UNIVERSITY OF CALIFORNIA

Los Angeles

# Spatially-Coupled Codes for Modern Data Storage Systems

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical and Computer Engineering

by

Homa Esfahanizadeh

2019

ABSTRACT OF THE DISSERTATION

Spatially-Coupled Codes for Modern Data Storage Systems

by

Homa Esfahanizadeh
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2019
Professor Lara Dolecek, Chair

The volume of data continues to rapidly grow as information pours from various platforms. The huge amount of data needs to be transferred and stored with extremely high reliability. The error correcting codes (ECCs) are an integral part of modern-day communication, computation, and data storage systems in order to safeguard data against the adverse effects of noise and interference. The spatially-coupled (SC) codes are a class of graph-based ECCs that have recently emerged as an excellent choice for error correction in modern data storage and communication due to their outstanding performance, low decoding latency, and simple implementation.

An SC code is constructed by coupling several instances of a block code into a single coupled chain. In the asymptotic limit of large code lengths, SC codes enjoy capacity achieving performance. Due to simplifying assumptions and averaging effects, results from the asymptotic domain are not readily translatable to the practical, finite-length setting. Despite this chasm, finite-length analysis of SC codes is still largely unexplored. We tackle the problem of finite-length optimized design of SC codes in the context of various channel models.

First, we present a systematic framework with low computational complexity for designing finite-length SC codes with superior error floor performance. Next, we tailor our design method for various channel models by targeting the combinatorial objects in the graph of SC codes that are detrimental over these settings. Then, we generalize our framework for

the finite-length analysis and design of irregular SC codes. Finally, we increase the coupling dimensionality, and we present a novel systematic framework to efficiently connect several SC codes and construct multi-dimensional spatially-coupled (MD-SC) codes.

In this research, we use advanced mathematical techniques from algebra, combinatorics, graph theory, probability theory, and optimization theory to develop algorithms and design frameworks with affordable complexity. Our frameworks are especially beneficial for modern storage applications, e.g. magnetic-recording and Flash memories.

The dissertation of Homa Esfahanizadeh is approved.

Dariush Divsalar

Jonathan Kao

Ali Mosleh

Guy Van den Broeck

Lara Dolecek, Committee Chair

University of California, Los Angeles

2019

*To my parents . . .*

*—Fatemeh Alvandi and Yadollah Esfahanizadeh—*

*for their endless love, support, and encouragement*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# ACKNOWLEDGMENTS

The last couple of years was not only a part of my graduate education, but more importantly, it was an era where I became stronger as I moved to the US from my homeland country. I feel extremely happy for everything that I learnt in this journey. Reaching this point was not possible without the priceless help and support of my family, my advisor, my collaborators, and my friends. I would like to express my sincere gratitude at the beginning of my dissertation which includes the results of my academic efforts during these years.

First, I wish to thank my advisor, Professor Lara Dolecek, for giving me the opportunity to come to UCLA and the honor to be her Ph.D. student. She has a massive technical knowledge and insightful information about the cutting edge research problems in the field. Her remarkable success as a female scientist has always been inspiring for me. She generously shared her wisdom with me, encouraged me to be creative, and guided me to develop my professional network and gain experience. I improved my research skills and extended my technical knowledge under her supervision which will impact my future career forever.

I am grateful for the support and thoughtful feedback from my other dissertation committee members: Professor Dariush Divsalar, Professor Guy Van den Broeck, Professor Ali Mosleh, and Professor Jonathan Kao. I would like to say special thanks to Dr. Divsalar for the interesting technical discussions that we had during my Ph.D. education. I would also like to thank the ECE Graduate Office, particularly Deeona Columbia and Ryo Arreola, who have been doing a great job helping graduate students in the department. I also respectfully acknowledge the help and support of colleagues and friends at Samsung where I did an internship.

I would like to thank Professor Yuval Cassuto from Technion - Israel Institute of Technology, Professor Joerg Kliewer from New Jersey Institute of Technology, Dr. Rick Galbraith from Western Digital Company, and Dr. Idan Goldenberg from SanDisk whom I have been fortunate to work with and learn from. It has been a great pleasure to collaborate with Dr. Ahmed Hareedy, Eshed Ram, Ruiyi Wu, Lev Tauz, Amirhossein Reisizadeh, Dr. Yuta

Toriyama, Siyi Yang, Andrew Tan, and Jose Carmona. I had the most collaborations with Dr. Ahmed Hareedy. He was like a mentor for me when I started my Ph.D, we published several papers in prestigious venues, and we won Memorable Paper Award together.

I would also like to thank my former and current lab-mates whom I have not had the chance to do research with but have shared tons of good memories and interesting discussions. They are Dr. Clayton Schoeny, Dr. Frederic Sala, Zehui Chen, Shahroze Kabir, in addition to our lab visitor Professor Laura Conde. Although I did not directly collaborate with Dr. Schoeny, Dr. Sala, and Professor Conde, they have always supported me and generously provided invaluable advice whenever I reached out to them. The time I spent with my collaborators and my lab-mates will always remain in my memory.

Moreover, I would like to wholeheartedly express my gratitude to my friends at UCLA, outside UCLA, and in my home country. First, I am grateful to Shaghayegh Mardani and Nazanin Farahpour. They are the very first people that I met when I came to UCLA, and we have been close friends ever since. We shared a million happy and sad moments. Besides, I cannot be thankful enough to my friends Hanieh Hashemi, Ehsan Abbasi, Ghazaal Ershadi, Aysan Rangchian, Farhad Shahmohammadi, and Saliha Yaylaci for a million invaluable memories that we made together and for always being there for me. Among my friends that I have known before coming to UCLA, I want to especially thank Parvin Taheri, Zeinab Kashani, Parivash Taremi, and Fahimeh Arab, whom I always feel connected to regardless of our physical distance.

Above all, I would like to sincerely thank my precious family. I am eternally grateful for my mother, Fatemeh Alvandi, and my father, Yadollah Esfahanizadeh, who have always been there for me and supported me to overcome the challenges and obstacles. I am very thankful to my kind and smart sister, Leila, who has always been a role model for me in my life and my education. I am grateful to my older brother, Mojtaba, and my younger brother, Amir Mohammad, for their limitless love and unconditional friendship. I am thankful to my uncle, Professor Mostafa Esfahanizadeh, who have always encouraged me through my academic journey and supported me in my life. I had to live far away from my family during my Ph.D., and it was very difficult for me not to be able to visit them. However, there was not a single

moment that I did not feel their love. I strongly believe I could not reach this point without their support and backing.

My research was sponsored in part by grants from the Advanced Storage Technology Consortium - the International Disk Drive Equipment and Material Association (ASTC-IDEMA). It was supported in part by the National Science Foundation (NSF) grants CCF 1718369, BSF 1718389, CCF-CAREER 1150212, BSF 1718389, CCF 1150212, along with grants from Western Digital Corporation (WDC). My research was also supported by the UCLA Dissertation Year Fellowship in my last year.

# VITA

| | |
|---|---|
| 2012 | Bachelor of Science, Electrical Engineering, Telecommunications, University of Tehran, Iran. |
| 2012-2014 | Teaching Assistant, University of Tehran, Iran. |
| 2015 | Master of Science, Electrical Engineering, Communication Systems, University of Tehran, Iran. |
| 2018 | Internship, Error Control Coding, Samsung Semiconductor Inc., San Diego, USA. |
| 2018 | Memorable Paper Award, Non-Volatile Memories Workshop, University of California, San Diego, USA. |
| 2018—2019 | Dissertation Year Fellow, University of California, Los Angeles, USA. |
| 2017, 2019 | Teaching Assistant, University of California, Los Angeles, USA. |
| 2017 | Ph.D. Candidate, Electrical and Computer Engineering, Signals and Systems, University of California, Los Angeles, USA. |

SELECTED PUBLICATIONS

**H. Esfahanizadeh**, L. Tauz, and L. Dolecek, "Multi-dimensional spatially-coupled code design: Enhancing the cycle properties," *IEEE Transactions on Communications* (under review), 2019.

**H. Esfahanizadeh**, A. Hareedy, and L. Dolecek,"Finite-length construction of high performance spatially-coupled codes via optimized partitioning and lifting," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 3-16, Jan. 2019.

**H. Esfahanizadeh**, A. Hareedy, R. Wu, R. Galbraith, and L. Dolecek, "Spatially-coupled codes for channels with SNR variation," *IEEE Transactions on Magnetics*, vol. 54, no. 11, pp. 1-5, Nov. 2018.

**H. Esfahanizadeh**, A. Hareedy, and L. Dolecek,"Spatially coupled codes optimized for magnetic recording applications," *IEEE Transactions on Magnetics*, vol. 53, no. 2, pp. 1-11, Feb. 2017.

B. Amiri, A. Reisizadehmobarakeh, **H. Esfahanizadeh**, J. Kliewer, and L. Dolecek, "Optimized design of finite-length separable circulant-based spatially-coupled codes: An absorbing set-based analysis," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4029-4043, Oct. 2016.

# CHAPTER 1

# Introduction and Preliminaries

All modern dense storage systems and communication applications require increasingly high levels of reliability which highlights the necessity of advanced error correcting codes (ECCs) for protecting data against noise and interference. The low-density parity-check (LDPC) codes, also known as graph-based codes, are first introduced by Gallager in his doctoral dissertation [1]. However, due to the complexity of their encoding/decoding, these codes were forgotten until their rediscovery in [2]. The LDPC codes can be fully characterized by either their sparse bipartite graphs, known as Tanner graphs, or by their sparse parity-check matrices. Nowadays, graph-based ECCs are extremely popular due to their excellent performance in a variety of applications and for their simple implementations.

Spatially-coupled (SC) are a class of graph-based codes that have recently emerged as an excellent choice for error correction due to their capacity approaching performance and low-latency decoding [3, 4]. An SC code is constructed by coupling several instances of an LDPC code into a single coupled chain. In the asymptotic limit of large code lengths, SC codes enjoy capacity achieving performance, and many elegant results have been derived in this setting, including threshold saturation, e.g., [5–7]. While the asymptotic analysis is important, due to simplifying assumptions, e.g., being cycle-free, and averaging effects, results from the asymptotic domain are not readily translatable to the practical, finite-length setting.

Despite this chasm, finite-length analysis of SC codes is still largely unexplored. There are several works that have studied the finite-length analysis and design of SC codes with emphasis on the error floor region, e.g., [8–11]. These works, while promising, have some limitations which will be discussed in more details in Section 1.3. We tackle the problem

of finite-length design of regular and irregular SC codes in the context of various channel models, including additive white Gaussian noise (AWGN), Flash, magnetic recording (MR), etc.

We focus on circulant-based (CB) SC codes because they offer simpler hardware implementation. For the rest of this dissertation, CB SC codes are recalled as SC codes for brevity. From an algebraic viewpoint, an SC code is constructed by partitioning the circulants in the parity-check matrix of a block code into several component matrices and piecing copies of the component matrices in a diagonal structure to construct the final SC code. For SC codes, "how to partition the underlying block code" and "how to adjust the circulant powers" highly influence the finite-length performance. By a judicious choice of these parameters, we develop a systematic framework with a reasonable computational complexity to design SC codes that show orders of magnitudes performance improvement in the error floor area compared to un-optimized constructions [12].

Moreover, we extensively investigate combinatorial structures in the graphical representation of the graph-based codes that are detrimental in the error floor area. We show that the type and configuration of problematic objects depend on the channel model, and thus, a channel-aware code design is required to achieve a good error-correction performance. We then adjust our framework to optimize SC code parameters to reduce the population of these problematic objects [13, 14].

We also investigate the channel non-uniformity and the solutions to alleviate it. The signal to noise ratio (SNR) is not constant over different parts of a storage device. We present an SC code construction scheme that provides a well-defined cooperation among different parts of the channel such that reliable data can help unreliable ones to be recovered, and consequently, to improve the overall error-correction performance. To this end, we present an SC code design along with a specific interleaving scheme that improve the robustness against the channel non-uniformity [15].

Next, we present a new methodology for analysis and design of finite-length irregular SC codes. While SC codes intrinsically possess a small amount of node degree irregularity

due to the termination effects, which aides in performance improvement, it is nonetheless customary to use regular underlying block to construct finite-length SC codes. We present a new scheme for constructing irregular SC codes with a superior performance in the error floor area. We show that this scheme has significantly better performance than random SC code constructions with the same node degree distribution [16].

Finally, we increase the coupling dimensionality, and we present a novel systematic framework to efficiently connect several SC codes and construct multi-dimensional spatially-coupled (MD-SC) codes. An efficient multi-dimensional coupling is very helpful to attain a better error-correction capability, especially for modern storage applications such as two-dimensional magnetic-recording (TDMR) and multi-layer Flash memories. Our well-designed MD-SC codes have a very low population of short cycles in their graphs. According to the simulation results, our MD-SC codes demonstrate dramatically lower error rates than their one-dimensional counterparts [17].

## 1.1 Structure of SC Codes

The SC codes are graph-based codes constructed by coupling together a series of disjoint block LDPC codes into a single coupled chain [3, 4]. This operation can be viewed as partitioning (also known as edge-spreading) the parity-check matrix $\mathbf{H}$ of a block code into component matrices $\mathbf{H}_y$, where $y \in \{0, 1, \ldots, m\}$, and piecing $L$ copies of the component matrices together to obtain the parity-check matrix $\mathbf{H}_{\text{SC}}$ of an SC code, as shown in Figure 1.1. The parameters $m$ and $L$ are called the memory and coupling length, respectively. Here, as the underlying block codes, we consider CB codes, where all circulants are non-zero [18].

A non-binary (NB) block/SC LDPC code defined over GF($q$), $q > 2$, has its parity-check matrix entries drawn from GF($q$). In the associated Tanner graph, edges exist between a variable node (VN) and a check node (CN) iff the corresponding entry in the parity-check matrix is non-zero. This non-zero value is called the *weight* of that edge. The *unlabeled graph* (*unlabeled parity-check matrix*) is generated by replacing all edge weights in the Tanner graph (all non-zero entries in the parity-check matrix) by 1s.

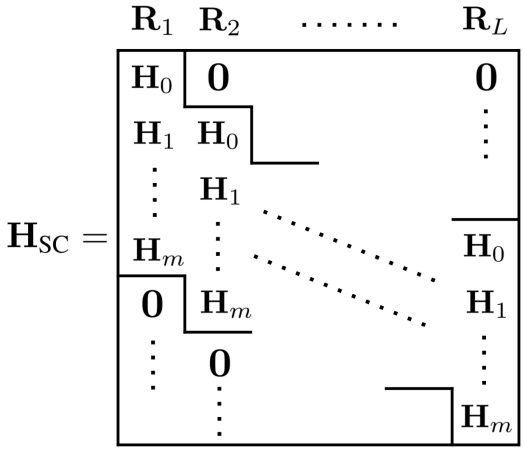



Figure 1.1: The parity-check matrix of an SC code with parameters $m$ and $L$.

### 1.1.1 Circulant-Based SC Codes

Throughout this dissertation, each column (resp., row) in a parity-check matrix corresponds to a VN (resp., CN) in the equivalent graph of the matrix. CB codes are a class of structured regular $(\gamma, \kappa)$ LDPC codes, where $\gamma$ is the column weight of the parity-check matrix, i.e., VN degree in the graph, and $\kappa$ is the row weight i.e., CN degree. CB codes offer simple hardware implementation thanks to their structure [18]. Suppose $z$ is the size of the constituent circulants, called circulant size. The parity-check matrix $\mathbf{H}$ of a CB code is constructed as follows:

$$\mathbf{H} = \begin{bmatrix} \sigma^{f_{0,0}} & \sigma^{f_{0,1}} & \dots & \sigma^{f_{0,\kappa-1}} \\ \sigma^{f_{1,0}} & \sigma^{f_{1,1}} & \dots & \sigma^{f_{1,\kappa-1}} \\ \vdots & \vdots & \dots & \vdots \\ \sigma^{f_{\gamma-1,0}} & \sigma^{f_{\gamma-1,1}} & \dots & \sigma^{f_{\gamma-1,\kappa-1}} \end{bmatrix}. \tag{1.1}$$

Each circulant has the form $\sigma^{f_{i,j}}$ where $i$, $0 \leq i \leq \gamma-1$, is the row group index, $j$, $0 \leq j \leq \kappa-1$, is the column group index, and $\sigma$ is the $z \times z$ identity matrix cyclically shifted one unit to the left. The term $f_{i,j}$ specifies the power of the circulant at row group index $i$ and column group index $j$. A circulant power $f_{i,j}$ has non-negative integer value, and $\sigma^0$ is the identity matrix of size $z \times z$. For example, the choice of $f_{i,j} = ij$, $\kappa = z$, and $z$ prime results in the class of array-based (AB) codes [19]. We use CB codes as the underlying block codes

of SC codes. We highlight that, in this dissertation, each circulant in (1.1) is a permutation of a single identity matrix, i.e., each circulant has weight 1. Circulants with larger weights have a negative impact on the girth [20], and we do not use them in our code construction.

SC codes have parity-check matrices with a band-diagonal structure. A CB SC code is constructed by partitioning the $\kappa\gamma$ circulants in the parity-check matrix $\mathbf{H}$ of a block code into component matrices $\mathbf{H}_y$, $0 \leq y \leq m$, where $m$ is referred to as the memory. Each component matrix $\mathbf{H}_y$ has the same size as $\mathbf{H}$. A component matrix contains a subset of circulants in $\mathbf{H}$, and the rest of its elements are zero. Every circulant in $\mathbf{H}$ is assigned to exactly one of the component matrices, and $\sum_{y=0}^{m} \mathbf{H}_y = \mathbf{H}$. Given the component matrices and the coupling length $L$, one can construct $\mathbf{H}_{\text{SC}}$ as shown in Figure 1.1.

The design rate for a CB code with parameters $\gamma$ and $\kappa$ is,

$$\mathbf{r}_{\text{d}} = 1 - \frac{\gamma}{\kappa}, \tag{1.2}$$

and the design rate for the corresponding SC code with parameters $m$ and $L$ is equal to,

$$\mathbf{r}_{\text{d}}^{\text{SC}} = 1 - \frac{(L+m)\gamma}{L\kappa}.$$

The rate loss due to spatial coupling is inversely proportional to the coupling length $L$. In the asymptotic regime, as $L \to \infty$, the design rate for an SC code and its underlying block code are equal.

Even for the case of regular underlying block code $\mathbf{H}$, SC codes have a right amount of CN degree irregularity that gives them performance advantage: At the point where two adjacent parity-check matrices are stitched together to form the parity-check matrix of the overall SC code, the CNs have connectivity that span VNs of both subgraphs. However, the CNs at the beginning and at the end of this multi block concatenation have lower degrees than the rest of the CNs to allow for termination. This setup allows for information propagation where one block, after it has been successfully decoded, aids its neighbor in decoding. Additionally, this block-convolutional structure is amenable for low-complexity windowed decoding [4].

**Definition 1.** *Consider an SC code with parameters $z$, $\kappa$, $\gamma$, $m$, and $L$. The replica $\mathbf{R}_r$, $r \in \{1, \ldots, L\}$, is a collection of columns in the matrix $\mathbf{H}_{\text{SC}}$ and is defined as:*

$$\mathbf{R}_r = \mathbf{H}_{\text{SC}}[0 : (L+m)\gamma z - 1][(r-1)\kappa z : r\kappa z - 1]. \tag{1.3}$$

The $L$ replicas, from left to right, are denoted by $\mathbf{R}_1, \ldots, \mathbf{R}_L$. Figure 1.1 illustrates the replicas on the parity-check matrix $\mathbf{H}_{\text{SC}}$ of an SC code. The notation $\mathbf{H}_{\text{SC}}[\rho_1 : \rho_2][\nu_1 : \nu_2]$ refers to a submatrix of $\mathbf{H}_{\text{SC}}$ with rows $\{\rho_1, \rho_1 + 1, \ldots, \rho_2\}$ and columns $\{\nu_1, \nu_1 + 1, \ldots, \nu_2\}$. In this dissertation, we are interested in the time-invariant SC codes, where the non-zero part of $\mathbf{R}_r$ is the same for any $r \in \{1, \ldots, L\}$:

$$\mathbf{R}_r[(r-1)\gamma z : (r+m)\gamma z - 1][0 : \kappa z - 1] = [\mathbf{H}_0^{\text{T}}\ \mathbf{H}_1^{\text{T}} \ldots \mathbf{H}_m^{\text{T}}]^{\text{T}}. \tag{1.4}$$

## 1.2 Combinatorial Objects of Interest

Under iterative decoding algorithms, certain structures in the graph of LDPC codes cause the error floor phenomenon. These structures are error-prone, and the errors resulting from them are not necessarily codeword errors. Let $q$ be the size of the finite field that the LDPC code is defined over. We review the key definitions here:

**Definition 2.** *Consider a subgraph induced by a subset $\mathcal{V}$ of VNs in the Tanner graph of an LDPC code. Set all the VNs in $\mathcal{V}$ to values in $\text{GF}(q)\backslash\{0\}$ and set all other VNs to 0. The set $\mathcal{V}$ is said to be an $(a, b)$ trapping set (TS) if the size of $\mathcal{V}$ is $a$ and the number of neighboring unsatisfied CNs is $b$ (for the given assignment of VN values) [21, 22].*

We note that in the binary case ($q = 2$), unsatisfied CNs are odd degree CNs connected to $\mathcal{V}$ in the induced subgraph, and satisfied CNs are even degree CNs connected to $\mathcal{V}$ in the induced subgraph.

**Definition 3.** *Consider a subgraph induced by a subset $\mathcal{V}$ of VNs in the Tanner graph of an LDPC code. Set all the VNs in $\mathcal{V}$ to values in $\text{GF}(q)\backslash\{0\}$ and set all other VNs to 0. The set*

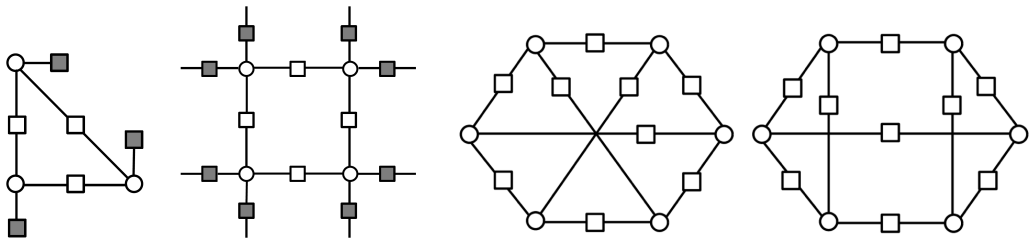



Figure 1.2: (a) The $(3,3)$ AS. (b) The $(4,8)$ TS. (c) The two non-isomorphic configurations for $(6,0)$ BAS.

