_1, r_2, r_3, r_4) &= F_2^{R_1}(p, r_1, r_2) \\ &\quad + L \cdot F_2^{R_2}(p, r_2, r_3) + (L-1) \cdot F_2^{R_3}(p, r_3, r_1 + r_4) \\ &\quad + (L-1) \cdot F_2^{R_1}(p, r_1 + r_4, r_2) + F_2^{R_3}(p, r_3, r_4), \end{aligned} \tag{10}$$

where $F_2^{R_n}(p, r_n, r_{n+1})$ *denotes the number of* $(3, 3)$ *ASs with two VNs in* R_n *of width* r_n *and one VN in* R_{n+1} *of width* r_{n+1}.

A complete discussion of the calculation for each term on the RHS in (10) as well as for the related terms $F_3^{R_{1/2/3}}$ and $F_4^{R_{1/2/3}}$ below in Lemma 7 and 8, respectively, can be found in [32].

As an example, the valid areas for the pair (j_1, j_3) when VNs 1 and 2 are in R_1 and VN 3 is in R_2 are shown in Figure 4(b). Note that based on the values of p, r_1, and r_2, the areas S_4 and S_5 can be empty set $\varnothing$, a triangle, or a trapezoid.

c) Number of ASs in Case 3: Similar to Case 2, we put a window over each two consecutive regions. Here, we count the number of $(3, 3)$ ASs which have two VNs in the second region and one VN in the first region.

Lemma 7: The total number of $(3, 3)$ *ASs in Case 3 is obtained as follows:*

$$\begin{aligned} F_3(p, L, r_1, r_2, r_3, r_4) &= F_3^{R_1}(p, r_1, r_2) \\ &\quad + L \cdot F_3^{R_2}(p, r_2, r_3) + (L-1) \cdot F_3^{R_3}(p, r_3, r_1 + r_4) \\ &\quad + (L-1) \cdot F_3^{R_1}(p, r_1 + r_4, r_2) + F_3^{R_3}(p, r_3, r_4), \end{aligned} \tag{11}$$

where $F_3^{R_n}(p, r_n, r_{n+1})$ *is the number of* $(3, 3)$ *ASs with one VN in* R_n *of width* r_n *and two VNs in* R_{n+1} *of width* r_{n+1}.

As an example for Case 3, the areas for the pair (j_1, j_3), when VN 3 is in R_1 and VNs 2 and 3 are in R_2, are displayed in Figure 4(c). Again, based on the values of p, r_1, and r_2, areas S_6 and S_7 can be $\varnothing$, triangles, or trapezoids.

d) Number of ASs in Case 4: Here, we put a window over each three consecutive regions ($\{R'_1, R'_2, R'_3\}$ through

$\{R'_{3L-1}, R'_{3L}, R'_{3L+1}\}$). For each window, we count the number of (3, 3) ASs which have one VN in each region.

Lemma 8: The total number of (3, 3) *ASs in Case 4 is obtained as follows:*

$$\begin{aligned} &F_4(p, L, r_1, r_2, r_3, r_4) \\ &\quad = (L-1)\cdot F_4^{R_2}(p, r_2, r_3, r_1 + r_4) + F_4^{R_1}(p, r_1, r_2, r_3) \\ &\qquad + F_4^{R_2}(p, r_2, r_3, r_4) + (L-1)\cdot F_4^{R_3}(p, r_3, r_1 + r_4, r_2) \\ &\qquad + (L-1)\cdot F_4^{R_1}(p, r_1 + r_4, r_2, r_3), \end{aligned} \tag{12}$$

where $F_4^{R_n}(p, r_n, r_{n+1}, r_{n+2})$ *is the number of* (3, 3) *ASs with one VN in* R_n *of width* r_n*, one VN in* R_{n+1} *of width* r_{n+1}*, and one VN in* R_{n+2} *of width* r_{n+2}.

As an example, Figure 4(d) highlights the valid areas for the pair (j_1, j_3) when VN 3 is in R_1, VN 2 is in R_2, and VN 1 is in R_3. Again based on the values of p, r_1, r_2, and r_3, the areas S_8 and S_9 can be $\varnothing$, triangles, or trapezoids.

For a given circulant size p, a coupling length L and a cutting vector $\boldsymbol{\xi}$, the following equation provides the exact number of (3, 3) ASs, which is denoted by $A_{(3,3)}(3, p, L, \boldsymbol{\xi})$:

$$A_{(3,3)}(3, p, L, \boldsymbol{\xi}) = \sum_{n=1}^{4} F_n(p, L, r_1, r_2, r_3, r_4), \tag{13}$$

where $r_1 = \xi_1$, $r_2 = \xi_2 - \xi_1$, $r_3 = \xi_3 - \xi_2$, and $r_4 = p - \xi_3$, and the functions $F_n(p, L, r_1, r_2, r_3, r_4), n \in \{1, 2, 3, 4\}$, is calculated as in (6), (10), (11), and (12).

Remark 4: The AS enumeration method presented in this section can be applied to any (a, b) *AS. However, for larger ASs, the problem is more involved as the problem of finding the valid points for the column group indices is over higher dimensional spaces. For larger ASs, column group indices are not always a function of only two column groups indices. Therefore, the areas for the valid column groups must be described over higher dimensional spaces.*

In the more general case of SCB-SC, the structure of the underlying SCB code imposes similar constraints on the indices of neighboring check and variable nodes [23]:

- ***Bit consistency:*** The neighboring check nodes of a VN must have distinct row group labels $i_1 \neq i_2$.
- ***Check consistency:*** The neighboring VNs of a check node must have distinct column group labels $j_1 \neq j_2$.
- ***Pattern consistency:*** If two VNs corresponding to columns (j_1, k_1) and (j_2, k_2) share a check node in row group i, then:

$$k_1 + f(i)g(j_1) \equiv k_2 + f(i)g(j_2) \mod p.$$

Note that the only difference between the general SCB-SC codes and AB-SC codes is that the pattern consistency constraints lead to different equations on the indices of the variable nodes which share a check node. Therefore, a similar approach is applicable for the enumeration of ASs in SCB-SC codes. In the case of (3, 3) ASs shown in Figure 3, the pattern consistency constraints imply that

$$\begin{aligned} k_1 + f(0)g(j_1) &\equiv k_2 + f(0)g(j_2) \mod p, \\ k_2 + f(1)g(j_2) &\equiv k_3 + f(1)g(j_3) \mod p, \\ k_1 + f(2)g(j_1) &\equiv k_3 + f(2)g(j_3) \mod p. \end{aligned} \tag{14}$$

Note that in order to have a linearly independent set of equations, we assume that f and g are one-to-one functions. The above set of equations, along with the check consistency constraints ($j_1 \neq j_2$, $j_1 \neq j_3$, and $j_2 \neq j_3$), results in a two-dimensional area for valid choices of the pair (j_1, j_3). Note that in the case of SCB-SC codes, the valid two-dimensional areas for the column group indices (j_1, j_3) depend on the choice of the functions $f(i)$ and $g(j)$ and vary for different choices of these functions. The following example illustrates the enumeration of ASs for SCB-SC codes.