*$\mathcal{V}$ is said to be an $(a,b)$ absorbing set (AS) if the size of $\mathcal{V}$ is $a$, the number of neighboring unsatisfied CNs is $b$, and each VN in $\mathcal{V}$ is connected to strictly more satisfied than unsatisfied neighboring CNs (for the given assignment of VN values) [23].*

The class of TSs subsumes the class of ASs. While TSs that are not ASs are usually harmless (these configurations are typically unstable under iterative decoding), we purposely recall the definition here. The reason is, as we see later, a systematic elimination of multiple problematic ASs can be achieved by focusing on the elimination of the common subgraph that these ASs share. In some cases, these common structures are certain TSs that on their own do not appear as decoding errors.

It is recently observed that in the context of LDPC codes for the MR applications, only certain subclasses of ASs matter [24]. The reason is that in the MR setting, the global iterations between the detector and the decoder can often (with a sufficient number of outer iterations) overcome the errors due to ASs that are on the verge of instability, and what remains then are the errors due to a certain subclass of ASs, which are called *balanced absorbing sets (BASs)*. The ASs are thus classified as either *balanced absorbing sets (BASs)* or *unbalanced absorbing sets (UBASs)*. Both are defined as follows:

**Definition 4.** *Let $g = \left\lfloor \frac{\gamma-1}{2} \right\rfloor$ for a given column weight $\gamma$. An $(a,b)$ AS with $0 \leq b \leq \left\lfloor \frac{ag}{2} \right\rfloor$ is defined as a balanced absorbing set (BAS), while an $(a,b)$ AS with $\left\lfloor \frac{ag}{2} \right\rfloor < b \leq ag$ is defined as an unbalanced absorbing set (UBAS) [24].*

For instance, for the binary case ($q = 2$), Figure 1.2 (a) depicts a $(3,3)$ AS, Figure 1.2 (b) depicts a $(4,8)$ TS which cannot be categorized as an AS, and Fig. 1.2 (c) shows two non-

isomorphic configurations for $(6, 0)$ BAS. In the graphical representation of problematic objects, VNs, satisfied CNs, and unsatisfied CNs are represented by white circles, white squares, and grey squares, respectively.

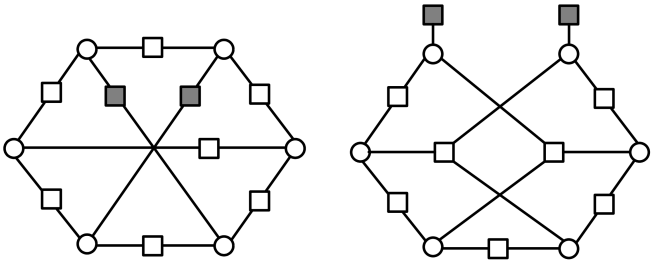
The NB LDPC codes offer superior performance over the binary LDPC codes, and are thus are more suitable for modern Flash memories. It was recently revealed that general absorbing sets of type two (GASTs) are the most dominant problematic object in the error floor of NB LDPC codes over practical, inherently asymmetric, Flash channels [25, 26], which is not similar to the case of canonical channels. The definitions of GASTs and unlabeled GASTs (UGASTs) are:

**Definition 5.** *Consider a subgraph induced by a subset $\mathcal{V}$ of VNs in the Tanner graph of an LDPC code. Set all the VNs in $\mathcal{V}$ to values in $\mathrm{GF}(q)\backslash\{0\}$ and set all other VNs to 0. The set $\mathcal{V}$ is said to be an $(a, b, d_1, d_2, d_3)$ GAST over $\mathrm{GF}(q)$ if the size of $\mathcal{V}$ is $a$, the number of unsatisfied CNs connected to $\mathcal{V}$ is $b$, the number of degree-1 (resp., 2 and $> 2$) CNs connected to $\mathcal{V}$ is $d_1$ (resp., $d_2$ and $d_3$), $d_2 > d_3$, all the unsatisfied CNs connected to $\mathcal{V}$ (if any) have either degree 1 or degree 2, and each VN in $\mathcal{V}$ is connected to strictly more satisfied than unsatisfied neighboring CNs (for the given assignment of VN values) [26].*

**Definition 6.** *Consider a subgraph induced by a subset $\mathcal{V}$ of VNs in the unlabeled Tanner graph of an LDPC code. Let $\mathcal{O}$ (resp., $\mathcal{T}$ and $\mathcal{H}$) be the set of degree-1 (resp., 2 and $> 2$) CNs connected to $\mathcal{V}$. This graphical configuration is an $(a, d_1, d_2, d_3)$ unlabeled GAST (UGAST) if it satisfies the following two conditions [26]:*

1. *$|\mathcal{V}| = a$, $|\mathcal{O}| = d_1$, $|\mathcal{T}| = d_2$, $|\mathcal{H}| = d_3$, and $d_2 > d_3$.*
2. *Each VN in $\mathcal{V}$ is connected to strictly more neighbors in $\{\mathcal{T} \cup \mathcal{H}\}$ than in $\mathcal{O}$.*

For instance, Figure 1.3 (a) depicts a $(6, 2, 0, 9, 0)$ GAST and Figure 1.3 (b) depicts a $(6, 2, 2, 5, 2)$ GAST [26]. Their unlabeled configurations are $(6, 0, 9, 0)$ UGAST and $(6, 2, 5, 2)$ UGAST, respectively. For the NB setup, the GASTs, BASs, ASs, and TSs of interest are described in terms of both the underlying topology (which is the unlabeled configuration)



(a) (b)

Figure 1.3: (a) The $(6, 2, 0, 9, 0)$ GAST. (b) The $(6, 2, 2, 5, 2)$ GAST. Appropriate NB edge weights are assumed.

and the edge weights. Eliminating structures with a given topology thus eliminates all NB configurations that have that topology.

## 1.3 Previous Works on Finite-Length SC Codes

There are several works that have studied the finite-length analysis and design of SC codes with emphasis on the error floor region. In [8] and [9], authors consider SC codes with memory $m = 1$ and focus on the restricted scheme of partitioning by cutting vectors. In [10], a construction method is presented for the class of AB SC codes with column weight $\gamma = 3$ and for different memories. In [11], a systematic partitioning scheme is introduced to reduce the population of cycles with length 4, i.e., cycles-4. Minimizing the number of cycles-4 is very costly to be addressed in the partitioning step. In [11], a heuristic algorithm for lifting is also presented to improve the girth properties in the final code. However, the algorithm does not incorporate the repetitive structure of SC codes, and it requires a large circulant size to ensure a good performance.

The cutting vector (CV) scheme for constructing CB SC codes was previously proposed in [8] and [9]. In this scheme, the underlying block code is partitioned via a so-called cutting vector $\boldsymbol{\zeta} = [\zeta_0\ \zeta_1\ \ldots\ \zeta_{\gamma-1}]$ into component matrices $\mathbf{H}_0$ and $\mathbf{H}_1$. The cutting vector $\boldsymbol{\zeta}$ is a vector of ascending natural numbers. Matrix $\mathbf{H}_0$ is constructed by copying all circulants of $\mathbf{H}$ with row and column group indices in $\{(i, j) \mid j < \zeta_i\}$ to the same coordinates in $\mathbf{H}_0$, and setting all remaining elements of $\mathbf{H}_0$ to 0. Matrix $\mathbf{H}_1$ is then simply $\mathbf{H}_1$=$\mathbf{H}$−$\mathbf{H}_0$. The CV

partitioning scheme can be generalized to construct SC codes with higher memories by using several cutting vectors.

Among the previous works in the literature, the CV partitioning is one of the most general scheme for constructing SC codes with various underlying block codes and column weights, and it has been widely used for constructing finite-length SC codes. Thus, in the rest of this dissertation, we compare the performance of our well-designed SC codes with SC codes that have similar parameters and are constructed by the CV scheme.

## 1.4 Outline of Contributions

### 1.4.1 Chapter 2 Contributions

In this chapter, we first introduce a general approach for the enumeration of combinatorial objects in the graph of finite-length SC codes. Our approach is general in the sense that it effectively works for SC codes with various partitioning schemes, column weights, and memories. Next, we present a two-stage framework for the construction of high performance SC codes optimized for AWGN channels. In the presented framework, we aim at minimizing the number of combinatorial objects that are detrimental in the error floor region.

In the first stage, we deploy a novel partitioning scheme, called the optimal overlap (OO) partitioning, to produce the optimal partitioning that corresponds to the smallest number of detrimental objects in the protograph. In the second stage, we apply a new circulant power optimizer (CPO) to further reduce the number of detrimental objects in the lifted graph.

### 1.4.2 Chapter 3 Contributions

In this chapter, we customize the stages of our SC design framework for constructing finite-length SC codes suitable for non-AWGN channels. We consider three different channel models, i.e., MR channels, Flash channels, and AWGN channels with SNR variation. We tackle the problem of optimized design of (NB) SC codes in the context of these channel models. For this purpose, we identify combinatorial structures in the graphical representation of the SC

codes that are detrimental in each channel setting.

An intriguing observation is that for the same SC code, the problematic objects for the MR channels and Flash channels are combinatorially different from AWGN setting, thus necessitating a careful code design approach for these applications. We then carefully identify the partitioning, circulant powers, and edge weights (for NB codes) in order to construct SC codes with a low population of problematic objects in their graphs in a systematic, low-complexity, and channel-aware fashion.

In MR systems, consecutive sections may experience different SNRs. To perform error correction over MR systems or similar non-uniform applications, one approach is to use an individual block code for each section. However, the performance over a section affected by a lower SNR is weaker compared to the performance over a section affected by a higher SNR. We present an SC code construction for channels with SNR variation. We then introduce a low-complexity interleaving scheme specific to SC codes that further improves their performance over channels with SNR variation.

### 1.4.3 Chapter 4 Contributions

It has long been known that irregular graph-based codes offer performance advantage over their regular counterparts. In this chapter, we present a novel combinatorial framework for designing finite-length irregular SC codes. Our irregular SC codes have the desirable properties of regular SC codes thanks to their structure while offering significant performance benefits that come with the node degree irregularity. Coding constructions proposed in this work contribute to the existing portfolio of finite-length graph-based LDPC code designs.

### 1.4.4 Chapter 5 Contributions

In this chapter, we present a systematic framework for constructing multi-dimensional SC (MD-SC) codes with notably better cycle properties than their one-dimensional counterparts. In our framework, the multi-dimensional coupling is performed via an informed relocation of problematic circulants. This work is general in the terms of the number of constituent SC

codes that are connected together, the number of neighboring SC codes that each constituent SC code is connected to, and the length of the cycles whose populations we aim to reduce. The results of this work can be particularly beneficial in data storage systems, e.g., two-dimensional magnetic-recording and multi-layer Flash memories, as high performance MD-SC codes are robust against various channel impairments and non-uniformity.

# CHAPTER 2

# Finite-Length Construction of High Performance SC Codes

## 2.1 Introduction

In this chapter, we propose a new combinatorial framework for the finite-length analysis and design of CB SC codes. We aim at constructing SC codes with the minimum number of problematic objects in their graphs. These problematic objects are certain configurations that depend on both the code specifications as well as the channel model [21, 23, 24, 26]. We first introduce a new enumeration approach that exploits the structure of SC codes in order to efficiently enumerate the combinatorial objects of interest. Our new approach can be applied to SC codes constructed by arbitrary partitioning and memory size.

Next, we present a systematic scheme for partitioning the underlying block code and constructing SC codes with a superior performance for AWGN channels. We operate on the protograph of the SC codes, and express the number of problematic objects we want to minimize in terms of the overlap parameters, which characterize the partitioning. Then, we solve a discrete optimization problem to determine the optimal overlap parameters. We call this new partitioning scheme the *optimal overlap (OO)* partitioning.

The OO partitioning scheme is, in particular, suitable for the code optimization in the regime outside the reach of brute force methods, since it finds the optimal partitioning in a systematic way and does not need a search among a possibly very large set of choices. We demonstrate that the new scheme achieves much better performance compared to the existing scheme of partitioning by cutting vectors (CV) [8, 9]. More importantly, our parti-

tioning scheme is presented for general memory $m$ and column weight $\gamma$. Given the optimal partitioning, we then apply a new heuristic program to adjust the circulant powers to further reduce the number of problematic objects in the final graphs. We call this heuristic program the *circulant power optimizer (CPO).*

The rest of the chapter is organized as follows. In Section 2.2, we present our general enumeration approach. In Section 2.3, we propose our two-stage framework for constructing finite-length time-invariant SC codes. More Specifically, Section 2.3.1 and Section 2.3.2 are devoted to the OO partitioning and the CPO algorithm, respectively, as two stages of our proposed framework. Our simulation results are given in Section 2.4. Finally, the conclusion appears in Section 2.5.

## 2.2 Enumeration of Combinatorial Objects

In this section, we introduce our new approach to enumerate combinatorial objects in the Tanner graph of SC codes. This approach can be applied to SC codes with any underlying block code, partitioning choice, memory, and column weight. Our main result is stated in Theorem 1. We first state the necessary auxiliary results.

**Remark 1.** *Each VN corresponds to one unique column and each CN corresponds to one unique row in the parity-check matrix. We thus say that an object "exists in the matrix" and "exists in the graph" of the code interchangeably. Besides, Theorem 1 and the preceding results are presented for $(a, b)$ ASs/TSs as the objects of interest. However, they can be fairly utilized when cycles, BASs, and UGASTs are targeted as these objects are subclasses of ASs/TSs.*

Let the shortest path that connects any two VNs of an $(a, b)$ AS/TS include at most $\lambda$ VNs (including the two VNs themselves). Given the configuration of an AS/TS, one can find the parameter $\lambda$ by known methods, e.g., Dijkstra's algorithm [27]. In the case that there exists at least one cycle that spans all VNs, $\lambda$ is upper-bounded by $\left\lfloor \frac{a}{2} \right\rfloor + 1$. We say two VNs are *adjacent* if they are connected via a CN. For example, a $(3, 3(\gamma - 2))$ AS/TS has $\lambda = 2$, because any two VNs are adjacent, see Figure 1.2 (a). The $(4, 8)$ TS shown in Figure 1.2 (b)

has $\lambda = 3$.

**Lemma 1.** *For an SC code with memory $m$, all VNs of an $(a, b)$ AS/TS belong to at most $\chi$ consecutive replicas, where*

$$\chi = (\lambda - 1)m + 1. \tag{2.1}$$

*Proof.* As shown in Figure 1.1, the maximum number of consecutive replicas with the property that their non-zero parts have some rows (CNs) in common is $(m + 1)$. As a result, any two adjacent VNs must be within at most $(m + 1)$ consecutive replicas, or equivalently, there are at most $(m - 1)$ different replicas between the replicas in which two adjacent VNs exist. Consider an $(a, b)$ AS/TS, and let $v_1$ be the VN with the lowest index (the index of the corresponding column in $\mathbf{H}_{\text{SC}}$), and $v_f$ be the VN with the highest index. These two VNs are connected on a path that includes at most $\lambda$ VNs. There are at most $(m - 1)$ different replicas between the replicas of any two adjacent VNs on this path, and the $\lambda$ VNs belong to at most $\lambda$ different replicas. Consequently, $v_1$ and $v_f$ must belong to at most $\chi$ consecutive replicas, where $\chi$ is given by: $\chi = (\lambda - 1)(m - 1) + \lambda = (\lambda - 1)m + 1$. The rest of VNs of the AS/TS must also belong to these $\chi$ consecutive replicas since they have indices between $v_1$ and $v_f$. □

**Definition 7.** *The matrix $\mathbf{\Pi}_r^k$, where $r \in \{1, \ldots, L\}$ and $k \in \{1, \ldots, L-r+1\}$, is a submatrix of $\mathbf{H}_{\text{SC}}$, and is defined as:*

$$\begin{aligned} &\mathbf{\Pi}_r^k = \mathbf{H}_{\text{SC}}[u_{r,1}^k : u_{r,2}^k][v_{r,1}^k : v_{r,2}^k], \\ &u_{r,1}^k = (r-1)\gamma z, \quad u_{r,2}^k = (r + m + k - 1)\gamma z - 1, \\ &v_{r,1}^k = (r-1)\kappa z, \quad v_{r,2}^k = (r + k - 1)\kappa z - 1. \end{aligned} \tag{2.2}$$

We say an $(a, b)$ AS/TS *starts* in $\mathbf{R}_r$ if among all its VNs, the one with the lowest associated column index belongs to $\mathbf{R}_r$

**Lemma 2.** *For an $(a, b)$ AS/TS that starts in replica $\mathbf{R}_r$ and spans $k$ replicas, all VNs and all their neighboring CNs have corresponding row and column indices in $\mathbf{\Pi}_r^k$.*

*Proof.* An AS/TS that starts in replica $\mathbf{R}_r$ and spans $k$ replicas must have its VNs in the replicas $\{\mathbf{R}_r, \ldots, \mathbf{R}_{r+k-1}\}$. Based on (1.3) and (1.4), the smallest submatrix of $\mathbf{H}_{\text{SC}}$ that spans all non-zero parts of these $k$ replicas is $\mathbf{\Pi}_r^k$. □

**Theorem 1.** *Consider an SC code with parameters $m$ and $L$. Let $F$ be the total number of $(a, b)$ ASs/TSs, and $F_r^k$ be the number of $(a, b)$ ASs/TSs that start in $\mathbf{R}_r$ and span $k$ consecutive replicas, $k \in \{1, 2, \ldots, \chi\}$. Then,*

$$F = \sum_{k=1}^{\chi} (L - k + 1) F_1^k. \tag{2.3}$$

*Proof.* By summing up the number of $(a, b)$ ASs/TSs over all possible starting replicas and spanning sizes, the total number of $(a, b)$ ASs/TSs can be written as:

$$F = \sum_{k=1}^{\chi} \sum_{r=1}^{L-k+1} F_r^k. \tag{2.4}$$

According to Lemma 2, $F_r^k$ is equivalent to the number of objects of interest that span $k$ replicas in $\mathbf{\Pi}_r^k$. Consider the matrix $\mathbf{\Pi}_{r+1}^k$, $r \in \{1, \ldots, L - k\}$:

$$\begin{aligned}
&\mathbf{\Pi}_{r+1}^k = \mathbf{H}_{\text{SC}}[u_{r+1,1}^k : u_{r+1,2}^k][v_{r+1,1}^k : v_{r+1,2}^k], \\
&u_{r+1,1}^k = u_{r,1}^k + \gamma z, \quad u_{r+1,2}^k = u_{r,2}^k + \gamma z, \\
&v_{r+1,1}^k = v_{r,1}^k + \kappa z, \quad v_{r+1,2}^k = v_{r,2}^k + \kappa z.
\end{aligned}$$

Because of the repetitive structure of SC codes (see also Figure 1.1), the following equality holds for the parity-check matrix $\mathbf{H}_{\text{SC}}$:

$$\mathbf{H}_{\text{SC}}[i + \gamma z][j + \kappa z] = \mathbf{H}_{\text{SC}}[i][j].$$

Then,

$$\begin{aligned}
\mathbf{\Pi}_{r+1}^k &= \mathbf{H}_{\text{SC}}[u_{r,1}^k + \gamma z : u_{r,2}^k + \gamma z][v_{r,1}^k + \kappa z : v_{r,2}^k + \kappa z] \\
&= \mathbf{H}_{\text{SC}}[u_{r,1}^k : u_{r,2}^k][v_{r,1}^k : v_{r,2}^k] = \mathbf{\Pi}_r^k.
\end{aligned}$$

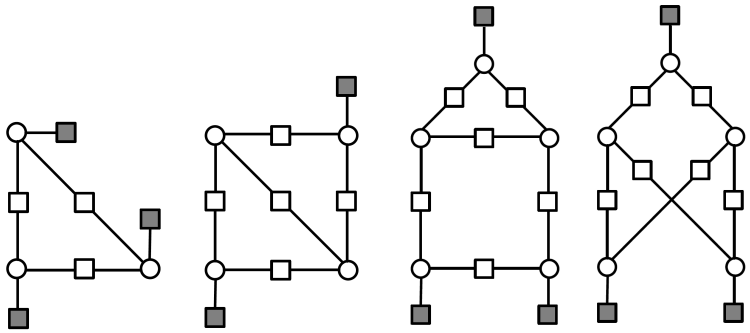



Figure 2.1: (a) The $(3,3)$ AS. (b) The $(4,2)$ AS. (c) Two non-isomorphic configurations for the $(5,3)$ AS.

Consequently,

$$\mathbf{\Pi}_{r+1}^{k} = \mathbf{\Pi}_{r}^{k} \Longrightarrow F_{r+1}^{k} = F_{r}^{k}.$$

By means of induction, we infer that:

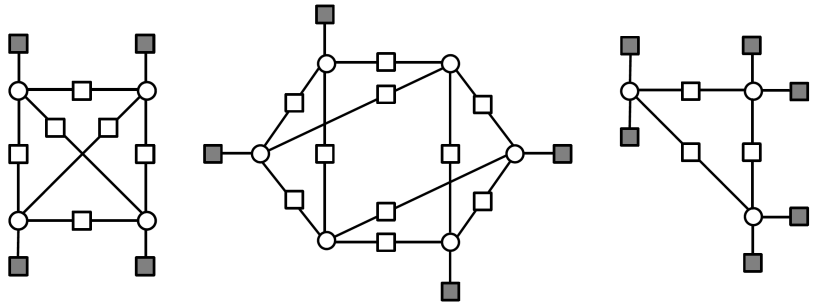
$$F_{r}^{k} = F_{1}^{k} \qquad \forall r \in \{1, 2, \ldots, L-k+1\}. \tag{2.5}$$

Combining (2.4) and (2.5), yields the final result in (2.3). □

The utility of Theorem 1 is to significantly reduce the search size by searching over $\mathbf{\Pi}_{1}^{\chi}$ rather than $\mathbf{H}_{\text{SC}}$. The number of problematic objects that span the first $k$ replicas in $\mathbf{\Pi}_{1}^{\chi}$, i.e., $F_{1}^{k}$, can be computed via an exhaustive search. In Section 2.3.1, we present a new scheme to efficiently find $F_{1}^{k}$ for the protograph of SC codes (SC codes with $z = 1$).

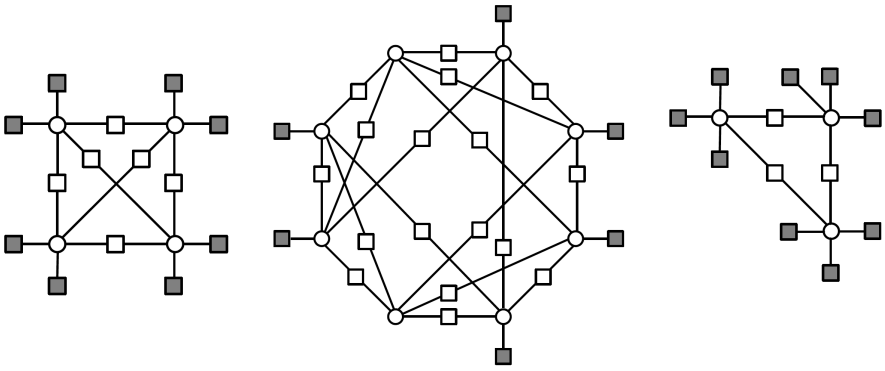
## 2.3 Two-Stage Framework for Constructing SC Codes for AWGN Channels

For CB codes with $z \geq \kappa$, by a careful choice of the circulant powers, e.g., AB where $f_{i,j} = ij$, it is easy to achieve zero cycles-4. Therefore, we consider CB codes having girth 6 in our analysis in this chapter. For CB codes with girth 6 simulated over AWGN channels, certain types of ASs are dominant in the error floor region. For $\gamma = 3$ codes, the $(3,3)$, $(4,2)$, and



(a) (b) (c)

Figure 2.2: (a) The $(4, 4)$ AS. (b) One configuration for the $(6, 4)$ AS. (c) The $(3, 6)$ TS as the common denominator.



(a) (b) (c)

Figure 2.3: (a) The $(4, 8)$ AS. (b) One configuration for the $(8, 6)$ AS. (c) The $(3, 9)$ TS as the common denominator.

$(5, 3)$ ASs are the dominant objects [9, 28], see Figure 2.1. For $\gamma = 4$ codes, the dominant objects are the $(4, 4)$ and $(6, 4)$ ASs [9, 28]. There are several non-isomorphic configurations for the $(6, 4)$ AS. The configuration of the $(4, 4)$ AS and one of the configurations of the $(6, 4)$ AS are illustrated in Figure 2.2. For $\gamma = 5$ codes, the dominant objects are the $(4, 8)$ and $(8, 6)$ ASs [28]. The configuration of the $(4, 8)$ AS and one of the configurations of the $(8, 6)$ AS are illustrated in Figure 2.3.

According to Figures 2.1-2.3, our extensive simulations, and the literature [9, 28], the $(3, 3(\gamma - 2))$ ASs/TSs appear as subgraphs of dominant problematic configurations for CB codes with column weight $\gamma$. We call the $(3, 3(\gamma - 2))$ AS/TS, i.e., cycle-6, the *common denominator*. For $\gamma = 3$, 4, and 5, the common denominators are the $(3, 3)$ AS, the $(3, 6)$ TS, and the $(3, 9)$ TS, respectively. In order to design high performance SC codes, we seek to

reduce the number of dominant ASs. To efficiently perform this task, we aim at minimizing the number of common denominator instances.

By minimizing the population of the common denominator instances as subgraphs, we reduce the number of all supergraphs and improve the code performance. Moreover, the common denominator has a simpler combinatorial characteristics, and thus it is easier to locate and operate on. Removing all the $(3, 3(\gamma - 2))$ ASs/TSs as instances of the common denominator is not feasible for many practical code parameters. Consequently, removing as many as possible of the $(3, 3(\gamma - 2))$ ASs/TSs is the ultimate goal in our SC code construction. This optimization results in a dramatic performance improvement as also verified by our simulation results.

In this section, we present a two-stage framework to design high performance SC codes suitable for AWGN channels.

1. In the first stage, we operate on the protograph of the SC code, and express the number of cycles-6 in terms of the overlap parameters, which characterize the partitioning of the block code. Then, we solve a discrete optimization problem to determine the optimal overlap parameters. We call this new partitioning scheme the OO partitioning.

2. In the second stage, given the optimal partitioning, we apply a new heuristic program to optimize the circulant powers in order to further reduce the number of cycles-6 in the graph of the SC code. This heuristic program is called CPO.

The next two subsections describe these stages in details.

### 2.3.1 Stage 1: Optimal Overlap Partitioning

Given a fixed set of code parameters, the partitioning provides an extra degree of freedom to construct SC codes with a lower population of problematic objects, e.g., cycles, ASs, etc. We note that for many practical settings, constructing an underlying block code with a given girth (resp., zero population of certain problematic objects) is fairly difficult, if not unfeasible. Via the proposed partitioning, we can reduce the number of smallest cycles (resp.,

problematic objects) as much as possible for an SC code.

The protograph matrix of a CB code is obtained by replacing each $z \times z$ non-zero circulant with 1 and each $z \times z$ zero circulant with 0. The protograph matrices of $\mathbf{H}$, $\mathbf{H}_0$, $\mathbf{H}_1$, ..., $\mathbf{H}_m$ are $\mathbf{H}^{\mathrm{p}}$, $\mathbf{H}_0^{\mathrm{p}}$, $\mathbf{H}_1^{\mathrm{p}}$, ..., $\mathbf{H}_m^{\mathrm{p}}$, respectively (all of size $\gamma \times \kappa$). The protograph matrix of $\mathbf{H}_{\mathrm{SC}}$ is $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$, and it is of size $(L+m)\gamma \times L\kappa$. This $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$ also has $L$ replicas, but with $1 \times 1$ circulants. The procedure of generating $\mathbf{H}_{\mathrm{SC}}$ from $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$ is called lifting.

In this subsection, we formulate the problem of identifying the optimal partitioning that results in the minimum number of cycles-6 in $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$ as an optimization problem over a set of integer-valued parameters, which we call independent non-zero overlap parameters. The new optimization problem has a dramatically smaller size compared to the original one that operates over all possible partitioning options. This novel combinatorial scheme is called the OO partitioning. In the OO partitioning, the resulting component matrices do not necessarily each comprise a contiguous set of circulants. This property is in contrast with the scheme of partitioning with cutting vectors in which component matrices – by design – have large overlaps [8, 9], which is an undesirable feature for the finite-length design, as we show later.

Although there are many instances of the cycle-4 in the protograph of an SC code, the goal in the OO partitioning of $\mathbf{H}^{\mathrm{p}}$ is minimizing the population of cycles-6 in $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$. This is because cycles-4 are typically easy to be all removed from the lifted graph of $\mathbf{H}_{\mathrm{SC}}$ by a careful choice of the circulant size and the circulant power arrangement. A cycle-6 in $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$, which is defined by the non-zero entries $\{(h_1, l_1), (h_1, l_2), (h_2, l_2), (h_2, l_3), (h_3, l_3), (h_3, l_1)\}$ in $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$, results in $z$ cycles-6 in $\mathbf{H}_{\mathrm{SC}}$ if and only if [29, 30]:

$$f_{h_1,l_1} + f_{h_2,l_2} + f_{h_3,l_3} = f_{h_1,l_2} + f_{h_2,l_3} + f_{h_3,l_1} \pmod{z}, \tag{2.6}$$

where $f_{h,l}$ is the power of the circulant with row group index $h$ and column group index $l$. Otherwise, this cycle results in zero cycles-6 in $\mathbf{H}_{\mathrm{SC}}$ [29, 30]. Moreover, a cycle-6 in the final (lifted) graph of an SC code can only be generated from a cycle-6 in the protograph.

Motivated by the above facts, the OO partitioning aims at deriving the overlap parameters of $\mathbf{H}^{\mathrm{p}}$ that result in the minimum number of cycles-6 in the graph of $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$. Then, in the next

subsection, we introduce CPO to further reduce the number cycles-6 in the graph of $\mathbf{H}_{\text{SC}}$ by breaking the condition in (2.6) for as many cycles-6 in the optimized graph of $\mathbf{H}_{\text{SC}}^{\text{p}}$ as possible. We establish a discrete optimization problem by expressing the number of cycles-6 in the graph of $\mathbf{H}_{\text{SC}}^{\text{p}}$ as a function of the overlap parameters and standard code parameters. We first introduce the overlap parameters.

**Definition 8.** *Let $\mathbf{\Pi}_1^{1,\text{p}}$ of size $(m+1)\gamma \times \kappa$ be the protograph matrix of $\mathbf{\Pi}_1^1 = [\mathbf{H}_0^{\text{T}} \ \dots \ \mathbf{H}_m^{\text{T}}]^{\text{T}}$. A degree-$d$ overlap among $d$ rows of $\mathbf{\Pi}_1^{1,\text{p}}$ indexed by $\{i_1, \dots, i_d\}$ is defined as a position (column) in which all these rows have 1$s$ simultaneously. A degree-$d$ overlap parameter $t_{\{i_1,\dots,i_d\}}$ is defined as the number of degree-$d$ overlaps among the rows indexed by $\{i_1, \dots, i_d\}$ in $\mathbf{\Pi}_1^{1,\text{p}}$. A degree-1 overlap parameter $t_{i_1}$ is defined as the number of 1$s$ in row $i_1$ of $\mathbf{\Pi}_1^{1,\text{p}}$.*

Remarks 2 and 3 discuss the properties of the overlap parameters in Definition 8.

**Remark 2.** *For an SC code with column weight $\gamma$, the maximum degree for an overlap parameter with non-zero value is $\gamma$. There are exactly $\gamma$ 1$s$ in any column of $\mathbf{\Pi}_1^{1,\text{p}}$, thus there is no position in a set of $d > \gamma$ rows such that all the rows have 1$s$ simultaneously.*

**Remark 3.** *Consider a set of rows $\{i_1, \dots, i_d\}$ of $\mathbf{\Pi}_1^{1,\text{p}}$ and $d > 1$. If there is at least one pair of distinct row indices $(i_u, i_v)$ such that $i_u, i_v \in \{i_1, \dots, i_d\}$ and $i_u = i_v \pmod{\gamma}$, then $t_{\{i_1,\dots,i_d\}} = 0$. This property holds because the matrix $\mathbf{H}^{\text{p}}$ is partitioned into $\mathbf{H}_0^{\text{p}}, \dots, \mathbf{H}_m^{\text{p}}$. Thus, there is zero overlap between the rows with the same indices in the component (protograph) matrices.*

Based on Definition 8, Remark 2, and Remark 3, the set of all non-zero overlap parameters is:

$$\begin{aligned} \mathcal{O} = \{t_{\{i_1,\dots,i_d\}} \mid 1 \le d \le \gamma, 0 \le i_1, \dots, i_d < (m+1)\gamma, \\ \forall \{i_u, i_v\} \subset \{i_1, \dots, i_d\} \ i_u \neq i_v \pmod{\gamma}\}. \end{aligned} \tag{2.7}$$

**Example 1.** *For an SC code with $\gamma = 3$ and $m = 1$, there are 26 non-zero overlap parameters,*

*and the set of non-zero overlap parameters is:*

$$\begin{aligned}\mathcal{O} &= \{t_{\{i_1,\ldots,i_d\}} \mid 1 \le d \le 3, 0 \le i_1,\ldots,i_d < 6,\\ &\qquad \forall\{i_u,i_v\} \subset \{i_1,\ldots,i_d\}\ i_u \neq i_v \pmod 3\}\\ &= \{t_0,t_1,t_2,t_3,t_4,t_5,t_{\{0,1\}},t_{\{0,2\}},t_{\{0,4\}},t_{\{0,5\}},t_{\{1,2\}},t_{\{1,3\}},t_{\{1,5\}},t_{\{2,3\}},t_{\{2,4\}},\\ &\qquad t_{\{3,4\}},t_{\{3,5\}},t_{\{4,5\}},t_{\{0,1,2\}},t_{\{0,1,5\}},t_{\{0,2,4\}},t_{\{0,4,5\}},t_{\{1,2,3\}},t_{\{1,3,5\}},t_{\{2,3,4\}},t_{\{3,4,5\}}\}.\end{aligned}$$

The overlap parameters in (2.7) are not independent. In fact, some overlap parameters are linear combinations of other overlap parameters. Lemma 3 introduces the independent non-zero overlap parameters. As we see later in this subsection, the number of cycles-6 can be expressed in terms of the overlap parameters. Thus, the significance of Lemma 3 is to reduce the complexity of the discrete optimization problem that specifies the optimal values for the overlap parameters, and consequently, the optimal partitioning.

**Lemma 3.** *The set of all independent non-zero overlap parameters is:*

$$\begin{aligned}\mathcal{O}_{\text{ind}} = \{t_{\{i_1,\ldots,i_d\}} \mid 1 \le d \le \gamma, 0 \le i_1,\ldots,i_d < m\gamma,\\ \forall\{i_u,i_v\} \subseteq \{i_1,\ldots,i_d\}\ i_u \neq i_v \pmod \gamma\}.\end{aligned} \tag{2.8}$$

*The overlap parameters that are not included in $\mathcal{O}_{\text{ind}}$ are either zero or linear functions of the overlap parameters in $\mathcal{O}_{\text{ind}}$. Let $0 \le i_1,\ldots,i_{d_1} < m\gamma$, $m\gamma \le j_1,\ldots,j_{d_2} < (m+1)\gamma$, and $1 \le (d_1+d_2) \le \gamma$. Then,*

$$t_{\{i_1,\ldots,i_{d_1},j_1,\ldots,j_{d_2}\}} = \quad t_{\mathcal{I}} + \sum_{\alpha=1}^{d_2} (-1)^{\alpha} \sum_{\substack{\{j'_1,\ldots,j'_\alpha\}\subset\mathcal{J}\\ [x_1\ \ldots\ x_\alpha]\in\{0,\ldots,m-1\}^\alpha}} t_{\mathcal{I}\cup\{x_1\gamma+\overline{j'_1},\ldots,x_\alpha\gamma+\overline{j'_\alpha}\}}, \tag{2.9}$$

*where $\mathcal{I} = \{i_1,\ldots,i_{d_1}\}$, $\mathcal{J} = \{j_1,\ldots,j_{d_2}\}$, $\overline{j} = (j \bmod \gamma)$, and in the case of $\mathcal{I} = \varnothing$, $t_{\mathcal{I}} = \kappa$. We note that row $x\gamma + \overline{j}$, where $x \in \{0,\ldots,m-1\}$ and $j \in \mathcal{J}$, belongs to the $x$'th component matrix of $\mathbf{\Pi}_1^{1,\text{p}}$, and corresponds to row $j$ of $\mathbf{\Pi}_1^{1,\text{p}}$.*

*Proof.* From Definition 8, $t_{\{i_1,\ldots,i_{d_1},j_1,\ldots,j_{d_2}\}}$ is the number of overlaps (column indices in which all the rows $\{i_1,\ldots,i_{d_1},j_1,\ldots,j_{d_2}\}$ in $\mathbf{\Pi}_1^{1,\text{p}}$ have 1s simultaneously).

- $\mathcal{I} \neq \varnothing$: To have an overlap at position (column index) $y \in \{1, \dots \kappa\}$ among the rows $\mathcal{I} \cup \mathcal{J}$ of $\mathbf{\Pi}_1^{1,\mathrm{p}}$: 1) the rows in $\mathcal{I}$ of $\mathbf{\Pi}_1^{1,\mathrm{p}}$ must have 1s at position $y$, 2) all rows in the first $m$ component matrices of $\mathbf{\Pi}_1^{1,\mathrm{p}}$ corresponding to the rows in $\mathcal{J}$ must have 0s at position $y$. In other words, the rows $\{x_1\gamma + \overline{j_1}, \dots, x_{d_2}\gamma + \overline{j_{d_2}}\}$ of $\mathbf{\Pi}_1^{1,\mathrm{p}}$ must have 0s at position $y$, where $[x_1 \ \dots \ x_{d_2}] \in \{0, \dots, m-1\}^{d_2}$. Aided by the principle of inclusion-exclusion:

$$t_{\mathcal{I}\cup\mathcal{J}} = t_{\mathcal{I}} + \sum_{\alpha=1}^{d_2} (-1)^{\alpha} \sum_{\substack{\{j'_1,\dots,j'_\alpha\}\subset\mathcal{J} \\ [x_1 \ \dots \ x_\alpha]\in\{0,\dots,m-1\}^\alpha}} t_{\mathcal{I}\cup\{x_1\gamma+\overline{j'_1},\dots,x_\alpha\gamma+\overline{j'_\alpha}\}}.$$

- $\mathcal{I} = \varnothing$: To have an overlap at position (column index) $y \in \{1, \dots \kappa\}$ among the rows $\mathcal{J}$ of $\mathbf{\Pi}_1^{1,\mathrm{p}}$, all rows in the first $m$ component matrices of $\mathbf{\Pi}_1^{1,\mathrm{p}}$ corresponding to the rows in $\mathcal{J}$ must have 0s at position $y$. In other words, the rows $\{x_1\gamma + \overline{j_1}, \dots, x_{d_2}\gamma + \overline{j_{d_2}}\}$ of $\mathbf{\Pi}_1^{1,\mathrm{p}}$ must have 0s at position $y$, where $[x_1 \ \dots \ x_{d_2}] \in \{0, \dots, m-1\}^{d_2}$. Aided by the principle of inclusion-exclusion:

$$t_{\mathcal{J}} = \kappa + \sum_{\alpha=1}^{d_2} (-1)^{\alpha} \sum_{\substack{\{j'_1,\dots,j'_\alpha\}\subset\mathcal{J} \\ [x_1 \ \dots \ x_\alpha]\in\{0,\dots,m-1\}^\alpha}} t_{\{x_1\gamma+\overline{j'_1},\dots,x_\alpha\gamma+\overline{j'_\alpha}\}}.$$

□

**Remark 4.** *The overlap parameters are defined over a subset of rows in the matrix* $\mathbf{\Pi}_1^{1,\mathrm{p}} = [(\mathbf{H}_0^{\mathrm{p}})^{\mathrm{T}} \dots (\mathbf{H}_m^{\mathrm{p}})^{\mathrm{T}}]^{\mathrm{T}}$. *Lemma 3 states that an overlap parameter defined over a set of rows which includes some of the rows of* $\mathbf{H}_m^{\mathrm{p}}$ *can be written as a linear combination of the overlap parameters that are defined over sets of rows that do not include the rows of* $\mathbf{H}_m^{\mathrm{p}}$.

**Example 2.** *For an SC code with* $m = 1$ *and* $\gamma = 3$,

$$\mathcal{O}_{\mathrm{ind}} = \{t_0, t_1, t_2, t_{\{0,1\}}, t_{\{0,2\}}, t_{\{1,2\}}, t_{\{0,1,2\}}\}.$$

*According to (2.9), the overlap parameters that are not in* $\mathcal{O}_{\mathrm{ind}}$ *are functions of the* 7 *overlap parameters in* $\mathcal{O}_{\mathrm{ind}}$ *as follows:*

$$
\begin{aligned}
&t_3 = \kappa - t_0, && t_{\{3,4\}} = \kappa - t_0 - t_1 + t_{\{0,1\}},\\
&t_4 = \kappa - t_1, && t_{\{3,5\}} = \kappa - t_0 - t_2 + t_{\{0,2\}},\\
&t_5 = \kappa - t_2, && t_{\{4,5\}} = \kappa - t_1 - t_2 + t_{\{1,2\}},\\
&t_{\{0,4\}} = t_0 - t_{\{0,1\}}, && t_{\{0,1,5\}} = t_{\{0,1\}} - t_{\{0,1,2\}},\\
&t_{\{0,5\}} = t_0 - t_{\{0,2\}}, && t_{\{0,2,4\}} = t_{\{0,2\}} - t_{\{0,1,2\}},\\
&t_{\{1,3\}} = t_1 - t_{\{0,1\}}, && t_{\{0,4,5\}} = t_0 - t_{\{0,1\}} - t_{\{0,2\}} + t_{\{0,1,2\}},\\
&t_{\{1,5\}} = t_1 - t_{\{1,2\}}, && t_{\{1,2,3\}} = t_{\{1,2\}} - t_{\{0,1,2\}},\\
&t_{\{2,3\}} = t_2 - t_{\{0,2\}}, && t_{\{1,3,5\}} = t_1 - t_{\{0,1\}} - t_{\{1,2\}} + t_{\{0,1,2\}},\\
&t_{\{2,4\}} = t_2 - t_{\{1,2\}}, && t_{\{2,3,4\}} = t_2 - t_{\{0,2\}} - t_{\{1,2\}} + t_{\{0,1,2\}},\\
&t_{\{3,4,5\}} = \kappa - t_0 - t_1 - t_2 + t_{\{0,1\}} + t_{\{0,2\}} + t_{\{1,2\}} - t_{\{0,1,2\}}.
\end{aligned}
$$

**Lemma 4.** *The number of independent non-zero overlap parameters described in Lemma 3 is:*

$$\mathcal{N}_{\text{ind}} = |\mathcal{O}_{\text{ind}}| = (m+1)^\gamma - 1. \tag{2.10}$$

*Proof.* According to (2.8), $|\mathcal{O}_{\text{ind}}|$ is the number of non-empty subsets of the set $S = \{0, 1, \ldots, m\gamma - 1\}$ with maximum size $\gamma$ such that no two elements in a subset have the same value mod $\gamma$. We first partition the set $S$ as follows:

$$
\begin{aligned}
S = \{0, \gamma, \ldots, (m-1)\gamma\} \cup \{1, \gamma+1, \ldots, (m-1)\gamma+1\} \cup \\
\cdots \cup \{\gamma-1, 2\gamma-1, \ldots, m\gamma-1\}.
\end{aligned}
\tag{2.11}
$$

All elements of any partition in (2.11) have the same value mod $\gamma$. Consequently, we have to pick at most one element from each partition to form a subset with the specified characteristics. All the $\gamma$ partitions are of size $m$; thus, the number of ways for choosing at most one element from each partition is $\left(\binom{m}{1} + \binom{m}{0}\right)^\gamma = (m+1)^\gamma$. The number of partitions in (2.11) is $\gamma$. Consequently, all subsets of $S$ that are constructed by choosing at most one element from each partition have maximum size $\gamma$. We only need to exclude the empty subset (corresponding to $d = 0$). As a result, $\mathcal{N}_{\text{ind}} = (m+1)^\gamma - 1$. □

**Example 3.** *The number of independent non-zero overlap parameters for $\gamma = 3$ and $m = 1$ is $|\mathcal{O}_{\text{ind}}| = 7$, while the number of non-zero overlap parameters is $|\mathcal{O}| = 26$ (see Examples 1*

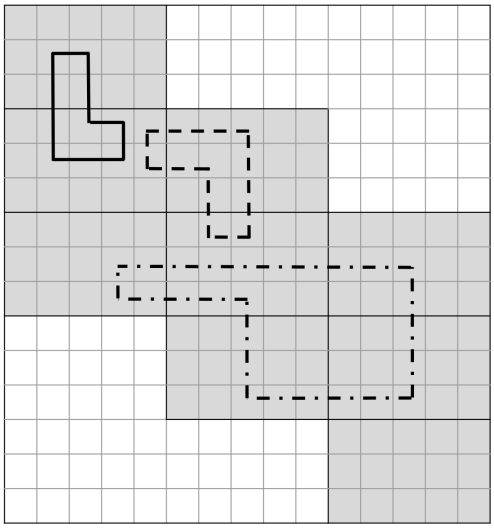


Figure 2.4: Examples of cycles-6 on $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$ with parameters $\kappa = 5$, $\gamma = 3$, $m = 2$, and $L = 3$. The cycles with solid lines, dashed lines, and dashed-dotted lines are spanning one, two, and three replicas, respectively. Component matrices are illustrated in gray.