Example 3: In this example, we consider two SCB-SC codes with different functions $f(i)$ *and* $g(j)$ *and provide the AS enumeration for these two codes. For code 1, we assume that* $f(i) = i^2$ *and* $g(j) = 2j$ *and for code 2 that* $f(i) = i^3$ *and* $g(j) = j$.

- *For case 1, by substituting* $f(0) = 0$, $f(1) = 1$, $f(2) = 4$ *and* $g(j) = 2j$ *in the three equations in* (14)*, the pattern consistency constraint results in*

$$\left.\begin{aligned} k_1 &\equiv k_2 \mod p \\ k_2 + 2j_2 &\equiv k_3 + 2j_3 \mod p \\ k_1 + 8j_1 &\equiv k_3 + 8j_3 \mod p \end{aligned}\right\} \Rightarrow j_2 \equiv 4j_1 - 3j_3 \mod p. \tag{15}$$

- *In the case of code 2, substituting* $f(0) = 0$, $f(1) = 1$, $f(2) = 8$ *and* $g(j) = j$ *in* (14) *results in the following equation over column group indices* j_1, j_2 *and* j_3.

$$\left.\begin{aligned} k_1 &\equiv k_2 \mod p \\ k_2 + j_2 &\equiv k_3 + j_3 \mod p \\ k_1 + 8j_1 &\equiv k_3 + 8j_3 \mod p \end{aligned}\right\} \Rightarrow j_2 \equiv 8j_1 - 7j_3 \mod p. \tag{16}$$

Note that the resulting equations in (15) *and* (16) *are not equivalent to the equation in* (5)*. Therefore the AS counts are different for the AB-SC code, the SCB-SC code 1 and the SCB-SC code 2, which ultimately result in different optimal cutting vectors for these three codes. Table IV in Section VI provides the enumeration results for the AB-SC codes, the SCB-SC code 1 and the SCB-SC code 2, resp., with coupling length* $L = 50$ *and circulant sizes* $p = 29, 67$.

2) Analysis of (4, 2) ASs: In this section, we summarize the approach to find the exact number of (4, 2) ASs in AB-SC codes with column-weight 3. Details can be found in [32]. Analogous results can be similarly obtained for SCB-SC codes. These results further explain the differences in both the error profile spectrum and the performance of SCB-LDPC and SCB-SC codes observed in Example 1.

It was shown in [22] that (in the block case) each (4, 2) AS is formed by exactly two distinct (3, 3) ASs. We first review how to count the number of (4, 2) ASs in AB-LDPC codes [22].

Lemma 9: ([22]*) The total number of* (4, 2) *ASs in the Tanner graph corresponding to* $H(3, p)$ *is equal to* $\frac{3p^2(p-1)}{2}$.

In the case of AB-LDPC codes, as shown in Figure 5, it can be proved that for each pair of unsatisfied check nodes in a (3, 3) AS, there always exists a new variable node to satisfy the two check nodes. Therefore, each (3, 3) AS in AB-LDPC codes always leads to three (4, 2) ASs. In contrast, in the

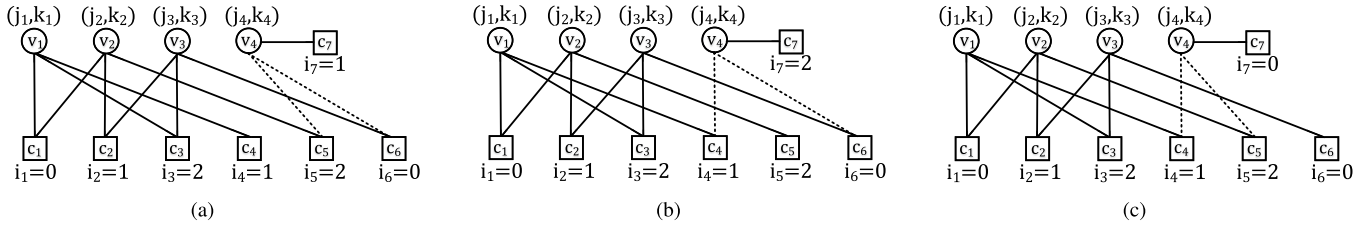


Fig. 5. The three possible cases for $(4, 2)$ ASs.

case of AB-SC codes, due to the structure of these codes, one may not always be able to find three new variable nodes connected to each pair of unsatisfied check nodes in a $(3, 3)$ AS to complete three $(4, 2)$ AS configurations.

By categorizing $(4, 2)$ ASs into three labeling cases in Figure 5, one can show that, based on pattern consistency conditions, each labeling imposes a different equation on the choice of column group indices:

$$\begin{aligned} &\text{Case (a): } j_1 \equiv j_4, \\ &\text{Case (b): } 2j_3 - j_1 \equiv j_4, \\ &\text{Case (c): } 3j_1 - 2j_3 \equiv j_4, \end{aligned} \tag{17}$$

where all equations are modulo p.

In the previous subsection, the problem of counting $(3, 3)$ ASs was mapped into a problem of counting the number of valid points for the pair (j_1, j_3) in a two dimensional space. Here, we apply a similar approach to count the number of $(4, 2)$ ASs. Equations in (17) further limit the areas for valid choices of (j_1, j_3) pairs in the two dimensional space. The following example clarifies the procedure for counting the number of $(4, 2)$ ASs in AB-SC codes based on the previously counted $(3, 3)$ ASs (details are omitted for brevity).