*and 2). This comparison shows the importance of Lemma 3 in reducing the number of overlap parameters that need to be optimized. This number, as we show later, determines the size of a discrete optimization problem that identifies the optimal partitioning.*

Next, we show that the number of cycles-6 in the protograph of an SC code can be expressed as a function of overlap parameters in $\mathcal{O}_{\mathrm{ind}}$. As we noted, the cycle-6 is the common denominator of the overwhelming majority of problematic objects for CB codes with different column weights over AWGN channels. A cycle-6 is formed of three distinct degree-2 overlaps, and each overlap corresponds to one VN in the graph of the code, see Figure 2.4.

**Lemma 5.** *Consider the protograph of an SC code with parameters $m$, $L$, and $\mathcal{O}$. Let $[x]^+ = \max\{x, 0\}$. The three VNs of a cycle-6 belong to one, two, or three different replicas. We partition cycles-6 with specific CNs into three categories, and enumerate them separately. Let $\mathbf{R}_r$ be the reference replica and $c_1 = (r-1)\gamma + i_1$, $c_2 = (r-1)\gamma + i_2$, and $c_3 = (r-1)\gamma + i_3$ be the CNs.*

1. *The number of cycles-6 with all VNs in one replica, say $\mathbf{R}_r$, and CNs $c_1$, $c_2$, and $c_3$ is:*

$$
\begin{aligned}
&\mathcal{A}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2,i_3\}}) \\
&= \left(t_{\{i_1,i_2,i_3\}}[t_{\{i_1,i_2,i_3\}} - 1]^+[t_{\{i_2,i_3\}} - 2]^+\right) \\
&+ \left(t_{\{i_1,i_2,i_3\}}(t_{\{i_1,i_3\}} - t_{\{i_1,i_2,i_3\}})[t_{\{i_2,i_3\}} - 1]^+\right) \\
&+ \left((t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})t_{\{i_1,i_2,i_3\}}[t_{\{i_2,i_3\}} - 1]^+\right) \\
&+ \left((t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})(t_{\{i_1,i_3\}} - t_{\{i_1,i_2,i_3\}})t_{\{i_2,i_3\}}\right).
\end{aligned} \tag{2.12}
$$

2. *The number of cycles-*6 *with VNs in two replicas, say two VNs in* $\mathbf{R}_r$ *and one VN in* $\mathbf{R}_q$*, and CNs* $c_1$*,* $c_2$*, and* $c_3$ *is:*

$$
\begin{aligned}
&\mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2+(r-q)\gamma,i_3+(r-q)\gamma\}}) \\
&= \left(t_{\{i_1,i_2,i_3\}}[t_{\{i_1,i_3\}} - 1]^+ t_{\{i_2+(r-q)\gamma,i_3+(r-q)\gamma\}}\right) \\
&+ \left((t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})t_{\{i_1,i_3\}}t_{\{i_2+(r-q)\gamma,i_3+(r-q)\gamma\}}\right),
\end{aligned} \tag{2.13}
$$

   *where* $c_2$ *and* $c_3$ *are the CNs connected via the VN that belongs to* $\mathbf{R}_q$*.*

3. *The number of cycles-*6 *with VNs in three replicas, say* $\mathbf{R}_r$ *and* $\mathbf{R}_q$ *and* $\mathbf{R}_s$ *(*$r < q < s$*), and CNs* $c_1$*,* $c_2$*, and* $c_3$ *is:*

$$
\begin{aligned}
&\mathcal{C}(t_{\{i_1,i_2\}}, t_{\{i_1+(r-q)\gamma,i_3+(r-q)\gamma\}}, t_{\{i_2+(r-s)\gamma,i_3+(r-s)\gamma\}}) \\
&= t_{\{i_1,i_2\}}t_{\{i_1+(r-q)\gamma,i_3+(r-q)\gamma\}}t_{\{i_2+(r-s)\gamma,i_3+(r-s)\gamma\}},
\end{aligned} \tag{2.14}
$$

   *where* $c_1$ *and* $c_2$ *are connected via the VN that belongs* $\mathbf{R}_r$*,* $c_1$ *and* $c_3$ *are connected via the VN that belongs to* $\mathbf{R}_q$*, and* $c_2$ *and* $c_3$ *are connected via the VN that belongs to* $\mathbf{R}_s$*.*

*Proof.* A cycle-6 is formed of three distinct degree-2 overlaps as illustrated in Figure 2.4. To avoid over-counting, we must consider degree-3 overlaps in order to guarantee that the overlaps of a cycle-6 that belong to one replica have distinct indices.

1. Because of the structure of SC codes, an overlap between rows $((r-1)\gamma+i_1, (r-1)\gamma+i_2)$ in $\mathbf{R}_r$ corresponds to an overlap between rows $(i_1, i_2)$ in $\mathbf{\Pi}_1^{1,\mathrm{p}}$. Similarly, an overlap between rows $((r-1)\gamma+i_1, (r-1)\gamma+i_3)$ in $\mathbf{R}_r$ corresponds to an overlap between rows $(i_1, i_3)$ in $\mathbf{\Pi}_1^{1,\mathrm{p}}$, and an overlap between rows $((r-1)\gamma+i_2, (r-1)\gamma+i_3)$ in $\mathbf{R}_r$

corresponds to an overlap between rows $(i_2, i_3)$ in $\mathbf{\Pi}_1^{1,\mathrm{p}}$. If $t_{\{i_1,i_2,i_3\}} = 0$, the number of ways we can pick the overlaps is $t_{\{i_1,i_2\}} t_{\{i_1,i_3\}} t_{\{i_2,i_3\}}$. Since we must consider the degree-3 overlap among the rows $i_1$, $i_2$, and $i_3$ of $\mathbf{\Pi}_1^{1,\mathrm{p}}$, we partition the enumeration into the following four cases:

- The overlap between rows $(i_1, i_2)$ is chosen from $t_{\{i_1,i_2,i_3\}}$ overlaps among the three rows, and the overlap between rows $(i_1, i_3)$ is chosen from other $(t_{\{i_1,i_2,i_3\}} - 1)$ overlaps among the three rows (if possible; that is why we use $[t_{\{i_1,i_2,i_3\}} - 1]^+$ ).
- The overlap between rows $(i_1, i_2)$ is chosen from $t_{\{i_1,i_2,i_3\}}$ overlaps among the three rows, and the overlap between rows $(i_1, i_3)$ is chosen from $(t_{\{i_1,i_3\}} - t_{\{i_1,i_2,i_3\}})$ overlaps that are exclusively between these two rows.
- The overlap between rows $(i_1, i_2)$ is chosen from $(t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})$ overlaps that are exclusively between these two rows, and the overlap between rows $(i_1, i_3)$ is chosen from $t_{\{i_1,i_2,i_3\}}$ overlaps among the three rows.
- The overlap between rows $(i_1, i_2)$ is chosen from $(t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})$ overlaps that are exclusively between these two rows, and the overlap between rows $(i_1, i_3)$ is chosen from $(t_{\{i_1,i_3\}} - t_{\{i_1,i_2,i_3\}})$ overlaps that are exclusively between these two rows.

These four cases correspond to the four terms in (2.12).

2. Because of the structure of SC codes, an overlap between rows $((r-1)\gamma+i_1, (r-1)\gamma+i_2)$ in $\mathbf{R}_r$, an overlap between rows $((r-1)\gamma + i_1, (r-1)\gamma + i_3)$ in $\mathbf{R}_r$, and an overlap between rows $((r-1)\gamma+i_2, (r-1)\gamma+i_3)$ in $\mathbf{R}_q$ correspond to overlaps between pairs of rows $(i_1, i_2)$, $(i_1, i_3)$, and $(i_2 + (r-q)\gamma, i_3 + (r-q)\gamma)$ in $\mathbf{\Pi}_1^{1,\mathrm{p}}$, respectively. The overlaps between rows $(i_1, i_2)$ and between rows $(i_1, i_3)$ belong to the same replica and must have distinct indices. The third overlap belongs to another replica and automatically has a distinct index. The number of options for the third overlap is $t_{i_2+(r-q)\gamma, i_3+(r-q)\gamma}$. We partition the enumeration into the following two cases:

- The overlap between rows $(i_1, i_2)$ is chosen from $t_{\{i_1,i_2,i_3\}}$ overlaps among the

three rows, and the overlap between rows $(i_1, i_3)$ is chosen from the $[t_{\{i_1,i_3\}} - 1]^+$ remaining options.

- The overlap between rows $(i_1, i_2)$ is chosen from $(t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})$ overlaps that are exclusively between these two rows, and the overlap between rows $(i_1, i_3)$ is chosen from the $t_{\{i_1,i_3\}}$ options.

These two cases correspond to the two terms in (2.13).

3. Because of the structure of SC codes, an overlap between rows $((r-1)\gamma+i_1, (r-1)\gamma+i_2)$ in $\mathbf{R}_r$, $((r-1)\gamma + i_1, (r-1)\gamma + i_3)$ in $\mathbf{R}_q$, and $((r-1)\gamma + i_2, (r-1)\gamma + i_3)$ in $\mathbf{R}_s$ correspond to overlaps between pairs of rows $(i_1, i_2)$, $(i_1 + (r-q)\gamma, i_3 + (r-q)\gamma)$, and $(i_2 + (r-s)\gamma, i_3 + (r-s)\gamma)$ in $\mathbf{\Pi}_1^{1,\mathrm{p}}$, respectively. These overlaps belong to different replicas and thus have distinct indices. Consequently, the number of cycles-6 in this case is the number of ways that we can choose these three overlaps, which is given in (2.14).

□

**Remark 5.** *For the enumeration of cycles-*6*, we only need overlap parameters of at most degree* 3*, regardless of the column weight* $\gamma$*.*

Theorem 2 expresses the number of cycles-6 in the protograph of an SC code as a function the overlap parameters. We recall that given the independent non-zero overlap parameters, the rest of overlap parameters can be found using Lemma 3. Let $\bar{i} = (i \bmod \gamma)$.

**Theorem 2.** *The number of cycles-*6 *in the protograph of an SC code with parameters* $\kappa$*,* $\gamma$*,* $m$*,* $L$*, and* $\mathcal{O}_{\mathrm{ind}}$ *is:*

$$F = \sum_{k=1}^{m+1} (L - k + 1) F_1^k, \tag{2.15}$$

*and* $F_1^k$*,* $1 \leq k \leq (m+1)$*, is formulated as follows:*

$$F_1^1 = \sum_{\substack{\{i_1,i_2,i_3\} \subset \{0,\ldots,(m+1)\gamma-1\} \\ \bar{i}_1 \neq \bar{i}_2\,,\bar{i}_1 \neq \bar{i}_3,\ \bar{i}_2 \neq \bar{i}_3}} \mathcal{A}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2,i_3\}}),$$

$$F_1^2 = \sum_{\substack{i_1\in\{0,\ldots,(m+1)\gamma-1\}\\ \{i_2,i_3\}\subset\{\gamma,\ldots,(m+1)\gamma-1\}\\ \overline{i_1}\neq\overline{i_2},\,\overline{i_1}\neq\overline{i_3},\,\overline{i_2}\neq\overline{i_3}}} \mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2-\gamma,i_3-\gamma\}}) + \sum_{\substack{i_1\in\{0,\ldots,(m+1)\gamma-1\}\\ \{i_2,i_3\}\subset\{0,\ldots,m\gamma-1\}\\ \overline{i_1}\neq\overline{i_2},\,\overline{i_1}\neq\overline{i_3},\,\overline{i_2}\neq\overline{i_3}}} \mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2+\gamma,i_3+\gamma\}}),$$

$$\begin{aligned} F_1^{k\geq 3} = & \sum_{\substack{i_1\in\{0,\ldots,(m+1)\gamma-1\}\\ \{i_2,i_3\}\subset\{(k-1)\gamma,\ldots,(m+1)\gamma-1\}\\ \overline{i_1}\neq\overline{i_2},\,\overline{i_1}\neq\overline{i_3},\,\overline{i_2}\neq\overline{i_3}}} \mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2+(1-k)\gamma,i_3+(1-k)\gamma\}}) \\ & + \sum_{\substack{i_1\in\{0,\ldots,(m+1)\gamma-1\}\\ \{i_2,i_3\}\subset\{0,\ldots,(m-k+2)\gamma-1\}\\ \overline{i_1}\neq\overline{i_2},\,\overline{i_1}\neq\overline{i_3},\,\overline{i_2}\neq\overline{i_3}}} \mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2+(k-1)\gamma,i_3+(k-1)\gamma\}}) \\ & + \sum_{q=2}^{k-1} \sum_{\substack{i_1\in\{(q-1)\gamma,\ldots,(m+1)\gamma-1\}\\ i_2\in\{(k-1)\gamma,\ldots,(m+1)\gamma-1\}\\ i_3\in\{(k-1)\gamma,\ldots,(m+q)\gamma-1\}\\ \overline{i_1}\neq\overline{i_2},\,\overline{i_1}\neq\overline{i_3},\,\overline{i_2}\neq\overline{i_3}}} \mathcal{C}(t_{\{i_1,i_2\}}, t_{\{i_1+(1-q)\gamma,i_3+(1-q)\gamma\}}, t_{\{i_2+(1-k)\gamma,i_3+(1-k)\gamma\}}). \end{aligned} \tag{2.16}$$

*The functions $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are defined in Lemma 5.*

*Proof.* For a cycle-6, all three VNs are adjacent (connected to each other via one distinct CN). As a result, for an SC code with memory $m$, a cycle-6 spans at most $\chi = m + 1$ consecutive replicas, and (2.15) directly follows from (2.3).

1. For $k = 1$, we look for the number of cycles-6 having all their three VNs in replica $\mathbf{R}_1$ of $\mathbf{H}_{\text{SC}}^{\text{p}}$. Based on Lemma 5, the number of cycles-6 with CNs $i_1$, $i_2$, and $i_3$ and VNs in $\mathbf{R}_1$ is $\mathcal{A}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2,i_3\}})$. Then, we just need to find possible choices for $i_1$, $i_2$, and $i_3$. First, all the CNs must belong to the non-zero part of $\mathbf{R}_1$, i.e., $\{0, \ldots, (m+1)\gamma\}$. Second, the rows correspond to these CNs must have non-zero overlap parameters, i.e., $\overline{i_1} \neq \overline{i_2}$, $\overline{i_1} \neq \overline{i_3}$, $\overline{i_2} \neq \overline{i_3}$ (see Remark 3). Putting this together results in $F_1^1$ in (2.16).

2. For $k = 2$, we look for the number of cycles-6 spanning two replicas $\mathbf{R}_1$ and $\mathbf{R}_2$ of $\mathbf{H}_{\text{SC}}^{\text{p}}$, such that either two VNs are in $\mathbf{R}_1$ and one VN is in $\mathbf{R}_2$, or vice versa. Based on

Lemma 5, the number of cycles-6 with two VNs in $\mathbf{R}_1$, one VN in $\mathbf{R}_2$, and CNs $i_1$, $i_2$, and $i_3$ is $\mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2-\gamma,i_3-\gamma\}})$. Then, we just need to find all possible choices for $i_1$, $i_2$, and $i_3$. Two CNs $i_2$ and $i_3$ are connected to VNs in replicas $\mathbf{R}_1$ and $\mathbf{R}_2$, thus they must belong to $\{\gamma, \ldots, (m+1)\gamma\}$. CN $i_1$ is connected to VNs in $\mathbf{R}_1$, thus it must belong to $\{0, \ldots, (m+1)\gamma\}$. Putting this together results in the first summation of $F_1^2$ in (2.16). The second summation, which is for the case when one VN belongs to $\mathbf{R}_1$ and two VNs belong to $\mathbf{R}_2$, can be found similarly.

3. For $3 \leq k \leq (m+1)$, we look for the number of cycles-6 spanning $k$ replicas of $\mathbf{H}_{\text{SC}}^{\text{p}}$ starting from $\mathbf{R}_1$. Then, the first VN belongs to $\mathbf{R}_1$, the last VN belongs to $\mathbf{R}_k$, and the middle VN belongs to $\mathbf{R}_q$, $(1 \leq q \leq k)$. The first two summations in the expression of $F_1^k$ in (2.16) correspond to the cases $q = 1$ and $q = k$ (the proof is similar to the previous case). For the case of $2 \leq q \leq (k-1)$ and based on Lemma 5, the number of cycles-6 with one VN in $\mathbf{R}_1$, one VN in $\mathbf{R}_q$, one VN in $\mathbf{R}_k$, and CNs $i_1$, $i_2$, and $i_3$ is $\mathcal{C}(t_{\{i_1,i_2\}}, t_{\{i_1+(1-q)\gamma,i_3+(1-q)\gamma\}}, t_{\{i_2+(1-k)\gamma,i_3+(1-k)\gamma\}})$. The CN $i_1$ is connected to VNs in $\mathbf{R}_1$ and $\mathbf{R}_q$, therefore it must belong to $\{(q-1)\gamma, \ldots, (m+1)\gamma\}$. The CN $i_2$ is connected to VNs in $\mathbf{R}_1$ and $\mathbf{R}_k$, therefore it must belong to $\{(k-1)\gamma, \ldots, (m+1)\gamma\}$. The CN $i_3$ is connected to VNs in $\mathbf{R}_q$ and $\mathbf{R}_k$, therefore it must belong to $\{(k-1)\gamma, \ldots, (m+q)\gamma\}$. Putting this together results in the third summation of $F_1^k$ in (2.16).

□

Now, define $F^*$ to be the minimum number of cycles-6 in $\mathbf{H}_{\text{SC}}^{\text{p}}$. Thus, our discrete optimization problem is formulated as follows:

$$F^* = \min_{\mathcal{O}_{\text{ind}}} F. \tag{2.17}$$

The constraints of our optimization problem are the conditions under which the overlap parameters and the subsequent partitioning are valid.

**Example 4.** *For an SC code with parameters $\gamma = 3$ and $m = 1$, the constraints of the*

*optimization problem in (2.17) are:*

$$
\begin{aligned}
&0 \leq t_0 \leq \kappa, \\
&0 \leq t_{\{0,1\}} \leq t_0, \\
&t_{\{0,1\}} \leq t_1 \leq \kappa - t_0 + t_{\{0,1\}}, \\
&0 \leq t_{\{0,1,2\}} \leq t_{\{0,1\}}, \\
&t_{\{0,1,2\}} \leq t_{\{0,2\}} \leq t_0 - t_{\{0,1\}} + t_{\{0,1,2\}}, \\
&t_{\{0,1,2\}} \leq t_{\{1,2\}} \leq t_1 - t_{\{0,1\}} + t_{\{0,1,2\}}, \\
&t_{\{0,2\}} + t_{\{1,2\}} - t_{\{0,1,2\}} \leq t_2 \leq \kappa - t_0 - t_1 + t_{\{0,1\}} + t_{\{0,2\}} + t_{\{1,2\}} - t_{\{0,1,2\}}, \\
&\left\lfloor \tfrac{3\kappa}{2} \right\rfloor \leq t_0 + t_1 + t_2 \leq \left\lceil \tfrac{3\kappa}{2} \right\rceil .
\end{aligned}
\tag{2.18}
$$

*The last constraint in (2.18) guarantees a so-called balanced partitioning between* $\mathbf{H}_0^{\mathrm{p}}$ *and* $\mathbf{H}_1^{\mathrm{p}}$*. A balanced partitioning is preferred to prevent the case where a group of non-zero elements in either* $\mathbf{H}_0^{\mathrm{p}}$ *or* $\mathbf{H}_1^{\mathrm{p}}$ *are involved in significantly more cycles than the remaining non-zero elements. This constraint, although it might result in a sub-optimal solution in the protograph (in a few cases), is observed to be beneficial when we apply the CPO algorithm to construct the final code.*

Consider an underlying block code with parameters $\kappa$ and $\gamma$. In the partitioning, each circulant of the parity-check matrix of the underlying block code, i.e., $\mathbf{H}$, can be assigned to any of the $(m+1)$ component matrices, resulting in $(m+1)^{\kappa\gamma}$ possible options. The goal is to choose a partitioning that results in the lowest number of cycles-6 in the protograph of an SC code. Considering all possible partitioning options in a brute force fashion to find the optimal one is not practical. We reduced the problem of finding the optimal partitioning for the protograph of an SC code into an optimization problem over $\mathcal{N}_{\mathrm{ind}} = (m+1)^{\gamma} - 1$ overlap parameters described in (2.8).