Example 4: Consider $(3, 3)$ *ASs with* v_1*,* v_2*, and* v_3 *all in region* R_1*. The valid area for the choice of the pair* (j_1, j_3) *in this case is shown in Figure 4(a) and the count of these ASs is expressed as* $F_1^{R_1}(p, r_1)$ *in* (9)*. Based on* (17)*, we find the areas for the valid choices of pairs* (j_1, j_3) *in the three* $(4, 2)$ *AS cases discussed above, shown in Figure 6.*

- ***Case (a):*** *The addition of variable node* v_4 *does not impose any new constraints of the choice of the pair* (j_1, j_3) *(Figure 6(a)).*
- ***Case (b):*** *The addition of* v_4 *imposes that* $0 \leq 2j_3 - j_1 \leq r_1 - 1 \mod p$*, as illustrated in Figure 6(b).*
- ***Case (c):*** *The addition of* v_4 *imposes that* $0 \leq 3j_1 - 2j_3 \leq r_1 + r_2 - 1 \mod p$*, which is illustrated in Figure 6(c).*

In the above example, we have only considered $(3, 3)$ ASs with all their variable nodes in region R_1 and we have computed the number of the resulting $(4, 2)$ ASs. To enumerate all $(4, 2)$ ASs, we need to consider each term in Lemmas 5, 6, 7, and 8 one by one. A detailed discussion on valid choices for pairs (j_1, j_3) for each term in Lemmas 5, 6, 7, and 8 can be found in [32].

Remark 5: Note that each $(3, 3)$ *AS in SCB-LDPC codes results in three* $(4, 2)$ *ASs whereas each* $(3, 3)$ *AS in SCB-SC codes does not necessarily result in three* $(4, 2)$ *ASs. The reason is that in the SCB-SC case, the valid areas for the pairs* (j_1, j_3) *in the cases (a), (b), and (c) above are always subareas of the valid area for* $(3, 3)$ *ASs. This is demonstrated in Example 4 and explains our observation in Example 1 where, compared to SCB-LDPC codes, a smaller percentage of errors in the error profile of AB-SC codes is due to* $(4, 2)$ *ASs.*

TABLE III
OPTIMAL CUTTING VECTORS FOR AB-SC CODES, VARIOUS CIRCULANT SIZES, AND $L = 50$

p	Optimal cutting vector
67	$[16, 34, 52]$
97	$[22, 48, 74]$
107	$[24, 53, 81]$
113	$[26, 56, 86]$
211	$[49, 105, 161]$
307	$[72, 153, 235]$

IV. OPTIMAL CUTTING VECTOR FOR AB-SC CODES WITH COLUMN WEIGHT THREE

Based on our analysis in Section III, we provide the optimal choice of the cutting vector for specific choices of parameters describing AB-SC codes; analogous results are readily available for SCB-SC codes.

Definition 4 ([24]): For any given prime circulant size p *and any given coupling length* L*, an optimal cutting vector corresponding to* (a, b) *ASs is defined as*

$$\boldsymbol{\xi}^*_{(a,b)}(p, L) = \arg\min_{\boldsymbol{\xi}} A_{(a,b)}(\gamma, p, L, \boldsymbol{\xi}), \tag{18}$$

where $A_{(a,b)}(\gamma, p, L, \boldsymbol{\xi})$ *is the exact number of* (a, b) *ASs in an SCB-SC code with a given circulant size* p*, a coupling length* L*, and a cutting vector* $\boldsymbol{\xi}$*.*

Remark 6: The problem of finding the optimal cutting vector is essentially mapped to finding an integer vector which minimizes the function in (18)*. Compared to [24], where the optimal cutting vector for* $p \leq 23$ *is found by an exhaustive computer search, our approach is computationally less complex and offers the optimal cutting vector for large choices of* p*. As an example, the list of optimal cutting vectors for AB-SC codes with* $L = 50$ *and* $p = 67, 97, 107, 113, 211, 307$ *is provided in Table III.*

Note that although the enumeration approach presented in this paper is not limited to high-rate codes, it simplifies the optimization of the cutting vector to a large extent if the row weight is large. As an example, for a rate $\approx \frac{1}{2}$ *code with row weight 7 and column weight is 3, there are only* $\binom{7}{3} = 35$ *vector combinations for the cutting vector. However, for a rate* ≈ 0.95 *code with row weight 67 and column weight 3, the*

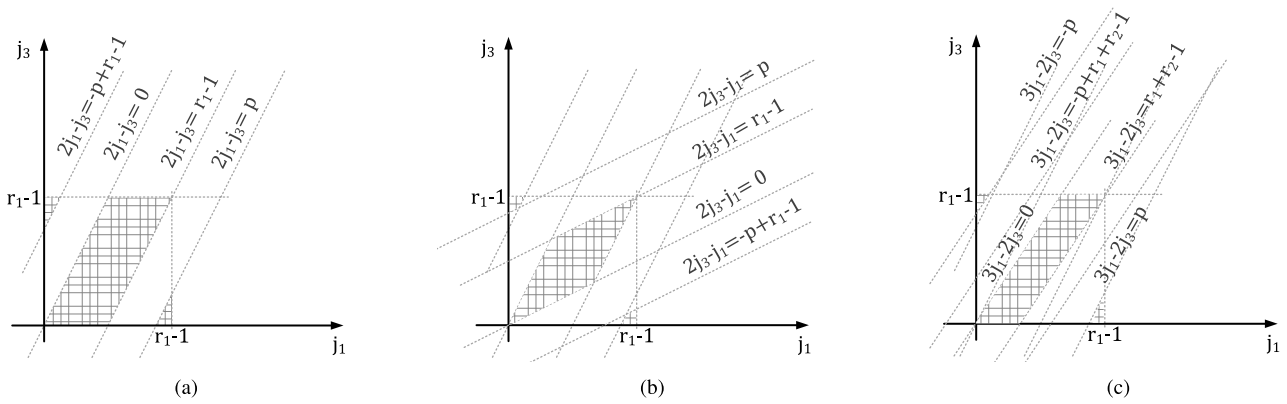


Fig. 6. Areas corresponding to valid choices for j_1 and j_3 in Example 4.

number of vector combinations for the cutting vector is $\binom{67}{3} = 47905$. *As a result, the introduced approach is most beneficial for high-rate codes as it significantly reduces the dimension of the search space for finding the best cutting vector.*

Although the choice of the cutting vector significantly affects the performance of binary SCB-SC codes, we show in Section V that the choice of the cutting vector is not a critical parameter in optimizing non-binary SCB-SC codes. In this case, the non-binary edge labels provide enough degree of freedom to remove problematic ASs from non-binary SCB-SC codes.

V. Absorbing Set Analysis for Column-Weight-Three Non-Binary SCB-SC Codes

It was recently shown that non-binary spatially-coupled codes have superior iterative threshold and finite-length performance compared to non-binary block LDPC codes [13]–[15]. Motivated by these findings, in this section we provide a study of non-binary ASs for non-binary SCB-SC codes. In particular, we first present the average number of $(3, 3)$ non-binary ASs (NB-ASs) in non-binary SCB-SC codes. We also provide an approach to design non-binary SCB-SC codes with a reduced number of problematic NB-ASs.

The following lemma provides the average number of $(3, 3)$ NB-ASs in non-binary SCB-SC codes with column-weight 3.

Lemma 10: Consider an SCB-SC code over $GF(q)$*, where* q *is a power of a prime number, with circulant size* p*, coupling length* L*, cutting vector* $\boldsymbol{\xi}$*, and random assignment of edge labels. The number of* $(3, 3)$ *NB-ASs, averaged over all possible edge label assignments, denoted by* $\overline{A}^{q}_{(3,3)}(3, p, L, \boldsymbol{\xi})$*, is*

$$\overline{A}^{q}_{(3,3)}(3, p, L, \boldsymbol{\xi}) = \frac{1}{(q-1)} \sum_{n=1}^{4} F_n(p, L, r_1, r_2, r_3, r_4), \tag{19}$$

where $r_1 = \xi_1$*,* $r_2 = \xi_2 - \xi_1$*,* $r_3 = \xi_3 - \xi_2$*, and* $r_4 = p - \xi_3$*, and the functions* $F_n(p, L, r_1, r_2, r_3, r_4), n = \{1, 2, 3, 4\}$*, are calculated based on* (6)*,* (10)*,* (11)*, and* (12)*.*

Proof: Consider the structure of a $(3, 3)$ AS in Figure 3 with non-binary edge labels $w_1, w_2, \ldots, w_6$. There exist $(q-1)^6$ unique choices for a set of six edge labels. Based on the weight condition of NB-ASs in (3), the edge labels in a NB-AS satisfy $w_1 w_3 w_5 = w_2 w_4 w_6 \mod q$. By choosing w_1 through w_5 independently from $GF(q)\backslash\{0\}$, the edge label w_6 can be uniquely determined. Thus, there exist $(q-1)^5$ choices for edge labels which result in ASs over $GF(q)$. As a result, the average number of $(3, 3)$ non-binary ASs is $\overline{A}^{q}_{(3,3)}(3, p, L, \boldsymbol{\xi}) = \frac{(q-1)^5}{(q-1)^6} A_{(3,3)}(3, p, L, \boldsymbol{\xi}) = \frac{1}{(q-1)} \sum_{n=1}^{4} F_n(p, L, r_1, r_2, r_3, r_4)$. ∎

The above lemma illustrates that on average, a ratio of $\frac{1}{(q-1)}$ of all binary $(3, 3)$ ASs in the unlabeled Tanner graph of a non-binary SCB-SC code result in problematic non-binary ASs. As an example, consider $q = 3$. Then, on average, 50% of the binary ASs in the unlabeled Tanner graph do not result in problematic substructures in the Tanner graph after a random assignment of edge labels over $GF(3)$. This observation in part explains the better performance of non-binary SCB-SC codes in error floor region compared to their binary counterparts.

In our previous work [30] we showed that non-binary edge labels provide a new degree of freedom in the design of non-binary block LDPC codes. By a systematic manipulation of the edge labels, our code design algorithm significantly decreases the number of problematic elementary NB-ASs in the Tanner graph of non-binary block LDPC codes. A similar approach can be exploited in the case of non-binary SCB-SC codes to design codes with superior error floor performance. For a given non-binary SCB-SC code, our proposed approach first identifies a list of problematic elementary ASs for the given code. Then, it alters the label of at least one edge in each problematic elementary NB-AS in the Tanner graph such that the weight condition of elementary NB-AS is violated for each targeted AS. This code design algorithm can be summarized as follows.

1) Based on the parameters of the code (column weight and girth), we choose a set of problematic elementary ASs which we want to eliminate.
2) For each targeted elementary AS we find all binary elementary ASs of the same size in the unlabeled Tanner graph. These unlabeled elementary ASs are the subgraphs which satisfy the topological conditions of elementary NB-AS.

TABLE IV

(3, 3) AS COUNT, SCB-SC CODES, $L = 50$, $q = 2$, CODE 1 WITH $f(i) = i^2$ AND $g(j) = 2j$ AND CODE 2 WITH $f(i) = i^3$ AND $g(j) = j$

	p	Optimal cutting vector	Count for optimal	Random cutting vector	Count for Random
AB-SC	29	$\boldsymbol{\xi}^*_{(3,3)}(29,50) = [6,14,21]$	521913	$\boldsymbol{\xi}_{(3,3)}(29,50) = [4,7,13]$	635651
	67	$\boldsymbol{\xi}^*_{(3,3)}(67,50) = [15,33,51]$	6723852	$\boldsymbol{\xi}_{(3,3)}(67,50) = [10,18,56]$	7261527
SCB-SC Code 1	29	$\boldsymbol{\xi}^*_{(3,3)}(29,50) = [5,14,25]$	494421	$\boldsymbol{\xi}_{(3,3)}(29,50) = [4,7,13]$	657198
	67	$\boldsymbol{\xi}^*_{(3,3)}(67,50) = [12,33,57]$	6394949	$\boldsymbol{\xi}_{(3,3)}(67,50) = [10,18,56]$	7249467
SCB-SC Code 2	29	$\boldsymbol{\xi}^*_{(3,3)}(29,50) = [5,14,23]$	493609	$\boldsymbol{\xi}_{(3,3)}(29,50) = [4,7,13]$	652906
	67	$\boldsymbol{\xi}^*_{(3,3)}(67,50) = [13,34,55]$	6408081	$\boldsymbol{\xi}_{(3,3)}(67,50) = [10,18,56]$	7219652

3) For each candidate subgraph found in Step 2, we check if the weights conditions of elementary NB-ASs are satisfied or not. Note that the weight condition is different for field sizes of powers of two and for $GF(3)$ (as shown in Lemma 1).
4) If the weight condition is also satisfied for a candidate subgraph, it is a problematic elementary NB-AS. By changing at least one edge label in the subgraph, we attempt to violate the edge label condition of ASs for at least one cycle in that subgraph.
5) We continue this process until all targeted elementary ASs are removed or no more elementary AS can be removed from the Tanner graph of the code.

Our results in Section VI show that our optimized NB-SCB-SC codes outperform unoptimized SCB-SC codes by more than one order of magnitude for the AWGN channel.