For example, when $\gamma = 3$, the number of optimization variables are 7 and 26 for memories 1 and 2, respectively. When $\gamma = 4$, the number of optimization variables are 15 and 80 for memories 1 and 2, respectively. Any partitioning choice that corresponds to the optimal overlap parameters results in the minimum number of cycles-6 in $\mathbf{H}_{\mathrm{SC}}^{\mathrm{p}}$. We note that this

is the first work that presents the optimal partitioning for code parameters $\gamma \in \{3, 4\}$, $m \in \{1, 2\}$, and practical row weights, thanks to the reduction in the complexity of the optimization problem. For a large number of optimization variables, one can manually force all degree-1 overlap parameters to be fixed ($\simeq \kappa/(m+1)$) and/or force all overlap parameters with degrees greater than an integer threshold to be 0.

Example 5 summarizes all the necessary steps for finding the optimal partitioning of an SC code with parameters $\kappa$, $\gamma = 3$, $m = 1$, and $L$.

**Example 5.** *Using Theorem 2, the number of cycles-*6 *in the protograph of an SC code with parameters* $\kappa$, $\gamma = 3$, $m = 1$, *and* $L$ *is described in terms of the* 7 *overlap parameters in* $\mathcal{O}_{\text{ind}} = \{t_0, t_1, t_2, t_{\{0,1\}}, t_{\{0,2\}}, t_{\{1,2\}}, t_{\{0,1,2\}}\}$ *as follows:*

$$F = LF_1^1 + (L-1)F_1^2,$$

*where* $F_1^1$ *and* $F_1^2$ *are:*

$$\begin{aligned} F_1^1 = {} & \mathcal{A}(t_{\{0,1,2\}}, t_{\{0,1\}}, t_{\{0,2\}}, t_{\{1,2\}}) + \mathcal{A}(t_{\{0,1,5\}}, t_{\{0,1\}}, t_{\{0,5\}}, t_{\{1,5\}}) \\ & + \mathcal{A}(t_{\{0,2,4\}}, t_{\{0,2\}}, t_{\{0,4\}}, t_{\{2,4\}}) + \mathcal{A}(t_{\{0,4,5\}}, t_{\{0,4\}}, t_{\{0,5\}}, t_{\{4,5\}}) \\ & + \mathcal{A}(t_{\{1,2,3\}}, t_{\{1,2\}}, t_{\{1,3\}}, t_{\{2,3\}}) + \mathcal{A}(t_{\{1,3,5\}}, t_{\{1,3\}}, t_{\{1,5\}}, t_{\{3,5\}}) \\ & + \mathcal{A}(t_{\{2,3,4\}}, t_{\{2,3\}}, t_{\{2,4\}}, t_{\{3,4\}}) + \mathcal{A}(t_{\{3,4,5\}}, t_{\{3,4\}}, t_{\{3,5\}}, t_{\{4,5\}}), \end{aligned} \tag{2.19}$$

$$\begin{aligned} F_1^2 = {} & \mathcal{B}(t_{\{0,4,5\}}, t_{\{0,4\}}, t_{\{0,5\}}, t_{\{1,2\}}) + \mathcal{B}(t_{\{3,4,5\}}, t_{\{3,4\}}, t_{\{3,5\}}, t_{\{1,2\}}) \\ & + \mathcal{B}(t_{\{1,3,5\}}, t_{\{1,3\}}, t_{\{1,5\}}, t_{\{0,2\}}) + \mathcal{B}(t_{\{3,4,5\}}, t_{\{3,4\}}, t_{\{4,5\}}, t_{\{0,2\}}) \\ & + \mathcal{B}(t_{\{2,3,4\}}, t_{\{2,3\}}, t_{\{2,4\}}, t_{\{0,1\}}) + \mathcal{B}(t_{\{3,4,5\}}, t_{\{3,5\}}, t_{\{4,5\}}, t_{\{0,1\}}) \\ & + \mathcal{B}(t_{\{0,1,2\}}, t_{\{0,1\}}, t_{\{0,2\}}, t_{\{4,5\}}) + \mathcal{B}(t_{\{1,2,3\}}, t_{\{1,3\}}, t_{\{2,3\}}, t_{\{4,5\}}) \\ & + \mathcal{B}(t_{\{0,1,2\}}, t_{\{0,1\}}, t_{\{1,2\}}, t_{\{3,5\}}) + \mathcal{B}(t_{\{0,2,4\}}, t_{\{0,4\}}, t_{\{2,4\}}, t_{\{3,5\}}) \\ & + \mathcal{B}(t_{\{0,1,2\}}, t_{\{0,2\}}, t_{\{1,2\}}, t_{\{3,4\}}) + \mathcal{B}(t_{\{0,1,5\}}, t_{\{0,5\}}, t_{\{1,5\}}, t_{\{3,4\}}). \end{aligned} \tag{2.20}$$

*We note that all the overlap parameters in (2.19) and (2.20) are linear combinations of the* 7

*independent non-zero overlap parameters in $\mathcal{O}_{\text{ind}}$, see Example 2. The functions $\mathcal{A}$ and $\mathcal{B}$ are defined in Lemma 5. Our discrete optimization problem is formulated as follows:*

$$F^* = \min_{t_0,t_1,t_2,t_{\{0,1\}},t_{\{0,2\}},t_{\{1,2\}},t_{\{0,1,2\}}} F. \tag{2.21}$$

*The constraints of the optimization problem are found in Example 4. The solution of our optimization problem is not unique. However, since all the solutions result in the same $F^*$, we work with one of these solutions, and call it an optimal vector, $\mathbf{t}^* = \left[t_0^*\ t_1^*\ t_2^*\ t_{\{0,1\}}^*\ t_{\{0,2\}}^*\ t_{\{1,2\}}^*\ t_{\{0,1,2\}}^*\right]$.*

### 2.3.2 Stage 2: Circulant Power Optimization

Each cycle-6 in $\mathbf{H}_{\text{SC}}^{\text{p}}$ results in either 0 or $z$ cycles-6, which are instances of the common denominator, in $\mathbf{H}_{\text{SC}}$ depending on the circulant power arrangement. In this subsection, we introduce an algorithm to further reduce the number of cycles-6 in the lifted matrix $\mathbf{H}_{\text{SC}}$ by manipulating the circulant powers. After picking an optimal vector $\mathbf{t}^*$ to partition $\mathbf{H}^{\text{p}}$ and construct $\mathbf{H}_{\text{SC}}^{\text{p}}$, we run our heuristic CPO to further reduce the number of $(3, 3(\gamma-2))$ ASs/TSs (the common denominator instances) in the graph of $\mathbf{H}_{\text{SC}}$ with column weight $\gamma$. We start with a set of circulant powers that results in zero cycles-4 in the lifted graph (the graph of $\mathbf{H}_{\text{SC}}$). Then, we iteratively change a subset of circulant powers such that the number of $(3, 3(\gamma-2))$ ASs/TSs is reduced while no cycles-4 are introduced. In our heuristic algorithm, we exploit the structure of SC codes to reduce the computational complexity. Recall that in codes that have no cycles-4, a cycle-6 is a $(3, 3(\gamma-2))$ AS/TS. The steps of the CPO are:

1. Initially, assign circulant powers $f_{i,j} = ij$, $0 \le i \le \gamma-1$ and $0 \le j \le \kappa-1$, (as in AB codes) to all the $\gamma\kappa$ 1s in $\mathbf{H}^{\text{p}}$ (results in zero cycles-4 in $\mathbf{H}$ and $\mathbf{H}_{\text{SC}}$).

2. Construct $\mathbf{\Pi}_1^{\chi,\text{p}}$, which contains $\chi = m+1$ replicas and has the size $(\chi+m)\gamma \times \chi\kappa$ (see Definition 7), using $\mathbf{H}^{\text{p}}$ and $\mathbf{t}^*$. (Recall that the VNs of a cycle-6 span at most $\chi = m+1$ consecutive replicas.) Circulant powers of the 1s in $\mathbf{\Pi}_1^{\chi,\text{p}}$ are copied from the 1s in $\mathbf{H}^{\text{p}}$.

3. Define a counting variable $\theta_{i,j}$, $0 \leq i < \gamma$ and $0 \leq j < \kappa$, for each of the 1s in $\mathbf{H}^{\mathrm{p}}$. Define another counting variable $\theta'_{i',j'}$, $0 \leq i' < (\chi + m)\gamma$ and $0 \leq j' < \chi\kappa$, for each of the elements in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$. Initialize all the variables in this step with zeros. Notice that only $\chi\gamma\kappa$ counting variables of the form $\theta'_{i',j'}$ are associated with 1s in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$. The remaining counting variables will remain zeros.

4. Locate all the cycles-4 and cycles-6 in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$.

5. Specify the cycles-6 in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$ that have (2.6) satisfied, and call them *active cycles*. Let $F_1^{k,a}$, $k \in \{1, \ldots, \chi\}$, be the number of active cycles starting at the first replica and having their VNs spanning $k$ consecutive replicas in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$. Thus, from (2.3), the number of active cycles having their VNs spanning $k$ consecutive replicas in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$ is $(\chi - k + 1)F_1^{k,a}$. (For example, for $k = 1$, $\chi F_1^{1,a}$ is the number of active cycles having their VNs spanning one replica.)

6. Compute the number of $(3, 3(\gamma - 2))$ ASs/TSs in $\mathbf{H}_{\mathrm{SC}}$ using the following formula (see (2.3) and (2.6)):
$$F_{\mathrm{SC}} = \sum_{k=1}^{\chi} \Big( (L - k + 1) F_1^{k,a} \Big)\, z. \tag{2.22}$$

7. Count the number of active cycles each 1 in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$ is involved in. Assign weight $w_k = (L - k + 1)/(\chi - k + 1)$ to the number of active cycles having their VNs spanning $k$ consecutive replicas in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$ (see Remark 6 for more clarification). Store the weighted count associated with each 1 in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$, which is indexed by $(i', j')$, in $\theta'_{i',j'}$.

8. Calculate the counting variables $\theta_{i,j}$, $\forall i, j$, associated with the 1s in $\mathbf{H}^{\mathrm{p}}$ from the counting variables $\theta'_{i',j'}$ associated with the 1s in $\mathbf{\Pi}_1^{\chi,\mathrm{p}}$ (computed in step 7) using the following formula:
$$\theta_{i,j} = \sum_{i':i'_\gamma = i} \; \sum_{j':j'_\kappa = j} \theta'_{i',j'}, \tag{2.23}$$
where $i'_\gamma = (i' \bmod \gamma)$ and $j'_\kappa = (j' \bmod \kappa)$. Sort the $\gamma\kappa$ 1s of $\mathbf{H}^{\mathrm{p}}$ in a descending list according to the counts in $\theta_{i,j}$, $\forall i, j$.

9. Pick a subset of 1s from the top of this list, and change the circulant powers associated with them.

10. Using these interim new powers, do steps 5 and 6.

11. If $F_{\rm SC}$ is reduced while maintaining no cycles-4 in $\mathbf{H}_{\rm SC}$, update $F_{\rm SC}$ and the circulant powers, then go to step 7. Otherwise, return to step 9 to pick a different set of circulant powers or/and a different subset of 1s.

12. Iterate until the target $F_{\rm SC}$ (set by the designer) is achieved, or the reduction in $F_{\rm SC}$ approaches zero.

Note that step 9 is performed heuristically. The number of 1s to work with and how to choose them depends on the circulant size, the values of the counts, and how these values are distributed. Moreover, tracking the counts of active cycles and the distribution of their values over different 1s in $\mathbf{H}^{\rm p}$ is the deciding principle in choosing which 1s to select in each iteration.

**Remark 6.** *A cycle-*6 *starting at the first replica and spanning $k$ consecutive replicas, $\forall k \in \{1, \ldots, \chi\}$, is repeated $(\chi - k + 1)$ times in $\mathbf{\Pi}_1^{\chi,{\rm p}}$ and $(L - k + 1)$ times in $\mathbf{H}_{\rm SC}^{\rm p}$. An active cycle spanning $k$ consecutive replicas, which involves a* 1 *in $\mathbf{\Pi}_1^{\chi,{\rm p}}$ indexed by $(i', j')$, is counted $w_k$ times in $\theta'_{i',j'}$, and thus exactly $(\chi - k + 1)w_k = (L - k + 1)$ times in $\theta_{i,j}$ for the* 1 *in $\mathbf{H}^{\rm p}$ indexed by $(i, j)$. This justifies the weighting factors used in steps 7 and 8.*

**Remark 7.** *A $(3, b')$ configuration, where $b' \leq 3(\gamma - 2)$, in the protograph of the SC code can result in $(3, 3(\gamma - 2))$ ASs/TSs in the final (lifted) graph depending on the circulant power arrangement. That is the reason why we consider all cycles-*6 *in $\mathbf{H}_{\rm SC}^{\rm p}$, even those that are $(3, b')$ configurations and $b' < 3(\gamma - 2)$. Note that typically the protograph of an SC codes has cycles-*4*. By the circulant power arrangement, we guarantee that the lifted graph (the SC code) does not have any cycle-*4*.*

**Definition 9.** *The partitioning of the underlying block code can be described by matrix $\mathbf{PM} = [g_{i,j}]$ of size $\kappa \times \gamma$, called partitioning matrix. Each element $g_{i,j} \in \{0, \cdots, m\}$ implies*

*that the corresponding element in* $\mathbf{H}^{\mathrm{p}}$ *(resp., the corresponding circulant in* $\mathbf{H}$*) is assigned to* $\mathbf{H}^{\mathrm{p}}_{g_{i,j}}$ *(resp.,* $\mathbf{H}_{g_{i,j}}$*). Besides, the circulant power matrix* $\mathbf{CM} = [f_{i,j}]$*, with dimension* $\gamma \times \kappa$*, stores the powers of non-zero circulants. Our OO-CPO approach for designing SC codes outputs these two matrices.*

**Example 6.** *Suppose we want to design an SC code with parameters* $\kappa = z = 7$*,* $\gamma = 4$*,* $m = 1$*, and* $L = 30$ *using the OO-CPO (OO partitioning and CPO algorithm). Solving the optimization problem in (2.17) yields the following optimal vector:*

$$\begin{aligned}\mathbf{t}^* &= [t_0^*\ t_1^*\ t_2^*\ t_3^*\ t_{\{0,1\}}^*\ t_{\{0,2\}}^*\ t_{\{0,3\}}^*\ t_{\{1,2\}}^*\ t_{\{1,3\}}^*\ t_{\{2,3\}}^*\ t_{\{0,1,2\}}^*\ t_{\{0,1,3\}}^*\ t_{\{0,2,3\}}^*\ t_{\{1,2,3\}}^*] \\ &= [3\ 4\ 3\ 4\ 0\ 1\ 2\ 2\ 2\ 0\ 0\ 0\ 0\ 0],\end{aligned}$$

*which results in* $F^* = 4{,}680$ *cycles-*6 *in the graph of* $\mathbf{H}^{\mathrm{p}}_{\mathrm{SC}}$*. Next, we apply the CPO algorithm. The partitioning and circulant power matrices attained by the OO-CPO framework are given below:*

$$\mathbf{PM} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{CM} = \begin{bmatrix} 0 & 4 & 5 & 2 & 5 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 6 & 5 \\ 0 & 2 & 4 & 6 & 1 & 3 & 5 \\ 0 & 3 & 6 & 2 & 0 & 0 & 3 \end{bmatrix}.$$

*We note that the uncoupled case with AB circulant power arrangement (*$m = 0$*,* $\mathbf{H}_0 = \mathbf{H}$*, and* $f_{i,j} = ij$*) results in* $\mathbf{H}_{\mathrm{SC}}$ *with* 35,280 *cycles-*6*, the OO partitioning with AB circulant power arrangement (only stage 1) results in* $\mathbf{H}_{\mathrm{SC}}$ *with* 5,747 *cycles-*6*, and the OO-CPO framework (both stages) results in* $\mathbf{H}_{\mathrm{SC}}$ *with* $2,870$ *cycles-*6*.*

## 2.4 Simulation Results

In this section, we compare the performance of SC codes constructed by the OO-CPO framework with uncoupled block codes and SC codes constructed by the CV method [8, 9]. We demonstrate both the reduction in the number of detrimental objects and the bit error rate (BER) performance improvement over AWGN channels, achieved by the OO-CPO framework.

All the codes in this section are binary. The terms CV-AB, OO-AB, and OO-CPO represent three different methods of constructing SC codes: CV partitioning and AB circulant powers ($f_{i,j} = ij$), OO partitioning and AB circulant powers, and OO partitioning and circulant powers attained by the CPO program.

Block Code 2.1 and Block Code 2.2 are AB block codes with $\kappa = z = 17$. Block Code 2.1 has $\gamma = 3$ and rate 0.824, and Block Code 2.2 has $\gamma = 4$ and rate 0.765. Uncoupled Block Codes 2.1 (resp., Uncoupled Block Codes 2.2) represents 30 uncoupled Block Codes 2.1 (resp., Block Codes 2.2), and it can be considered as an SC code with $m = 0$, $\mathbf{H}_0 = \mathbf{H}$, and $L = 30$. SC Code 2.1, SC Code 2.2, ..., SC Code 2.7 are SC codes with parameters $\kappa = z = 17$, $L = 30$, length 8,670 bits, and constructed by different methods.

SC Code 2.1, SC Code 2.2, and SC Code 2.3 have $m = 1$, $\gamma = 3$, and rate 0.818, and they are constructed by the CV-AB ($\boldsymbol{\zeta} = [4\ 9\ 13]$), OO-AB, and OO-CPO methods, respectively. SC Code 2.2 and SC Code 2.3 have the partitioning matrix:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix},$$

and SC Code 2.3 has the following circulant power matrix:

$$\begin{bmatrix} 0 & 0 & 2 & 9 & 0 & 7 & 4 & 16 & 2 & 4 & 2 & 9 & 0 & 4 & 13 & 1 & 1 \\ 13 & 1 & 2 & 6 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 13 & 12 & 0 & 14 & 8 & 16 \\ 0 & 2 & 0 & 0 & 8 & 10 & 8 & 14 & 16 & 1 & 3 & 5 & 7 & 15 & 5 & 5 & 2 \end{bmatrix}.$$

SC Code 2.4, SC Code 2.5, and SC Code 2.6 have $m = 1$, $\gamma = 4$, and rate 0.757, and they are constructed by the CV-AB ($\boldsymbol{\zeta} = [3\ 7\ 11\ 15]$), OO-AB, and OO-CPO methods, respectively.

SC Code 2.5 and SC Code 2.6 have the partitioning matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and SC Code 2.6 has the following circulant power matrix:

$$\begin{bmatrix} 0 & 10 & 2 & 8 & 2 & 0 & 5 & 7 & 15 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 11 & 15 & 2 & 14 & 10 & 3 & 6 & 7 & 8 & 9 & 4 & 11 & 12 & 8 & 14 & 10 & 16 \\ 11 & 2 & 4 & 12 & 8 & 11 & 12 & 9 & 15 & 4 & 13 & 5 & 6 & 1 & 11 & 13 & 15 \\ 11 & 3 & 6 & 9 & 2 & 16 & 8 & 4 & 7 & 10 & 13 & 16 & 2 & 5 & 8 & 6 & 14 \end{bmatrix}.$$

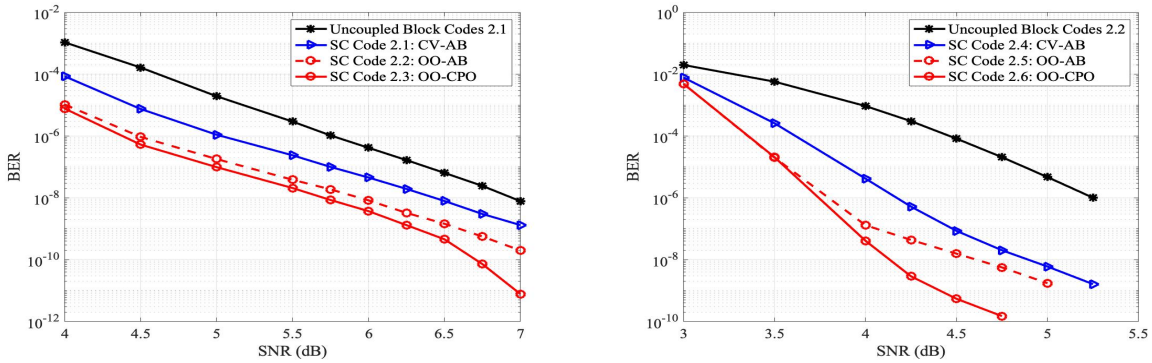
SC Code 2.7 has $m = 2$, $\gamma = 3$, and rate 0.812, and it is constructed by the OO-CPO method. SC Code 2.7 has the following partitioning matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 2 & 0 & 2 & 2 & 0 & 1 & 1 & 0 & 1 & 2 & 2 & 2 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and the following circulant power matrix:

$$\begin{bmatrix} 9 & 5 & 4 & 1 & 2 & 15 & 10 & 14 & 0 & 3 & 0 & 0 & 0 & 0 & 7 & 4 & 0 \\ 0 & 13 & 6 & 3 & 6 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 0 & 2 & 3 & 6 & 8 & 1 & 12 & 14 & 16 & 1 & 12 & 5 & 7 & 9 & 11 & 13 & 15 \end{bmatrix}.$$

Table 2.1 shows the number of cycles-6 for Uncoupled Block Codes 2.1 and 2.2 and SC Codes 2.1, ..., 2.7. According to our results, an SC code constructed by the CV-AB method achieves about 57% reduction in the number of cycles-6 compared to the uncoupled case for $m = 1$ and $\gamma \in \{3, 4\}$. When we apply our OO-CPO framework for $m = 1$ and $\gamma \in \{3, 4\}$, the reduction in the population of cycles-6 compared to the uncoupled case reaches




(a) (b)

Figure 2.5: BER curves over AWGN channel for SC codes with length 8,670 bits, memory $m = 1$, and constructed with different methods: (a) $\gamma = 3$, (b) $\gamma = 4$.

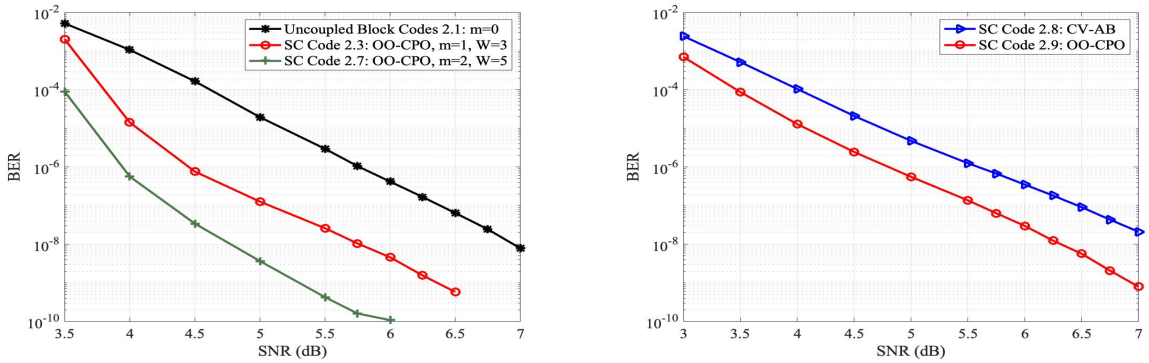
up to 89%. For the OO-CPO framework with $m = 2$ and $\gamma = 3$, the number of cycles-6 is 0 which means the girth of the code becomes 8.

Next, we compare the performance of the SC codes over AWGN channels. We include the performance curves for the uncoupled codes as references since a chain of $L$ uncoupled block codes can be viewed as an SC code with $m = 0$ and coupling length $L$. Figure 2.5 (a) shows the BER curves in the error floor region for Uncoupled Block Codes 1 ($m = 0$) and SC Codes 2.1, 2.2, and 2.3 ($m = 1$). All these codes have $\gamma = 3$. The figure demonstrates that our OO-CPO method outperforms the CV method by nearly 2 orders of magnitude, and that each stage of the framework is necessary to achieve this improvement.

Figure 2.5 (b) shows similar findings for SC codes with $\gamma = 4$. Based on our results, the performance improvement of our two-stage framework relative to the CV method is preserved

Table 2.1: Population of cycles-6 for SC codes with $z = \kappa = 17$, $L = 30$, and different construction methods.

| Construction method | Code name | $m$ | $\gamma$ | No. cycles-6 |
|---|---|---|---|---|
| Uncoupled-AB | Uncoupled Block Codes 2.1 | 0 | 3 | 138,720 |
| | Uncoupled Block Codes 2.2 | 0 | 4 | 554,880 |
| CV-AB | SC Code 2.1 | 1 | 3 | 59,024 |
| | SC Code 2.4 | 1 | 4 | 238,697 |
| OO-CPO | SC Code 2.3 | 1 | 3 | 14,960 |
| | SC Code 2.6 | 1 | 4 | 91,494 |
| | SC Code 2.7 | 2 | 3 | 0 |




(a) (b)

Figure 2.6: BER curves over AWGN channel: (a) SC codes with length 8,670 bits, $\gamma = 3$, and different memories, (b) SC codes with length 2,940 bits, rate 0.564, and constructed with different methods.

when we increase the column weight. SC Code 2.6 constructed by the OO-CPO framework achieves nearly 5 orders of magnitude performance improvement in the early error floor region and more than 1.3 dB SNR gain compared to the uncoupled setting ($m = 0$).

In Figure 2.6 (a), we show the effect of increasing the memory on the performance of SC codes, where we used a windowed min-sum decoding [4]. This comparison is fair in the sense of having comparable decoding latency and complexity (of the same order) as suggested in [31–33]. If the windowed decoding is used, the decoding latency depends on the constraint length not the codeword length. When we increase the memory, we need to use a bigger window size for decoding, and consequently we increase the decoding latency and complexity. However, the performance improves dramatically.

Figure 2.6 (a) shows the BER performance in the error floor region for SC codes with different memories. As we see, increasing the memory notably improves the error floor performance of SC codes. For Uncoupled Block Codes 2.1, SC Code 2.3, and SC Code 2.7, we use a block decoder, a windowed decoder with window size $W = 3$, and a windowed decoder with $W = 5$, respectively. Figures 2.5 and 2.6 also demonstrate a good performance improvement in the waterfall region achieved via our OO-CPO framework. One important reason is that the multiplicity of the low-weight codewords that include $(3, 3(\gamma - 2))$ ASs/TSs as subgraphs in their configurations strongly affects the waterfall performance.

Finally, we compare the performance of low-rate SC codes constructed by different methods. SC Codes 2.8 and SC Codes 2.9 have parameters $\kappa = z = 7$, $\gamma = 3$, $m = 1$, and $L = 60$. They have code length 2,940 bits and rate 0.564. SC Code 2.8 is constructed by the CV-AB ($\boldsymbol{\zeta} = [2\ 4\ 6]$) method, and SC Code 2.9 is constructed by the OO-CPO method. The partitioning and circulant power matrices for SC Code 2.9 are given below:

$$\mathbf{PM} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \qquad \mathbf{CM} = \begin{bmatrix} 0 & 1 & 3 & 5 & 2 & 4 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 5 & 0 & 2 & 4 & 6 & 2 \end{bmatrix}.$$

The number of cycles-6 for SC Code 2.8 and SC Code 2.9 are 6,650 and 413, respectively, which shows 94% reduction attained by our framework compared to the CV method. As Figure 2.6 (b) shows, our framework achieves nearly 1.4 orders of magnitude performance improvement at SNR = 7.0 dB and nearly 0.6 dB SNR gain at BER = $10^{-6}$ (using min-sum decoding).

**Remark 8.** *We note that the comparison between SC codes and their uncoupled counterparts is fair in terms of the number of problematic objects due to their similar number of VNs, CNs, and edges. In terms of the constraint length, and consequently, the decoding latency, an SC code with memory $m$ has a constraint length that equals $(m + 1)$ times the length of its underlying block code.*

## 2.5 Conclusion

In this chapter, we presented a new combinatorial approach for the finite-length analysis and design of CB SC codes. We exploited the structure of SC codes to present a new methodology for the enumeration of combinatorial objects, which can be applied to SC codes constructed by a wide variety of partitioning schemes and memories. Next, we introduced a novel partitioning scheme that operates on the protograph of an SC code to minimize the number of detrimental objects. Then, we proposed a heuristic algorithm for circulant power optimization that further reduces the population of these problematic objects in the

final graph. The proposed OO-CPO framework is an effective tool to construct SC codes that have a notably better performance than other existing construction techniques. In next chapter, we investigate the SC codes constructed by our new framework in modern dense storage applications.

## Acknowledgment

The majority of the material in this chapter is published in [12]. The author would like to thank the collaborators in this publication.

# CHAPTER 3

# Tailoring the SC Design Framework for Storage Applications

## 3.1 Introduction

In today's technological world, a huge amount of data is generated every day that need to be stored with a high reliability. Modern storage systems, e.g., Flash memories and magnetic recording (MR) systems, operate at very low error rates which require ECCs with outstanding performance. Certain combinatorial objects in the graph of LDPC codes are responsible for most errors that occur in the error floor region. These objects are not necessarily codework errors, and the errors resulting from them cannot be easily resolved in the belief propagation decoding.

In this chapter, we tackle the problem of optimized design of SC codes in the context of various storage applications. In particular, we identify combinatorial structures in the graphical representation of the LDPC, and more specifically SC codes, that are detrimental over different channel models corresponding to different storage devices. We show that the type and configuration of problematic objects depend on the channel model, and thus, a channel-aware code design is required to achieve a good performance in modern storage applications. We optimize SC codes to reduce the population of these problematic objects in their graphs.

Last but not the list, we investigate the channel non-uniformity and the ECC solutions to alleviate it. The SNR is not constant over different parts of a storage device. We present an ECC scheme that provides a well-defined cooperation among different parts of the channel

such that reliable data can help unreliable ones to be recovered, and consequently, to improve the overall error-correction performance. To this end, we present an SC code design along with a specific interleaving scheme that has robustness against the channel non-uniformity.

The rest of this chapter is organized as follows: In Section 3.2, we present an SC code design for MR storage devices. In Section 3.3, we present an SC code design for Flash memories. In Section 3.4, we present an SC code design along with a novel interleaving scheme for AWGN Channels with SNR variations. Finally, the conclusion appears in Section 3.5.

## 3.2 SC Code Design for MR Systems

MR applications require channel codes with outstanding error-correction capabilities. Currently, LDPC codes are a preferred choice in MR technologies [34–36] due to their great performance and low-complexity decoding. In this section, we investigate the practical utility of SC codes over MR channels. We demonstrate that, with appropriate code optimization techniques developed here for MR applications, SC codes can offer substantial performance advantage over popular LDPC codes. We present an explicit SC code design methodology specifically tailored for MR applications.

An intriguing observation made in [24] is that for the same LDPC code, the problematic objects for MR channels are combinatorially different from those encountered in the AWGN setting, thus necessitating a careful code design methodology for the MR applications. Here, we show that the same property holds for SC codes. Then, We demonstrate that the partitioning choice directly affects the cardinality of these problematic objects, which are shown to be certain ASs, see Definitions 3 and 4. In particular, we show that the count of detrimental ASs is the highest – and consequently that the performance is the worst – in the degenerate case which precisely corresponds to uncoupled block LDPC codes. We therefore demonstrate that coupling always improves the performance, and that the degree of improvement is dependent on the SC code design.

By a careful analysis of the error profile of SC codes in the error floor region, we demonstrate that several problematic combinatorial objects in the MR setting share a

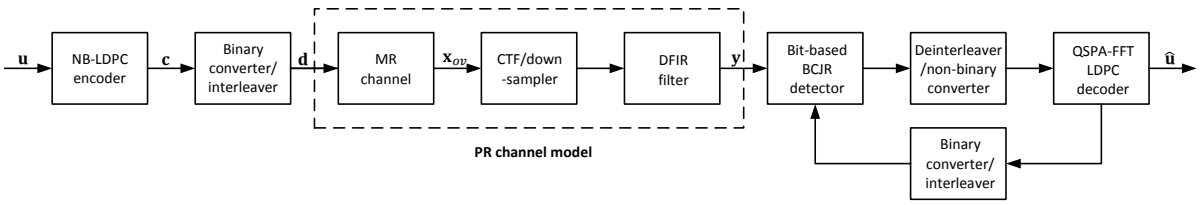



Figure 3.1: System model for one-dimensional MR channels utilizing an NB-LDPC code.

common denominator structure. We analytically find the partitioning choice that corresponds to the minimal number of these structures in the protograph, which in turn yields our first step in the code optimization. As the next step, we apply a heuristic CPO program to further reduce the population of the common denominator instances in the final (lifted) graph. For non-binary (NB) SC codes, we then perform existing edge weight optimization techniques, known as WCM method in [24] and [26], to reduce the number of remaining detrimental objects.

### 3.2.1 Overview of the MR System

Here, we briefly overview the key components of our one-dimensional MR system. We use the same MR setup as in [24]. We describe the system model for the one-dimensional MR system which will be used in this dissertation. Consider an NB LDPC code over GF($q$) with message length $n_1$ and codeword length $n_2$. As shown in Figure 3.1, the system model has the following components:

- **Encoding:** The message sequence $\mathbf{u} \in \mathrm{GF}(q)^{n_1}$ is encoded into a codeword $\mathbf{c} \in \mathrm{GF}(q)^{n_2}$. We focus on SC codes for which $n_2 = L\kappa z$.

- **Transmission:** Binary conversion, modulation, and interleaving are applied sequentially to the codeword $\mathbf{c}$ in order to generate the sequence of data $\mathbf{d}$, which is then written onto the MR channel.

- **Channel:** The MR channel incorporates inter-symbol interference (ISI) in addition to jitter and electronic noise. The channel oversampled output sequence is $\mathbf{x}_{ov}$. As reported in [24], the channel density is set to 1.4.

- **Filtering:** To generate the sequence $\mathbf{y}$, $\mathbf{x}_{ov}$ is passed through a continuous-time filter (CTF), a down-sampler, and a digital finite impulse response (DFIR) filter. The partial response (PR) channel consists of the MR channel, CTF, and DFIR units. As reported in [24], we use the equalization target [8 14 2].

- **Detection/Decoding**: The message sequence $\mathbf{u}$ is iteratively recovered via a finite-precision fast Fourier transform based $q$-ary sum-product algorithm (FFT-QSPA) LDPC decoder [37] in addition to a BCJR detector [38] based on pattern-dependent noise prediction (PDNP) [39]. For detection simplicity, we use a bit-based detector [24].

We refer to the internal iterations inside the LDPC decoder as local iterations, while a global iteration refers to one looping between the detector and the decoder. The decoder performs a specified number of local iterations (fewer if a codeword is reached) between any two successive global iterations. We use 10 global iterations and 20 local iterations for the simulation results reported later in this section.

### 3.2.2 Problematic Objects of SC Codes for MR Applications

We first briefly review the combinatorial objects of interest in the graph of LDPC codes that are known to be more problematic over MR channels. These objects were defined and studied in Section 1.2, and we revisit them here. In the context of the MR applications, only certain subclasses of ASs matter [24]. Thus, the ASs are classified as either *balanced* (BASs) or *unbalanced* (UBASs); both are defined in Definition 4. The BASs are the problematic objects over MR channels that need to be avoided in the code design.

We consider the class of AB SC codes as an exemplar, and determine the dominant BASs for column weights $\gamma = 3$ and $\gamma = 4$. In each case, we establish a common denominator structure that the unlabeled configurations of dominant BASs share. Here and elsewhere, dominant BASs refer to BASs that cause most of the decoding errors in the error floor region.

Table 3.1: Error profile (number of specific errors out of 100 errors) for SC Code 3.1 at SNR = 17.25 dB and FER = $2.33 \times 10^{-6}$.

| Error Type | $(6,0)$ | $(8,0)$ | $(10,0)$ | $(6,1)$ | $(8,2)$ | Other |
|---|---|---|---|---|---|---|
| Count | 76 | 8 | 6 | 3 | 2 | 5 |

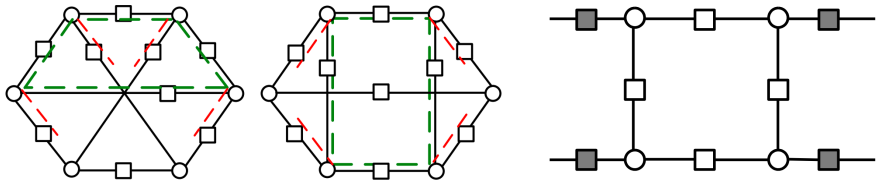
#### 3.2.2.1 AB SC Codes with Column Weight 3

For SC codes with $\gamma = 3$, we recognized that the dominant BASs are $(6,0)$, $(8,0)$, and $(10,0)$ in the MR setting. For example, consider SC Code 3.1 defined over GF(4) with parameters $\kappa = z = 23$, $\gamma = 3$, and $L = 5$. SC Code 3.1 has random CV partitioning, AB circulant powers, and random edge weights, and the length and rate are 5,290 bits and 0.843, respectively. The error profile for SC Code 3.1 over the MR channel is shown in Table 3.1. The errors are collected at SNR = 17.25 dB.

As we see, the dominant BASs are $(6,0)$, $(8,0)$, and $(10,0)$. We note that this is in contrast to the AWGN setting, where $(3,3)$ and $(4,2)$ ASs are deemed most problematic [8]. On the other hand, unlabeled configurations of $(6,1)$ and $(8,2)$ BASs that appear in Table 3.1 are in fact the same as the unlabeled configurations of $(6,0)$ and $(8,0)$ BASs, respectively. Thus, these configurations still share the same common denominator as $(6,0)$ and $(8,0)$ BASs.

An interesting property is that there are exactly two $(6,0)$ BASs that are topologically non-isomorphic, as shown in Figure 3.2 (a). As it is apparent from the figure, both of these configurations have $(4,4)$ ASs as substructures, while for example $(3,3)$ AS is just found in one of the configurations. Additionally, $(4,4)$ AS also exists as a substructure in other relevant BASs such as $(8,0)$ and $(10,0)$. As a result, we identify $(4,4)$ AS, shown in Figure 3.2 (b), as the common denominator structure that we seek to minimize.

We thus focus our attention on the $(4,4)$ substructure as the new object of interest in the partitioning and circulant power assignment, and argue that minimizing the population of the $(4,4)$ AS, as the parent configuration (substructure), reduces the number of all children configurations (superstructures), and improves the code performance. We also remark that for this column weight, $(4,4)$ AS is not an object of interest in canonical AWGN channels. As a result, using AWGN-optimized codes is an inferior strategy for MR channels.



(a) (b)

Figure 3.2: (a) The two non-isomorphic (unlabeled) configurations for $(6, 0)$ BAS. In each configuration, one $(4, 4)$ substructure is shown in green dashed lines as an example, and unsatisfied CNs are marked with red dashed lines. (b) The configuration of $(4, 4)$ AS.

Table 3.2: Error profile (number of specific errors out of 100 errors) for SC Code 3.2 at SNR $= 16.25$ dB and FER $= 9.41 \times 10^{-9}$.

| Error Type | $(8, 2)$ | $(12, 0)$ | $(10, 0)$ | $(8, 4)$ | $(10, 2)$ | Other |
|---|---|---|---|---|---|---|
| Count | 51 | 17 | 16 | 4 | 3 | 9 |

#### 3.2.2.2 AB SC Codes with Column Weight 4

For SC codes with $\gamma = 4$ in the MR setting, the dominant BASs are $(8, 2)$, $(8, 4)$, $(10, 0)$, and $(12, 0)$ which is in contrast to the AWGN setting where $(6, 4)$ ASs are the most problematic [23]. This observation again demonstrates that using AWGN-optimized codes is not the best strategy for MR channels. For example, consider SC Code 3.2 defined over GF(4) with parameters $\kappa = z = 29$, $\gamma = 4$, and $L = 5$. SC Code 3.2 has random CV partitioning, AB circulant powers, and random edge weights, and the length and rate are 8,410 bits and 0.834, respectively. The error profile for SC Code 3.2 over the MR channel is shown in Table 3.2. The errors are collected at SNR $= 16.25$ dB.

As an example, Figure 3.3 depicts two non-isomorphic configurations for $(8, 2)$ BAS and one configuration for $(10, 0)$ BAS. Both depicted configurations for the $(8, 2)$ BAS share $(4, 8)$ and $(3, 6)$ TSs as common substructures (as do other $(8, 2)$ BAS configurations not pictured here), but the $(10, 0)$ BAS configuration, shown in Figure 3.3 (c), does not include any $(3, 6)$ TS (neither do some other $(10, 0)$ BAS configurations not pictured here). As a result, we are interested in minimizing the number of instances of $(4, 8)$ TS shown in Figure 3.3 (d) as the common denominator structure. The $(4, 8)$ TS does not appear as a decoding error by itself. Here, reducing the population of the common denominator structure that most

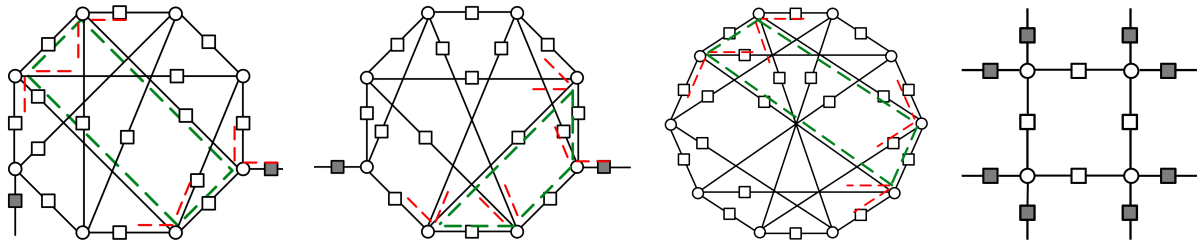



Figure 3.3: Unlabeled configurations for (a) $(8, 2)$ BAS. (b) $(8, 2)$ BAS. (c) $(10, 0)$ BAS. (d) $(4, 8)$ TS. Examples of $(4, 8)$ TS as substructures are shown in green dashed lines in (a)-(c), and unsatisfied CNs are marked with red dashed lines.

dominants BASs share in their Tanner graphs leads us to the $(4, 8)$ TS that does not appear as a decoding error by itself.

### 3.2.3 Tailoring SC Code Design for MR Applications

In this subsection, we adjust the stages of our methodology for designing SC codes for MR applications. Spatial coupling effectively rewires the connections in the Tanner graph of several block LDPC codes. As a result, some of the necessary existence conditions for BASs are violated, so that the number of detrimental objects is always fewer for SC codes compared to the uncoupled block codes. The amount of reduction of the number of these detrimental objects is dictated by the choice of the partitioning and the circulant power assignment. Similar to our SC design framework for AWGN channels introduced in Section 2, our procedure for the optimization of SC codes for MR channels includes two main steps: optimal partitioning and optimized lifting. For the NB designs, one has an additional choice of selecting edge weights.

First, given an underlying block code and desired memory and coupling length, we carefully build and solve the optimization problem of the OO partitioning, focusing on the objects of interest in the case of MR channels. Next, we customize the CPO program for reducing the population of problematic objects in the (unlabeled) lifted graph. For the NB codes, we then apply the WCM technique from [24, 26] for the edge weight optimization to eliminate as many as possible of the remaining dominant detrimental objects. According to

Section 3.2.2, the common denominator structures for SC codes with $\gamma = 3$ and $\gamma = 4$ are $(4, 4)$ AS and $(4, 8)$ TS, respectively. Both of these objects can be categorized as a cycle-8 with no interconnections.