Remark 7: In contrast to binary SCB-SC codes (see Section III), the choice of the cutting vector in optimized non-binary SCB-SC codes is not as critical: For a given underlying SCB-LDPC code, different choices of the cutting vector result in $\binom{p}{\gamma}$ different binary SCB-SC codes. For the given underlying binary SCB-LDPC code, a given cutting vector, and field size q, there are $(q-1)^{\gamma p^2}$ different non-binary SCB-SC codes (obtained by different edge label selections). Since generally $(q-1)^{\gamma p^2} \gg \binom{p}{\gamma}$, this roughly verifies the significantly larger degrees of freedom offered by the choice of the edge labels compared to the ones offered by the choice of the cutting vector.

VI. EXPERIMENTAL RESULTS

In this section, we present the simulation results for our designed binary and non-binary AB-SC codes.[8] Note that a sliding window decoder based on the soft-xor algorithm is implemented to capture these simulation results.

Figure 7 displays the performance of high-rate column-weight-3 AB-SC codes (Figure 7(a)) and SCB-SC codes discussed in Example 3 (Figure 7(b)), constructed using different cutting vectors. The performance curves shown in this figure are for column-weight-3 codes with constraint length $\nu = 841$, $p = 29$, $L = 50$, and design rate $R_L = 0.89$, constructed using cutting vectors $\boldsymbol{\xi}^*_{(3,3)}(29,50)$ (optimal for (3, 3) ASs) and a randomly selected cutting vector $\boldsymbol{\xi} = [4,7,13]$. It can be observed that the optimal cutting vector for (3, 3) ASs $\boldsymbol{\xi}^*_{(3,3)}(29,50)$ improves the performance of the AB-SC and SCB-SC codes in Figure 7 by about one order of magnitude in the error floor region, compared to codes with the random cutting vector.

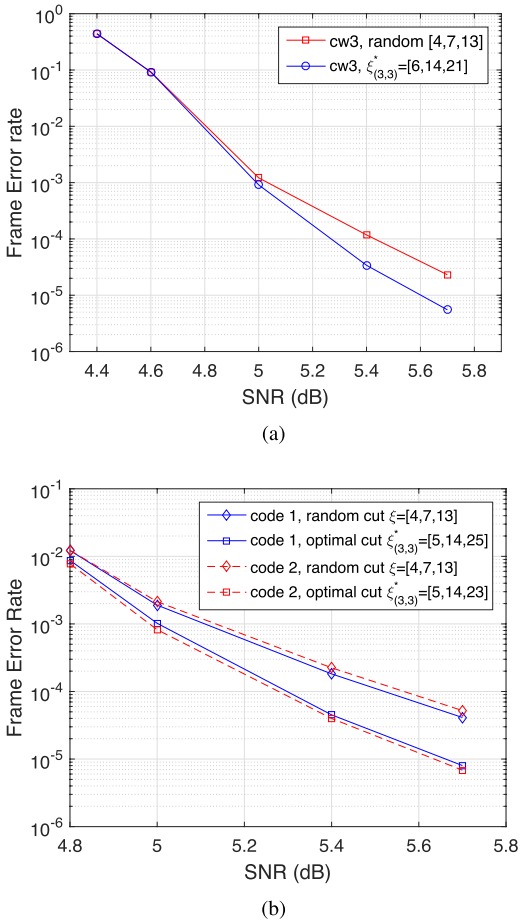


Fig. 7. Performance comparison for binary SC codes constructed by various cutting vectors: (a) Column-weight-3 AB-SC codes and (b) for column-weight-3 SCB-SC codes, code 1 with $f(i) = i^2$, $g(j) = 2j$ and code 2 with $f(i) = i^3$, $g(j) = j$.

Table IV compares the number of (3, 3) binary ASs for the codes shown in Figure 7. Table IV also includes the number of (3, 3) ASs in AB-SC, SCB-SC code 1 and SCB-SC code 2 when cutting vectors $\boldsymbol{\xi} = [4,7,13]$ for $p = 29$ and $\boldsymbol{\xi} = [10,18,56]$ for $p = 67$ are selected at random. It can be observed that the optimal cutting vectors reduce the number

[8]Due to limited computation resources, the simulation results for AB-SC codes presented in this paper are limited to constraint lengths up to 4489 variables nodes. Note that the AS enumeration method and the performance improvement approach presented in this paper are not limited to any specific constraint length and can be applied to longer codes as well.

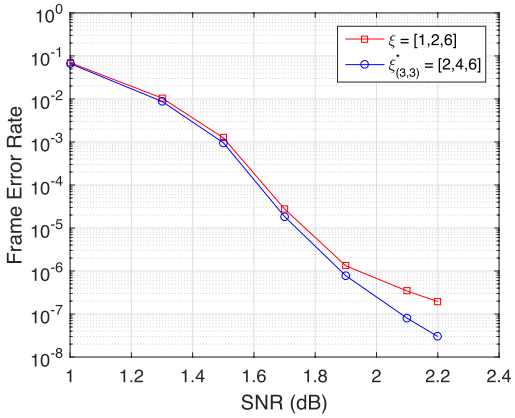


Fig. 8. Performance results for AB-SC codes with $p = 7$, $\gamma = 3$, $L = 500$, and cutting vectors $[1, 2, 6]$ and $[2, 4, 6]$.

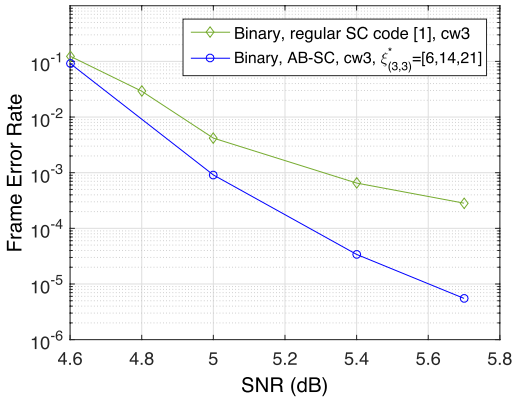


Fig. 9. Performance comparison for optimized binary AB-SC and regular binary SC codes [1].

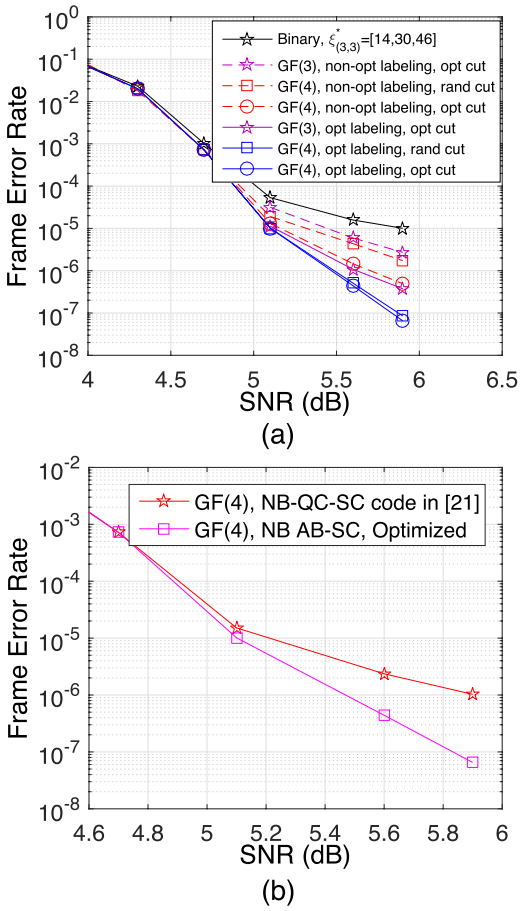


Fig. 10. (a) Performance comparison for optimized and unoptimized non-binary AB-SC codes over $GF(3)$ and $GF(4)$ with $p = 43$, column-weight 3, design rate $R = 0.929$, and cutting vectors $\boldsymbol{\xi}_1 = [10, 22, 34]$ and $\boldsymbol{\xi}_2 = [5, 29, 33]$, where $\boldsymbol{\xi}_1$ is optimized and $\boldsymbol{\xi}_2$ selected randomly, respectively. The binary AB-SC code has constraint length $\nu = 3721$ and design rate $R = 0.95$, and is constructed using an optimal cutting vector for $(3, 3)$ ASs. (b) Performance comparison for optimized non-binary AB-SC and non-binary quasi-cyclic SC codes in [21].

of $(3, 3)$ ASs by more than 12% compared to number of ASs for the randomly selected cutting vectors. This in part explains the superior error floor performance of the codes with optimal cutting vectors.

Figure 8 presents simulation results for binary AB-SC codes with circulant size $p = 7$, column weight $\gamma = 3$, rate $R \approx 0.57$, coupling length $L = 500$, and cutting vectors $\xi^*_{(3,3)} = [2, 4, 6]$ (optimal) and $\xi = [1, 2, 6]$ (random). The code with an optimal cutting vector shows a performance improvement of approximately one order of magnitude compared to the code with a random cutting vector. This result illustrates that our optimization technique is also effective for the design of low-rate AB-SC codes.

Figure 9 provides performance results for randomly-generated regular SC codes [1] with an underlying 90×840 regular code, column weight 3, rate ≈ 0.89, and coupling length $L = 50$. This figure also includes the performance results for AB-SC codes with an optimal cutting vector of $[6, 14, 21]$, constraint length $\nu = 841$, $p = 29$, $L = 50$, and design rate $R = 0.89$. These results show that our optimized AB-SC code have significantly (two orders of magnitude) better error floor performance compared to unoptimized regular SC codes.

For the binary SCB-SC codes, our code design is limited to the choice of the cutting vector, once the underlying block code is specified. In the non-binary case, the choice of the edge labels offers significantly more degrees of freedom that can be exploited in an optimized code design. In fact, our results show that the cutting vector choice is not a critical parameter in the design of non-binary SCB-SC codes since we can remove all non-binary ASs of interest only by manipulating the edge labels.

Figure 10(a) shows the simulation results for non-binary AB-SC codes over $GF(4)$ with constraint length $\nu = 3698$ bits. To construct the non-binary codes, we first form binary AB-SC parity-check matrices with $p = 43$ and column-weight 3 using the optimal cutting vector ($\boldsymbol{\xi}_1 = [10, 22, 34]$) and a randomly chosen cutting vector ($\boldsymbol{\xi}_2 = [5, 29, 33]$). Then, the unoptimized non-binary AB-SC codes are constructed by randomly assigning non-binary edge labels on top of the unlabeled AB-SC Tanner graphs. The optimized codes are constructed by manipulating the edge labels (based on the algorithm addressed in Section V) such that $(3, 3)$, $(4, 2)$,

and $(4, 4)$ non-binary ASs are completely removed from the Tanner graphs of the original codes. Figure 10 shows that our optimized codes have a performance improvement of more than one order of magnitude compared to the original codes. Furthermore, we observe that the performance of the optimized code with a randomly chosen cutting vector is very close to the optimized code with the optimal cutting vector. This observation suggests that the choice of the cutting vector does not significantly affect the performance of our designed non-binary AB-SC codes. The non-binary edge labels offer enough degrees of freedom in codes design such that all problematic small NB-AS can be removed from the designed codes, regardless of the choice of the cutting vector. In the same figure, it is shown that the unoptimized AB-SC code over $GF(3)$ and $GF(4)$ has better frame error rate (FER) performances compared to the binary AB-SC code. An interesting observation is the improved FER performance of the optimized AB-SC code over $GF(3)$ compared to the unoptimized AB-SC code over $GF(4)$, which in combination with its lower decoder complexity, can make the optimized AB-SC code over $GF(3)$ a very attractive candidate for practical implementation. The performance of a binary AB-SC code with $\nu = 3721$ and $p = 61$ constructed using an optimal cutting vector for $(3, 3)$ AS $\boldsymbol{\xi}^{*}_{(3,3)}(61, 50) = [14, 29, 46]$ is also plotted in Figure 10. It can be observed that although the constraint length of the binary AB-SC code and non-binary AB-SC codes are similar, the non-binary codes have superior performance.

Figure 10(b) includes performance comparison for non-binary quasi-cyclic spatially-coupled (NB-QC-SC) code in [21] and optimized non-binary AB-SC codes over $GF(4)$ introduced in this paper. The NB-QC-SC code is constructed based on the "Replicate-and-Mask" approach in [21] with coupling length $L = 50$, column weight 3, constraint length 3658 bits, and circulant size $p = 31$ (i.e., the submatrix-constrained base matrix B is over $GF(2^5)$). Note that the mask matrix provided in [36, eq. (6)] is used to construct the NB-QC-SC code as it guarantees a minimum girth of 6 (based on [21, Th. 2]). The NB-AB-SC code is constructed with comparable constraint length $\nu = 3698$ bits, circulant size 43, and column weight 3. These results show that our designed non-binary codes have superior error floor performance ($\sim$ one order of magnitude) compared to state-of-the-art NB-QC-SC codes.