The main difference between optimizing SC codes for MR channels and AWGN channels is in the protograph design. The common denominator of problematic objects for AWGN channels and MR channels are cycle-6 and cycle-8 with no interconnections, respectively. A cycle-6 in the graph of LDPC codes can only be generated from a cycle-6 in the protograph; however, a cycle-8 in the graph of LDPC codes can be generated from 9 different patterns in the protograph. This difference makes the optimization problem for finding the optimal partitioning considerably more challenging for MR channels.

It is known that each cycle in the (unlabeled) lifted graph is derived from a configuration in the protograph under specific conditions on the powers of the circulants involved in that configuration, [29, 30]. In the OO stage, we minimize the population of the configurations in the protograph that can possibly result in cycles-8 with no interconnections after lifting, and in the CPO stage, we operate on the circulant powers.

#### 3.2.3.1 Patterns of Interest in the Protograph Design

There are several configurations in the protograph of an LDPC code that can result in cycles-8 in the lifted graph. These configurations are called *patterns*. All distinct patterns, denoted by $\mathrm{P}_1, \ldots, \mathrm{P}_9$, are illustrated in Figure 3.4.

**Lemma 6.** *Each way of traversing a pattern for generating a cycle-8 is called a cycle-8 candidate. Let $\xi_{\mathrm{P}_l}$ be the number of distinct cycle-8 candidates for pattern $\mathrm{P}_l$. Then,*

| $l$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $\xi_{\mathrm{P}_l}$ | 1 | 3 | 3 | 6 | 6 | 1 | 2 | 2 | 1 |

The proof is given in [40]. One cycle-8 candidate for each pattern $\mathrm{P}_l$, $l \in \{1, \ldots, 9\}$, is shown with red lines in Figure 3.4.

**Remark 9.** *Since (protograph) parity-check matrix of an LDPC code with $\gamma = 3$ cannot*

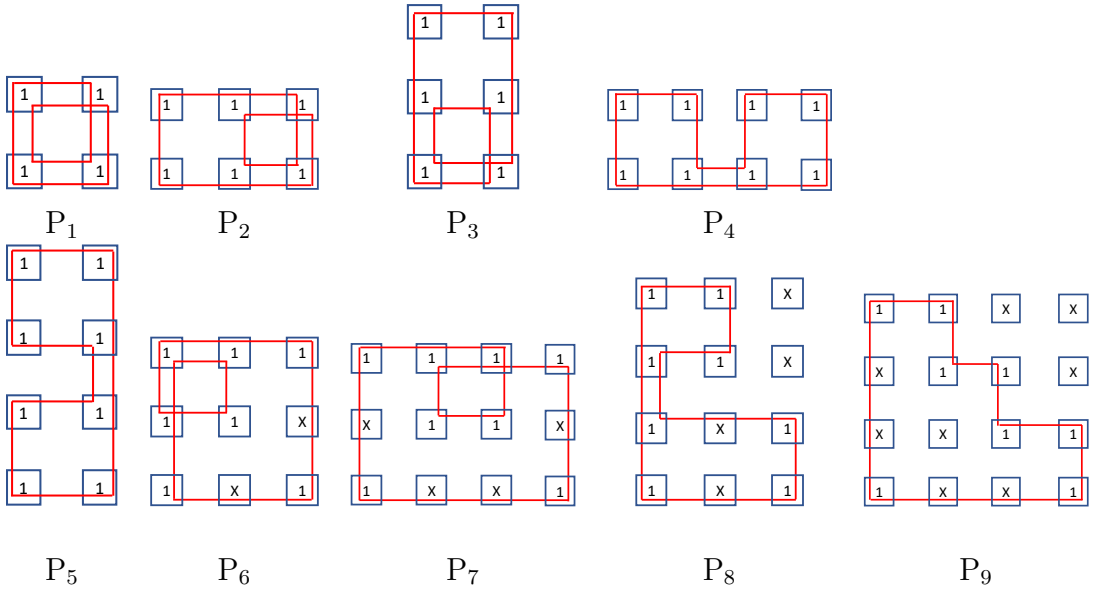



Figure 3.4: Distinct patterns in the protograph of an LDPC code that can result in cycles-8 in the lifted graph. One way of traversing each pattern to obtain a cycle-8 is shown with red lines.

*have four* 1*s in a column,* $\mathrm{P}_5$ *and* $\mathrm{P}_8$ *cannot exist in protograph of LDPC codes with* $\gamma = 3$*, reducing the number of distinct patterns from nine to seven.*

All the patterns except for $\mathrm{P}_1$ result in either 0 or $z$ cycles-8 after lifting based on the powers of the circulants involved in the pattern [29,30]. However, an instance of $\mathrm{P}_1$ results in either 0 or $z/2$ cycles-8 after lifting based on the powers of its engaged circulants. Thus, we define *pattern weight* $\Xi_{\mathrm{P}_l}$ as follows:

$$\Xi_{\mathrm{P}_l} = \begin{cases} 0.5\,\xi_{\mathrm{P}_l}, & \text{if} \quad l = 1, \\ \xi_{\mathrm{P}_l}, & \text{if} \quad l \in \{2, \ldots, 9\}. \end{cases} \tag{3.1}$$

#### 3.2.3.2 Optimal Overlap Partitioning for Pattern Optimization

Here, we extend the OO partitioning scheme introduced in Section 2.3.1 in order to minimize the weighted summation of the number of pattern instances in the protograph of an SC code rather than minimizing the number of cycles-6. The overlap parameters, introduced in Definition 8, are extremely helpful to quantify the number of combinatorial objects, e.g.,

cycles, patterns, ..., in protograph of SC codes in terms of a set of integer-valued parameters. As stated in Lemma 3, the set of all independent non-zero overlap parameters is:

$$\mathcal{O}_{\text{ind}} = \{t_{\{i_1,\ldots,i_d\}} \mid 1 \leq d \leq \gamma, 0 \leq i_1, \ldots, i_d < m\gamma, \\ \forall\{i_u, i_v\} \subseteq \{i_1, \ldots, i_d\}\, i_u \neq i_v \pmod{\gamma}\}.$$

The overlap parameters that are not included in $\mathcal{O}_{\text{ind}}$ are either zero or functions of the overlap parameters in $\mathcal{O}_{\text{ind}}$.

Now, we define the optimization problem for identifying the optimal partitioning for MR channels. The weighted summation of the number of instances of all patterns is:

$$F_{\text{tot}} = \sum_{l=1}^{9} \Xi_{\text{P}_l} F_{\text{P}_l}, \tag{3.2}$$

where $F_{\text{P}_l}$ is the number of instances of $\text{P}_l$ and can be expressed in terms of the overlap parameters. Then, any set of independent non-zero overlap parameters that minimizes $F_{\text{tot}}$ is preferable, and any partitioning choice that corresponds to this set of overlap parameters is optimal. Thus, we call this approach OO partitioning.

In [40], the number of instances of each pattern, i.e., $F_{\text{P}_l}$ and $l \in \{1, \ldots, 9\}$, in the protograph of an SC code with parameters $\kappa$, $\gamma \geq 3$, $m = 1$, and $L$ are precisely derived in terms of the overlap parameters in $\mathcal{O}_{\text{ind}}$. Our discrete optimization problem is:

$$F^*_{\text{tot}} = \min_{\mathcal{O}_{\text{ind}}} F_{\text{tot}}. \tag{3.3}$$

The constraints of our optimization problem are the conditions under which the overlap parameters and the subsequent partitioning are valid.

#### 3.2.3.3 Circulant Power Optimization for MR Systems

Using the optimal partitioning obtained by the OO scheme, we construct the protograph of the SC code, i.e., $\mathbf{H}^{\text{p}}_{\text{SC}}$. The next step is using a customized version of the CPO program to

prevent as many remaining patterns as possible from being reflected in the lifted graph as cycles-8 with no interconnections. We remind that cycle-8 with no interconnections is the common denominator of problematic BASs for $\gamma = 3$ and $\gamma = 4$, see Figure 3.2 and Figure 3.3. In this part, we study the changes that need to be applied to the CPO program introduced in Section 2.3.2 to make it suitable for MR channels. More details can be found in [40].

Each pattern $\mathrm{P}_l$ spans at most either $\chi = m+1$ or $\chi = 2m+1$ replicas, depending on the value of $l$. Thus, it suffices for the CPO program to operate on $\mathbf{\Pi}_1^{\chi=2m+1,\mathrm{p}}$, see Definition 7. We describe a cycle-8 candidate in $\mathbf{\Pi}_1^{\chi=2m+1,\mathrm{p}}$ in terms of the row group and column group indices of the involved circulants as $h_1$–$l_1$–$h_2$–$l_2$–$h_3$–$l_3$–$h_4$–$l_4$, which is a particular way of traversing a pattern. We note that the row group and column group indices do not need to be necessarily distinct. For example for $\mathrm{P}_1$ there are two distinct row group indices and two distinct column group indices. A cycle-8 candidate results in $z$ (or $z/2$ in the case of $\mathrm{P}_1$ only) cycles-8 after lifting if and only if [29]:

$$f_{h_1,l_1} + f_{h_2,l_2} + f_{h_3,l_3} + f_{h_4,l_4} = f_{h_1,l_2} + f_{h_2,l_3} + f_{h_3,l_4} + f_{h_4,l_1} \pmod{z} \tag{3.4}$$

There are two conditions that are equivalent to having interconnections in the configuration of a cycle-8. In order to have an interconnection between VNs with column group indices $l_1$ and $l_3$ using CN with row group index $h_5$ in the lifted graph, the following condition must hold:

$$f_{h_1,l_1} + f_{h_2,l_2} + f_{h_5,l_3} = f_{h_1,l_2} + f_{h_2,l_3} + f_{h_5,l_1} \pmod{z} \tag{3.5}$$

Similarly, in order to have an interconnection between VNs with column group indices $l_2$ and $l_4$ using CN with row group index $h_6$ in the lifted graph, the following condition must hold:

$$f_{h_1,l_1} + f_{h_6,l_2} + f_{h_4,l_4} = f_{h_1,l_2} + f_{h_6,l_4} + f_{h_4,l_1} \pmod{z} \tag{3.6}$$

By applying the necessary modifications to the CPO program due to the facts discussed in this part, we can heuristically adjust the circulant powers to eliminate as many as possible of the remaining cycles-8 with no interconnections in the (unlabeled) lifted graph.

#### 3.2.3.4 Edge Weight Optimization for MR Systems

As the final step of our design procedure, we replace all 1s of $\mathbf{H}_{\mathrm{SC}}$, that is constructed by the OO-CPO approach, with non-zero elements in GF($q$) and apply the edge weight optimization to remove as many as possible of remaining dominant BASs. For the NB setting, both the topology and edge weights can be manipulated to eliminate a problematic AS. Using edge weight optimization proposed in [26] and [24], we attempt to eliminate as many as possible of remaining problematic BASs.

### 3.2.4 Simulation Results

In this subsection, we compare the number of common denominator instances along with the frame error rate (FER) performances over MR channels for several codes that are constructed using different methods. We first describe our code parameters. SC Code 3.3, SC Code 3.4, SC Code 3.5, and SC Code 3.6 are NB SC codes with parameters $\gamma = 3$, $\kappa = 19$, $z = 46$, $m = 1$, and $L = 5$, and they are defined over GF(4). Thus, these codes have length 8,740 bits and rate 0.811. Moreover,

- SC Code 3.3 has CV partitioning [8], $f_{i,j} = 2i^2j$, and random edge weights.
- SC Code 3.4 has OO partitioning, $f_{i,j} = 2i^2j$, and random edge weights.
- SC Code 3.5 has OO partitioning, CPO applied, and random edge weights.
- SC Code 3.6 has OO partitioning, CPO applied, and optimized edge weights.

The cutting vector used for constructing SC Code 3.3 is $\boldsymbol{\zeta} = [4\ 9\ 15]$. The partitioning matrix for SC Codes 3.4, 3.5, and 3.6 is given below:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix},$$

and the circulant power matrix for SC Codes 3.5 and 3.6 is given below:

$$\begin{bmatrix} 11 & 1 & 43 & 23 & 16 & 29 & 6 & 40 & 22 & 29 & 6 & 22 & 20 & 35 & 40 & 0 & 8 & 0 & 0 \\ 23 & 15 & 3 & 26 & 33 & 44 & 36 & 17 & 35 & 34 & 33 & 11 & 10 & 26 & 28 & 39 & 31 & 34 & 36 \\ 0 & 8 & 12 & 24 & 32 & 32 & 39 & 10 & 30 & 26 & 40 & 42 & 20 & 12 & 20 & 28 & 36 & 44 & 20 \end{bmatrix}.$$

Besides, Block Code 3.1 is a CB block code with $\kappa = 19$, $z = 46$, $\gamma = 3$, $f_{i,j} = 2i^2j$, and it is defined over GF(4). Uncoupled Block Codes 3.1 represents 5 uncoupled Block Codes 3.1, and it can be considered as an SC code with $m = 0$, $\mathbf{H}_0 = \mathbf{H}$, $L = 5$, length 8,740 bits, and rate 0.842. Block Code 3.2 is a block code defined over GF(4), which is also protograph-based, designed as in [26]. It has $\gamma = 3$, $z = 46$, length 8,832 bits, rate 0.81 (similar to SC Codes 3.3, ..., 3.6), and it has un-optimized edge weights.

First, the number of cycles-8 with no interconnections in the unlabeled graphs are shown in Table 3.3, which demonstrates the significant gains achieved by the OO-CPO approach compared to other techniques. In particular, the proposed OO-CPO approach achieves 92% reduction in the number of the common denominator instances compared to the uncoupled setting, and 78% reduction compared to the CV technique. Moreover, all stages of the code design framework are necessary and helpful in order to construct high performance SC codes suitable for MR channels.

Second, we compare the error floor performance of Uncoupled Block Codes 3.1, SC Codes 3.3, ..., 3.6, and Block Code 3.2 over MR channels. The MR channel model we use is given in Figure 3.1. Figure 3.5 demonstrates the effectiveness of our SC code design for MR channels. In particular, SC Code 3.5 (designed using the OO-CPO approach) outperforms SC Code 3.3 (designed using the CV approach) by about 3 orders of magnitude at SNR = 15 dB, and by about 1.1 dB at FER = $10^{-5}$. More intriguingly, SC Code 3.5 outperforms Block Code 3.2 by

Table 3.3: Population of cycles-8 with no interconnections for SC codes with $\gamma = 3$, $\kappa = 19$, $z = 46$, $m = 1$, $L = 5$, and different construction methods.

| Uncoupled Block Codes 3.1 | SC Code 3.3 | SC Code 3.4 | SC Code 3.5 |
|---|---|---|---|
| 2,425,120 | 845,434 | 579,968 | 184,667 |

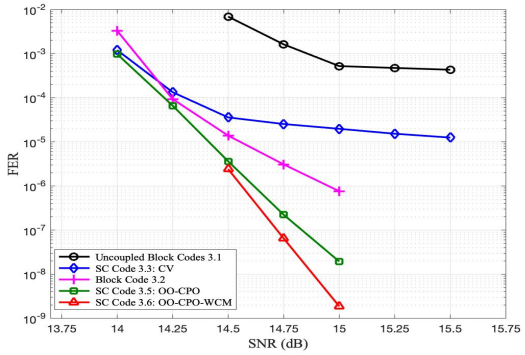



Figure 3.5: FER curves over MR channel for codes with similar length and rate constructed with different methods.

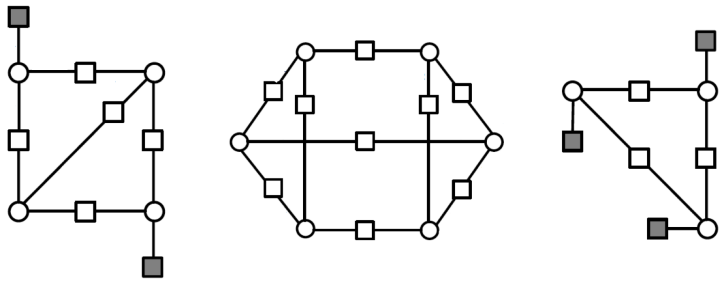
about 1.6 orders of magnitude at SNR = 15 dB, and by almost 0.4 dB at FER = $10^{-6}$. From Figure 3.5, the WCM (edge weight optimization) framework provides 1 order of magnitude additional gain.

## 3.3 SC Code Design for Flash Memories

Modern dense Flash memories operate at a very low error rate which require powerful ECCs with outstanding error-correction capabilities. The SC codes are a suitable choice to be used for Flash memories due to their capacity approaching performance, low-latency decoding, and high error-correction capability. In this section, we tailor our multi-stage framework, introduced in Chapter 2, for designing NB SC codes suitable for Flash memories.

For this purpose, we revisit the general absorbing sets of type two (GASTs), which are originally introduced in [26] and defined in this dissertation as Definition 5. The GASTs are the combinatorial objects in graph of NB LDPC codes that are problematic over Flash channels. We then customize our SC design framework for avoiding as many as dominant GASTs in the graph of SC codes as possible via targeting their common denominator in the partitioning and lifting stages, and then targeting the GAST configurations in the edge weight optimization stage.

We first identify the dominant GASTs in the error profile of SC codes and extract their common denominator substructure. In the first stage of the customized framework, we



(a) (b) (c)

Figure 3.6: Configurations for (a) $(4, 2, 2, 5, 0)$ GAST. (b) $(6, 0, 0, 9, 0)$ GAST. (c) The $(3, 3)$ AS, i.e., cycle-6, as the common denominator. Appropriate edge weights are assumed for NB configurations.

use OO partitioning to minimize the population of the common denominator instances in the protograph of SC codes. As the second stage, we apply the CPO program to further reduce the number of the common denominator instances in the (unlabeled) lifted graphs by heuristically adjusting the circulant powers. As the final stage, we optimize the edge weights to eliminate as many as remaining instances of the targeted GASTs in the labeled graphs as possible using the WCM method in [26].

### 3.3.1 Problematic Objects of SC Codes for Flash Memories

The LDPC codes are among the most attractive ECC solutions to be deployed over Flash memories [41]. The NB LDPC codes offer superior performance over their binary counterparts, and thus they are more preferred for modern Flash memories. The GASTs are the objects that dominate the error floor of NB LDPC codes over practical Flash channels due to the channel asymmetry. The GASTs and their unlabeled versions (UGASTs) are defined in Definitions 5 and 6.

In order to simultaneously target multiple objects in the unlabeled graph of SC codes, we first derive the dominant common denominator of multiple detrimental UGASTs. For the dominant UGASTs encountered in NB LDPC codes with $\gamma = 3$ simulated over Flash channels, the $(3, 3)$ AS, i.e., cycle-6, occurs as the dominant common substructure [26]. Two dominant GASTs for NB LDPC codes with $\gamma = 3$ over Flash channel along with their

common denominator are illustrated in Figure 3.6.

### 3.3.2 Tailoring SC Code Design for Flash Channels

The goal is to design NB SC codes with a small multiplicity of detrimental GASTs in their graphs. We first reduce the population of cycles-6, as the instances of the common denominator of problematic UGASTs, in the unlabeled graph of SC codes via OO partitioning and CPO program. We note that the common denominator is the same for Flash memories and AWGN channels. Thus, one can use the method described in Chapter 2, i.e., OO-CPO, to design the unlabeled graph of SC codes for Flash memories. After applying the OO-CPO technique to optimize the unlabeled graph of the SC codes, we then optimize the edge weights. In particular, we use the WCM framework in [24, 26] to remove GASTs from the labeled graph of SC codes through edge weight processing.

### 3.3.3 Simulation Results

In this subsection, we evaluate the performance of our designed SC codes over practical Flash channel models. We first compare the number of cycles-6, as instances of the common denominator of interest, in graph of SC codes constructed using various methods. Then, we compare the performance of these SC codes over Flash channels. We first describe our code parameters.

SC Code 3.7, SC Code 3.8, and SC Code 3.9 are NB SC codes with parameters $\kappa = z = 19$, $\gamma = 3$, $m = 1$, $L = 20$, and defined over GF(4). Thus, these codes have length 14,440 bits and rate 0.834. SC Code 3.7 is constructed using the CV partitioning and AB circulant powers (CV-AB), SC Code 3.8 is constructed using OO partitioning and AB circulant powers (OO-AB), and SC code 3.9 is constructed using OO partitioning and circulant powers attained by the CPO program (OO-CPO). The cutting vector used for constructing SC Code 3.7 is

Table 3.4: Population of cycles-6 for SC codes with $\gamma = 3$, $\kappa = z = 19$, $m = 1$, $L = 20$, and different construction methods.

| Uncoupled Block Codes 3.3 | SC Code 3.7 | SC Code 3.8 | SC Code 3.9 |
|---|---|---|---|
| 129,960 | 55,366 | 30,571 | 16,340 |

$\boldsymbol{\zeta} = [5\ 9\ 15]$. The partitioning matrix for SC Code 3.8 and SC Code 3.9 is given below:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix},$$

and the circulant power matrix for SC Code 3.9 is given below:

$$\begin{bmatrix} 0 & 6 & 17 & 18 & 8 & 8 & 8 & 16 & 6 & 10 & 0 & 9 & 6 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 1 & 14 & 15 & 16 & 5 & 13 \\ 9 & 2 & 4 & 6 & 5 & 10 & 12 & 14 & 16 & 18 & 1 & 3 & 5 & 7 & 16 & 14 & 4 & 1 & 8 \end{bmatrix}.$$

Moreover, Block Code 3.3 is an AB block code with $\kappa = z = 19$ and $\gamma = 3$, and it is defined over GF(4). Uncoupled Block Codes 3.3 represents 20 uncoupled Block Codes 3.3, and it can be viewed as an SC code with $m = 0$, $\mathbf{H}_0 = \mathbf{H}$, $L = 20$, length 14,440 bits and rate 0.842. The number of cycles-6 in the unlabeled graph of these codes are shown in Table 3.4.

Next, we study the performance gains achieved by our NB SC design over Flash memories. For this purpose, we consider the labeled graph of Uncoupled Block Codes 3.3, SC Code 3.7, SC Code 3.8, and SC Code 3.9. We consider both the random edge weight assignment and the optimized edge weight assignment using WCM method in [26]. The Flash channel model we use is the normal-Laplace mixture (NLM) Flash channel [25]. We use 3 reads, and the sector size is 512 bytes.