VII. Conclusion

In this paper, we presented a novel approach to enumerate problematic ASs in SCB-SC codes for transmission over the AWGN channel. In particular, we focused on binary column-weight 3 with a syndrome former memory of one and posed the AS enumeration problem as a problem of counting the number of valid integer points within a two-dimensional area. Using this approach, the exact number of smallest ASs was presented as a function of the code parameters: circulant size, coupling length, and the cutting vector. The resulting closed-form expression was then employed to find the optimal cutting vector, which significantly reduces the number of ASs for any choice of the circulant size in binary SCB-SC codes. For non-binary AB-SC codes we showed that, based on an analytical expression for the average number of $(3, 3)$ non-binary ASs, on average only a ratio of $\frac{1}{q-1}$ of binary ASs in the unlabeled Tanner graph result in problematic non-binary ASs. This result verifies the superior error floor performance of non-binary SC codes compared to their binary counterparts. Finally, we proposed an optimization algorithm which is able to improve the error-floor performance in the non-binary case even further via an iterative manipulation of edge labels in the Tanner graph of the code. We note that the enumeration and characterization of ASs for SCB-SC codes with a syndrome former memory of $m > 1$ will be the subject of future studies. Another interesting future direction includes an error floor analysis and the characterization of ASs for other structured and unstructured spatially-coupled codes.

References

[1] 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. 803–834, Feb. 2011.

[2] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[3] S. Kudekar and K. Kasai, "Threshold saturation on channels with memory via spatial coupling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint Petersburg, Russia, Jul./Aug. 2011, pp. 2562–2566.

[4] S. Kudekar and K. Kasai, "Spatially coupled codes over the multiple access channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint Petersburg, Russia, Jul./Aug. 2011, pp. 2816–2820.

[5] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially-coupled codes on intersymbol-interference channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, Jun. 2012, pp. 2181–2186.

[6] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "New families of LDPC block codes formed by terminating irregular protograph-based LDPC convolutional codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, USA, Jun. 2010, pp. 824–828.

[7] K. Kasai and K. Sakaniwa, "Spatially-coupled MacKay–Neal codes and Hsu–Anastasopoulos codes," *IEICE Trans. Fundam.*, vol. E94-A, no. 11, pp. 2161–2168, Nov. 2011.

[8] M. Lentmaier, G. P. Fettweis, K. S. Zigangirov, and D. J. Costello, Jr., "Approaching capacity with asymptotically regular LDPC codes," in *Proc. IEEE Inf. Theory Appl. (ITA) Workshop*, San Diego, CA, USA, Feb. 2009, pp. 173–177.

[9] D. G. M. Mitchell, A. E. Pusane, and D. J. Costello, Jr., "Minimum distance and trapping set analysis of protograph-based LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 254–281, Jan. 2013.

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

[11] M. Stinner and P. M. Olmos, "Analyzing finite-length protograph-based spatially coupled LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun. 2014, pp. 891–895.

[12] N. Obata, Y.-Y. Jian, K. Kasai, and H. D. Pfister, "Spatially-coupled multi-edge type LDPC codes with bounded degrees that achieve capacity on the BEC under BP decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 2433–2437.

[13] I. Andriyanova and A. Graell i Amat, "Threshold saturation for nonbinary SC-LDPC codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 62, no. 5, pp. 2622–2638, Mar. 2016.

[14] L. Wei, D. G. M. Mitchell, T. E. Fuja, and D. J. Costello, Jr., "Design of spatially coupled LDPC codes over GF(q) for windowed decoding," *IEEE Trans. Inf. Theory*, to be published. [Online]. Available: http://arxiv.org/abs/1411.4373

[15] 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.

[16] D. J. Costello, Jr., L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell, and R. Smarandache, "Spatially coupled sparse codes on graphs: Theory and practice," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 168–176, Jul. 2014.

[17] A. E. Pusane, R. Smarandache, P. O. 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.
[18] S. Johnson and G. Lechner, "Spatially coupled repeat-accumulate codes," *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 373–376, Feb. 2013.
[19] P. M. Olmos, D. G. M. Mitchell, D. Truhachev, and D. J. Costello, Jr., "Improving the finite-length performance of spatially coupled LDPC codes by connecting multiple code chains," *IEEE Trans. Inf. Theory*, 2014.
[20] P. M. Olmos and R. L. Urbanke, "A scaling law to predict the finite-length performance of spatially-coupled LDPC codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3164–3184, Jun. 2015.
[21] K. Liu, M. El-Khamy, J. Lee, I. Kang, and A. Yedla, "Non-binary algebraic spatially-coupled quasi-cyclic LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 511–515.
[22] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.
[23] J. Wang, L. Dolecek, and R. D. Wesel, "The cycle consistency matrix approach to absorbing sets in separable circulant-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2293–2314, Apr. 2013.
[24] D. G. M. Mitchell, L. Dolecek, and D. J. Costello, Jr., "Absorbing set characterization of array-based spatially coupled LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 886–890.
[25] M. Karimi and A. H. Banihashemi, "Efficient algorithm for finding dominant trapping sets of LDPC codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6942–6958, Nov. 2012.
[26] Y. Sun, M. Karkooti, and J. R. Cavallaro, "VLSI decoder architecture for high throughput, variable block-size and multi-rate LDPC codes," in *Proc. IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, USA, May 2007, pp. 2104–2107.
[27] E. M. Kurtas and B. Vasic, Eds., *Advanced Error Control Techniques for Data Storage Systems*. Boca Raton, FL, USA: CRC Press, 2005.
[28] T. Mohsenin, D. N. Truong, and B. M. Baas, "A low-complexity message-passing algorithm for reduced routing congestion in LDPC decoders," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 5, pp. 1048–1061, May 2010.
[29] B. Amiri, A. Reisizadeh, J. Kliewer, and L. Dolecek, "Optimized array-based spatially-coupled LDPC codes: An absorbing set approach," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 51–55.
[30] B. Amiri, J. Kliewer, and L. Dolecek, "Analysis and enumeration of absorbing sets for non-binary graph-based codes," *IEEE Trans. Commun.*, vol. 62, no. 2, pp. 398–409, Feb. 2014.
[31] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7389–7415, Dec. 2014.
[32] B. Amiri, A. Reisizadeh, J. Kliewer, and L. Dolecek. "Optimized design of finite-length spatially-coupled codes: An absorbing set-based analysis," UCLA, Los Angeles, CA, USA, Tech. Rep. [Online]. Available: http://uclacodess.org/papers/TechnicalReport_BA3.pdf
[33] H. Uchikawa, K. Kasai, and K. Sakaniwa, "Terminated LDPC convolutional codes over GF(2^p)," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, Sep./Oct. 2010, pp. 195–200.
[34] V. A. Chandrasetty, S. J. Johnson, and 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.
[35] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. Int. Symp. Turbo Codes Rel. Topics*, Brest, France, Sep. 2000.
[36] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2277–2292, Apr. 2013.
[37] E. Ankan, N. Hassan, M. Lentmaier, G. Montorsi, and J. Sayir, "Challenges and some new directions in channel coding," *J. Commun. Netw.*, vol. 17, no. 4, pp. 328–338, Aug. 2015.
[38] D. J. Costello, Jr., A. E. Pusane, S. Bates, and K. S. Zigangirov, "A comparison between LDPC block and convolutional codes," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, Feb. 2006.
[39] A. J. 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, Sep. 1999.
[40] K. Zhang, X. Huang, and Z. Wang, "High-throughput layered decoder implementation for quasi-cyclic LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 985–994, Aug. 2009.
[41] C.-W. Sham, X. Chen, F. C. M. Lau, Y. Zhao, and W. M. Tam, "A 2.0 Gb/s throughput decoder for QC-LDPC convolutional codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 7, pp. 1857–1869, Jul. 2013.
[42] Y. Sun, G. Wang, and J. R. Cavallaro, "Multi-layer parallel decoding algorithm and VLSI architecture for quasi-cyclic LDPC codes," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 1776–1779.
[43] M. Karimi and A. H. Banihashemi, "On the girth of quasi-cyclic protograph LDPC codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4542–4552, Jul. 2013.
[44] R. Smarandache and P. O. Vontobel, "Quasi-cyclic LDPC codes: Influence of proto- and Tanner-graph structure on minimum Hamming distance upper bounds," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 585–607, Feb. 2012.
[45] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, Jr., "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.
[46] M. P. C. Fossorier, "Quasicyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 1788–1793, Aug. 2004.
[47] S. Ranganathan, K. Vakilinia, L. Dolecek, D. Divsalar, and R. Wesel, "Some results on spatially-coupled protograph LDPC codes," in *Proc. IEEE Inf. Theory Appl. (ITA) Workshop*, San Diego, CA, USA, Feb. 2016.
[48] D. G. M. Mitchell, R. Smarandache, M. Lentmaier, and D. J. Costello, Jr., "Quasi-cyclic asymptotically regular LDPC codes," in *Proc. IEEE Inf. Theory Workshop*, Dublin, Ireland, Aug./Sep. 2010, pp. 1–5.
[49] A. Sridharan, D. J. Costello, Jr., and R. M. Tanner, "A construction for low density parity check convolutional codes based on quasi-cyclic block codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Lausanne, Switzerland, Jun./Jul. 2002, p. 481.

Behzad Amiri (S'10) received the B.S. degree from the Isfahan University of Technology in 2010, and the M.S. and Ph.D. degrees from the University of California at Los Angeles, in 2012 and 2015, respectively, all in electrical engineering. He is currently with Pure Storage Inc., Mountain View, CA, USA, where he is part of the ash reliability and channel architecture team. His research mainly focuses on graph-based code designs and iterative algorithms.

Amirhossein Reisizadehmobarakeh received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2014, and the M.S. degree in electrical engineering from the University of California at Los Angeles (UCLA), Los Angeles, CA, USA, in 2016, where he is currently pursuing the Ph.D. degree in electrical engineering department. His research interests include information and coding theory, wireless communication, and computer science. He received the Electrical Engineering Department Fellowship from UCLA in 2014 for his outstanding academic achievements. He was also ranked 11th out of over 400,000 participants in Iran's nationwide university entrance exam in 2009.

Homa Esfahanizadeh (S'16) received the M.S. and B.S. degrees from the University of Tehran in 2015 and 2012, respectively. She is currently pursuing the Ph.D. degree with the Electrical Engineering Department, University of California at Los Angeles, CA, USA. She is currently with the Laboratory for Robust Information Systems, and her focus is on coding schemes for modern storage systems. Her research interests include coding theory, signal processing, and graph theory.

Jörg Kliewer (S'97–M'99–SM'04) received the Dipl.-Ing. (M.Sc.) degree in electrical engineering from the Hamburg University of Technology, Hamburg, Germany, in 1993, and the Dr.-Ing. (Ph.D.) degree in electrical engineering from the University of Kiel, Germany, in 1999.

He was a Research Assistant with the University of Kiel from 1993 to 1998, where he was a Senior Researcher and Lecturer from 1999 to 2004. In 2004, he visited the University of Southampton, U.K., for one year, and from 2005 to 2007, he was with the University of Notre Dame, IN, USA, as a Visiting Assistant Professor. From 2007 to 2013 he was with New Mexico State University, Las Cruces, NM, USA, as an Associate Professor. He is currently with the New Jersey Institute of Technology, Newark, NJ, USA, as an Associate Professor. His research interests span coding and information theory, graphical models, and statistical algorithms, which includes applications to networked communication and security, data storage, and biology.

Dr. Kliewer was a recipient of the Leverhulme Trust Award and the German Research Foundation Fellowship Award in 2003 and 2004, respectively. He was an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 to 2014. Since 2015, he has served as an Area Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS. He is also a member of the Editorial Board of the IEEE INFORMATION THEORY SOCIETY NEWSLETTER since 2012.

Lara Dolecek (S'05–M'10–SM'13) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering and computer sciences, and the M.A. degree in statistics, from the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA. She is an Associate Professor with the Electrical Engineering Department, University of California at Los Angeles (UCLA), Los Angeles, CA, USA. Her research interests span coding and information theory, graphical models, statistical algorithms, and computational methods, with applications to emerging systems for data storage, processing, and communication. She received the 2007 David J. Sakrison Memorial Prize for the most outstanding doctoral research in the Department of Electrical Engineering and Computer Sciences, UC Berkeley. Prior to joining UCLA, she was a Post-Doctoral Researcher with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology. She received the IBM Faculty Award (2014), the Northrop Grumman Excellence in Teaching Award (2013), the Intel Early Career Faculty Award (2013), the University of California Faculty Development Award (2013), the Okawa Research Grant (2013), the NSF CAREER Award (2012), and the Hellman Fellowship Award (2011). With her research group and collaborators, she received the best paper award from the IEEE Globecom 2015 conference, and two 2016 Best-of-Selse awards. She currently serves as an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS. She has served as a consultant for a number of companies specializing in data communications and storage.