Let RBER be the raw bit error rate, and UBER be the uncorrectable bit error rate. One formulation of UBER, which is recommended by industry, is FER divided by the sector size in bits. The performance gain achieved using three stages of our framework, i.e., OO-CPO-WCM, is highlighted in Figure 3.7. Our best NB SC code, i.e., SC Code 3.9 with optimized edge

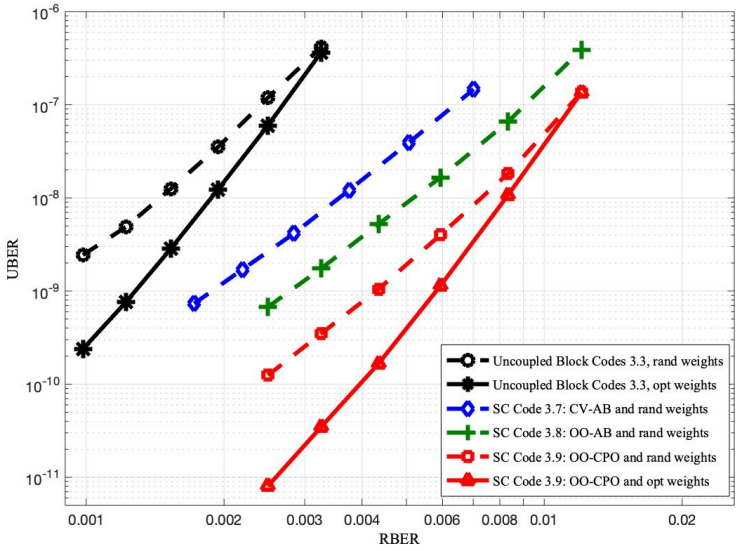



Figure 3.7: Simulation results over the NLM Flash channel for codes with similar length and rate constructed with different methods.

weights, achieves more than 500% RBER gain compared to Uncoupled Block Codes 3.3 with random edge weights over a practical Flash channel.

## 3.4 SC Code Design for Channels with SNR Variations

In a magnetic recording (MR) device, some sections can be more error prone than other sections because of the read/write mechanism and physical properties of the device [42]. A realistic channel model for MR systems must consider the variation of SNR among consecutive sections of a hard disk drive. We develop ECCs that address the SNR variation for data storage systems. For channels with uniform SNR, i.e., channels with a single SNR value, the goal is to find a code that achieves a certain level of BER for that SNR. For channels with SNR variation, conventional ECCs are designed to achieve the target BER for the section with the lowest SNR. For the sections with higher SNR, this approach results in an additional redundancy which is not required to achieve the target BER.

One solution for handling the non-uniformity of SNR is using interleaving. Suppose the channel consists of $N$ sections of equal sizes, and the length of each codeword is also equal to

the length of a section. Each codeword is divided into $N$ chunks. Then, different chunks from all codewords are interleaved such that one chunk from each codeword is passed through one section of the channel. As a result, the average SNR that all codewords are affected by is the same and equal to the average SNR of the channel, and one just needs to consider the average SNR rather than the worst SNR in the code design process [42]. One disadvantage of this approach is that the length of each codeword is equal to the length of one section of the channel which might be relatively short.

In this section, we first present a sub-optimal partitioning, called minimum overlap (MO) scheme. The MO scheme is a low-complexity technique for partitioning CB codes and constructing SC codes. The MO partitioning scheme is more preferred for the cases where finding the OO partitioning is not computationally affordable due to the size of its discrete optimization problem, or for the cases where we want to identify an appropriate partitioning for a fixed set of circulant powers. The reason why we introduce the MO scheme in this section is that all SC codes that will be evaluated over channels with SNR variation in this section are constructed using MO partitioning.

It may be worthwhile to note that the content of this dissertation does not obey the chronological order of the contributions. In fact, the MO partitioning was introduced prior to the OO partitioning. However, due to the benefits that MO partitioning still offers, and because all the SC codes that we evaluated over non-uniform AWGN channels are constructed by the MO partitioning, we present and study the MO partitioning in this section.

Next, we present a finite-length analysis and construction of SC codes for channels with SNR variation. Our SC code construction provides local error correction for each section by means of the underlying codes that cover one section each, and simultaneously, an added level of error correction by means of coupling among the underlying codes. Consequently, and because of the structure of SC codes, more reliable sections can help unreliable ones to achieve an improved performance. Finally, we introduce a low-complexity interleaving scheme specific to SC codes that further improves their performance over channels with SNR variation.

Our simulation results show that our SC codes outperform individual block codes by more than 1 and 2 orders of magnitudes in the error floor region compared to the block codes with and without regular interleaving, respectively. This improvement is more pronounced for larger values of memory and column weight.

### 3.4.1 Minimum Overlap Partitioning for Constructing SC Codes

We present a systematic partitioning scheme, with a relatively low computational complexity, to construct high performance SC codes. We remind that the overlap between two rows of a matrix is defined as the number of columns in which the two rows simultaneously have non-zero values, see Definition 8. Let $t_y$ be the maximum overlap between rows (in pairs) of the $y$'th component matrix of $\mathbf{H}^{\mathrm{p}}$, i.e., $\mathbf{H}^{\mathrm{p}}_y$.

The technique aims to minimize the overall overlap $t = \min_{y\in\{0,\cdots,m\}} t_y$ in a balanced partitioning in order to structurally prevent certain detrimental structures in the graph representation of the code that are intrinsically formed by the overlaps, e.g., cycles and ASs. We first present the MO partitioning for partitioning CB codes with column weight $\gamma = 3$. We then extend the MO scheme for partitioning CB codes with column weight $\gamma = 2(m+1)$ into component matrices with minimum overlap, such that there are two elements (resp., circulants) in each column of $\mathbf{H}^{\mathrm{p}}$ (resp., $\mathbf{H}$) that are assigned to the same component matrix (balanced partitioning). This extension is helpful to systematically construct SC codes with $(\gamma = 4, m = 1)$ and $(\gamma = 6, m = 2)$.

#### 3.4.1.1 MO Partitioning for SC Codes with Column Weight 3

We focus on CB codes with $\gamma = 3$ as underlying block codes, and we consider AWGN channels. As a result, the problematic objects are $(3, 3)$, $(4, 2)$, and $(5, 3)$ ASs, shown in Figure 2.1, as verified by comprehensive Monte Carlo simulations. For SC codes with $\gamma = 3$ over AWGN channels, cycle-6 is the common denominator of interest as discussed in detail in Section 2.3.

We first consider the case where $m = 1$. Each circulant of $\mathbf{H}$ can belong to either of the two component matrices, leading to $2^{\kappa\gamma}$ possible partitioning choices. Considering all

possible ways of partitioning in a brute force way to find the optimal one is not practical. We narrow down the search to a subset of partitioning options that contains at least near-optimal solutions. We specify the circulants that belong to $\mathbf{H}_0$. Then, the remaining circulants belong to $\mathbf{H}_1$. Our MO partitioning scheme is described below:

1. Each row group of $\mathbf{H}$ has $\kappa$ circulants. We define the set $S = \{1, \cdots, \kappa\}$ which contains one element for each circulant.

2. For any row group of $\mathbf{H}$, we assign half of circulants to $\mathbf{H}_0$ and the remaining to $\mathbf{H}_1$[1]. We choose the half-half split to balance the distribution of circulants among the component matrices. The set $B_i \subset S$ contains the indices of circulants in the $i$'th row group that belong to $\mathbf{H}_0$.

3. Let $t$ be the maximum number of elements that any pair of sets $(B_i, B_j)$, $i, j \in \{0, 1, 2\}$ and $i \neq j$, have in common, i.e.,

$$t = \max\{|B_i \cap B_j| \ \ \{i, j\} \subset \{0, 1, 2\}\}.$$

   We consider an initial value $t = 0$.

4. The sets $B_0$, $B_1$, and $B_2$ are chosen such that any two sets have at most $t$ elements in common, and there is no element that appears in all of them. If there is no solution, we increment $t$ and repeat step 4. Otherwise, we record all solutions as valid MO partitioning candidates.

5. Since we find the lowest possible $t$ such that step 4 produces at least one solution, we call this approach minimum overlap (MO) partitioning.

Lemma 7 states the number of possible MO partitioning options for $\gamma = 3$ and $m = 1$.

**Lemma 7.** *Let $\mathcal{N}$ be the number of possible MO partitioning choices for an SC code with parameters $\kappa$, $\gamma = 3$, $m = 1$, and $t$:*

[1]If $\kappa$ is not even, we assign $\lfloor \kappa/2 \rfloor$ out of $\kappa$ circulants, in each row group of $\mathbf{H}$, to $\mathbf{H}_0$ and the remaining ones to $\mathbf{H}_1$.

$$\mathcal{N} \approx \sum_{a=0}^{a_u} \sum_{b=0}^{b_u} \sum_{c=c_l}^{c_u} \binom{\kappa}{\lfloor \kappa/2 \rfloor} \binom{\lfloor \kappa/2 \rfloor}{a}^2 \binom{\lfloor \kappa/2 \rfloor - a}{b} \binom{\lfloor \kappa/2 \rfloor - a}{c} \binom{a}{\lfloor \kappa/2 \rfloor - b - c}, \quad (3.7)$$

*where*

$$a_u = \min\{\lfloor \kappa/2 \rfloor, t\}, \qquad c_l = \max\{\lfloor \kappa/2 \rfloor - a - b, 0\},$$
$$b_u = \min\{\lfloor \kappa/2 \rfloor - a, t\}, \qquad c_u = \min\{\lfloor \kappa/2 \rfloor - \max\{a, b\}, t\}.$$

*Proof.* The number of possible ways to assign half of the elements in $S$ to $B_0$ is approximately equal to $\mathcal{N}_0$:

$$\mathcal{N}_0 = \binom{\kappa}{\lfloor \kappa/2 \rfloor}.$$

Consider one of the choices for assigning the elements in $B_0$. Next, we choose the elements of $B_1$. The number of ways to assign half of elements of $S$ to the set $B_1$, such that $|B_0 \cap B_1| = a$, is approximately equal to $\mathcal{N}_1^a$ ($|.|$ shows the number of elements in a set):

$$\mathcal{N}_1^a = \binom{\lfloor \kappa/2 \rfloor}{a} \binom{\kappa - \lfloor \kappa/2 \rfloor}{\lfloor \kappa/2 \rfloor - a} \approx \binom{\lfloor \kappa/2 \rfloor}{a}^2.$$

The parameter $\mathrm{a_u}$ indicates the maximum number of elements that two sets $B_0$ and $B_1$ can have in common, and it is equal to:

$$\mathrm{a_u} = \min\{|B_0|, |B_1|, t\} \approx \min\{\lfloor \kappa/2 \rfloor, t\}.$$

Next, consider one of the possible choices for assigning the elements of $B_0$ and $B_1$, such that $|B_0 \cap B_1| = a$. The number of ways to assign half of elements of $S$ to $B_2$, such that $|B_0 \cap B_2| = b$ and $|B_1 \cap B_2| = c$ is approximately equal to $\mathcal{N}_2^{b,c|a}$:

$$\mathcal{N}_2^{b,c|a} = \binom{\lfloor \kappa/2 \rfloor - a}{b} \binom{\lfloor \kappa/2 \rfloor - a}{c} \binom{\kappa - \lfloor \kappa/2 \rfloor - \lfloor \kappa/2 \rfloor + a}{\lfloor \kappa/2 \rfloor - b - c}$$

$$\approx \binom{\lfloor \kappa/2 \rfloor - a}{b} \binom{\lfloor \kappa/2 \rfloor - a}{c} \binom{a}{\lfloor \kappa/2 \rfloor - b - c}.$$

Given the parameter $a$, the maximum number of elements that $B_0$ and $B_2$ can have in common is:

$$\mathrm{b_u} = \min\{|B_0 \setminus B_1|, t\} \approx \min\{\lfloor \kappa/2 \rfloor - a, t\}.$$

Besides, the maximum number of elements that $B_1$ and $B_2$ can have in common is:

$$\mathrm{c_u} = \min\{|B_1 \setminus B_0|, |B_2 \setminus B_0|, t\} \approx \min\{\lfloor \kappa/2 \rfloor - \max\{a, b\}, t\}.$$

Moreover, the minimum number of elements that $B_1$ and $B_2$ must have in common to have $|B_2| = \lfloor \kappa/2 \rfloor$ is $\mathrm{c_l}$:

$$\mathrm{c_l} = \max\{|B_2 \setminus B_0| - |S \setminus B_0 \setminus B_1|, 0\} \approx \max\{\lfloor \kappa/2 \rfloor - a - b, 0\}.$$

Finally, the total number of possible MO partitioning choices for a CB SC code is equal to $\mathcal{N}$:

$$\mathcal{N} \approx \sum_{a=0}^{\mathrm{a_u}} \sum_{b=0}^{\mathrm{b_u}} \sum_{c=c_l}^{\mathrm{c_u}} \mathcal{N}_0 \mathcal{N}_1^a \mathcal{N}_2^{b,c|a},$$

which results in (3.7). □

Among these $\mathcal{N}$ candidates for MO partitioning, we find the one that results in the minimum number of problematic objects in the lifted graph. For example, consider an AB code with $\kappa = z = 7$ and $\gamma = 3$ as the underlying block code. The number of partitioning choices that achieve the minimum possible overlap, i.e., $t_{\min} = 1$, is equal to $\mathcal{N} = 15{,}330$ which is dramatically less than the number of all possible partitioning choices, i.e., $2^{21}$. A partitioning matrix with $t = 1$ that results in the minimum number of cycles-6 in the

corresponding SC code with coupling length $L = 30$ is given below:

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

It is interesting to note that for this set of code parameters, the optimal MO partitioning coincides with the optimal partitioning that we found thorough an exhaustive search.

Now, we develop MO partitioning approach for designing SC codes with higher memories. Increasing the memory provides more degrees of freedom. However, there is a trade-off between the memory and latency, and by increasing the memory, we also increase the latency in the windowed decoding [4]. Consider an SC code with $\gamma = 3$ and $m = 2$. We intend to partition $\mathbf{H}$ into three component matrices, i.e. $\mathbf{H}_y$, $y \in \{0, 1, 2\}$, such that $\mathbf{H} = \mathbf{H}_0 + \mathbf{H}_1 + \mathbf{H}_2$.

For any row group of $\mathbf{H}$, we assign one third of circulants to each of $\mathbf{H}_0$, $\mathbf{H}_1$, and $\mathbf{H}_2$[2]. The set $B_i \in S$ contains the indices of circulants in the row group $i$ that belong to $\mathbf{H}_0$, the set $B'_i \in S$ contains the indices of circulants in the row group $i$ that belong to $\mathbf{H}_1$ ($B_i \cap B'_i = \varnothing$), and the set $B''_i = S \setminus B_i \setminus B'_i$ likewise contains the indices of circulants in the row group $i$ that belong to $\mathbf{H}_2$. We impose the *zero-overlap condition* on these index sets, as follows: for any pair $(i, j)$ such that $i, j \in \{0, 1, 2\}$ and $i \neq j$, $B_i \cap B_j = \varnothing$, $B'_i \cap B'_j = \varnothing$, and $B''_i \cap B''_j = \varnothing$. Since the zero-overlap solution always exists for $m = 2$, further increasing $m$ does not decrease the minimum overlap value for case $\gamma = 3$.

For example, consider again an AB code with $\kappa = z = 7$ and $\gamma = 3$ as the underlying block code. A partitioning matrix that satisfies the zero-overlap condition and results in the minimum number of cycles-6 in the corresponding SC code with parameters $m = 2$ and $L = 30$ is given below:

$$\mathbf{PM} = \begin{bmatrix} 2 & 2 & 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 & 2 \end{bmatrix}.$$

[2]If $\kappa$ is not divisible by 3, we choose the nearest balanced partitioning, such as $\{\lfloor \kappa/3 \rfloor, \lfloor \kappa/3 \rfloor, \kappa - 2\lfloor \kappa/3 \rfloor\}$.

#### 3.4.1.2 MO Partitioning for SC Codes with Higher Column Weights

Here, we extend the MO technique in order to partition CB codes with column weight $\gamma = 2(m+1)$ into component matrices with minimum overlap, such that there are two elements (resp., circulants) in each column of $\mathbf{H}^{\mathrm{p}}$ (resp., $\mathbf{H}$) that are assigned to the same component matrix (balanced partitioning). This extension is helpful to systematically construct SC codes with $(\gamma = 4, m = 1)$ and $(\gamma = 6, m = 2)$. Theorem 3 states the minimum overlap value, and it also suggests a partitioning construction that achieves the minimum overlap value.

**Theorem 3.** *Consider an SC code with parameters $m$, $\kappa$, and $\gamma = 2(m+1)$. For a balanced partitioning where there are exactly two elements in any column of $\mathbf{H}^{\mathrm{p}}$ that are assigned to each component matrix $\mathbf{H}^{\mathrm{p}}_y$, $y \in \{0, \cdots, m\}$, the minimum overlap value is:*

$$t_{\min} = \left\lceil \frac{\kappa}{\binom{\gamma}{2}} \right\rceil . \tag{3.8}$$

*Proof.* Assume that the $\kappa$ columns in matrix $\mathbf{H}^{\mathrm{p}}$ are divided into stripes of size $\omega$ columns. Then, the number of stripes is $\lceil \frac{\kappa}{\omega} \rceil$, and if $\omega$ does not divide $\kappa$, the last stripe has a lower size than $\omega$. Since a stripe is a submatrix of $\mathbf{H}^{\mathrm{p}}$, the overlap parameter is defined for a stripe as well. We denote the overlap parameter for a stripe as $t_\omega$. The goal is to make each stripe have the minimum overlap parameter value. Each column of $\mathbf{H}^{\mathrm{p}}$ has $\gamma = 2(m+1)$ elements, and there are two elements in each column that are assigned to the same component matrix. This implies that the minimum possible value for the overlap parameter in each stripe is $t_\omega = 1$, even if the length of the stripe is 1 column. Then, we choose the maximum stripe length such that $t_\omega = 1$ still holds. The maximum $\omega$ such that $t_\omega = 1$ holds is:

$$\omega_{\max} = \binom{\gamma}{2}, \tag{3.9}$$

which is achieved by choosing a different pair of elements in each column in the stripe to be assigned to the same component matrix. Each stripe adds one unit to the final overlap

parameter value. Thus,

$$t_{\min} = \left\lceil \frac{\kappa}{\omega_{\max}} \right\rceil = \left\lceil \frac{\kappa}{\binom{\gamma}{2}} \right\rceil.$$

□

Next, we explain a systematic approach to find a partitioning that achieves the minimum overlap parameter. Given one stripe of matrix $\mathbf{H}^{\mathrm{p}}$, the corresponding stripes can be identified for $\mathbf{H}^{\mathrm{p}}_y$, $0 \leq y \leq m$, and for any linear combinations of them. In the construction procedure, we focus on properly constructing the first stripe with the overlap parameter $t_\omega = 1$ in the component matrix $\mathbf{H}^{\mathrm{p}}$. The other stripes, except for the last stripe, are constructed by arbitrary permutations of the columns of the first stripe, and the last strip is constructed from an arbitrary subset of columns of the first stripe. Consider the first stripe, and let $\mathcal{O}^\ell_y$, $0 \leq y \leq m$, $0 \leq \ell \leq \binom{\gamma}{2} - 1$, be the set of indices of the rows with non-zero values in column $\ell$ of that stripe in $\mathbf{H}^{\mathrm{p}}_y$. An element of $\mathcal{O}^\ell_y$ takes a value in $\{0, 1, \ldots, \gamma - 1\}$. Note that $|\mathcal{O}^\ell_y| = 2$, $\forall \ell$ and $\forall y$, because our partitioning is balanced.

For the case of $\gamma = 4$, the size of each stripe, except for the last stripe, is $\binom{4}{2} = 6$ columns. The construction that achieves $t_\omega = 1$ is simply to select distinct sets $\mathcal{O}^\ell_0$, $\forall \ell$, because this implies that the sets $\mathcal{O}^\ell_1 = \{0, \cdots, \gamma - 1\} \setminus \mathcal{O}^\ell_0$, for that stripe in $\mathbf{H}^{\mathrm{p}}_1$ are also distinct.

For the case of $\gamma = 6$, the size of each stripe, except for the last stripe, is $\binom{6}{2} = 15$. Here, we develop an algorithm to perform the construction. Let $\mathcal{O}^\ell_{0+1}$, $0 \leq \ell \leq 14$, be the set of indices of the rows with non-zero values in column $\ell$ of our stripe in the matrix $\mathbf{H}^{\mathrm{p}}_0 + \mathbf{H}^{\mathrm{p}}_1$. Note that $|\mathcal{O}^\ell_{0+1}| = 4$, $\forall \ell$. The algorithm determines each set $\mathcal{O}^\ell_{0+1}$ in the stripe sequentially starting from $\ell = 0$. Once the set $\mathcal{O}^\ell_{0+1}$ is determined, the algorithm determines the set $\mathcal{O}^\ell_0$, and consequently the set $\mathcal{O}^\ell_1$, such that:

$$\mathcal{O}^\ell_{0+1} \neq \mathcal{O}^r_{0+1},\ \mathcal{O}^\ell_0 \neq \mathcal{O}^r_0,\ \text{and}\ \mathcal{O}^\ell_1 \neq \mathcal{O}^r_1,\ \forall r < \ell. \tag{3.10}$$

There are $\binom{4}{2} = 6$ options to choose the set $\mathcal{O}^\ell_0$ out of the elements in the set $\mathcal{O}^\ell_{0+1}$. However, not all of these options are valid options because of the constraint in (3.10). For each $\ell$, the algorithm chooses the set $\mathcal{O}^\ell_{0+1}$ out of $\binom{6}{4} - \ell = 15 - \ell$ options such that the

number of valid options for the set $\mathcal{O}_0^\ell$ is maximized. This algorithm always starts with 6 options to choose the set $\mathcal{O}_0^0$ from the elements in $\mathcal{O}_{0+1}^0$, and ends with 1 option to choose the set $\mathcal{O}_0^{14}$ from the elements in $\mathcal{O}_{0+1}^{14}$. Note that once the sets $\mathcal{O}_{0+1}^\ell$, $\forall \ell$, are properly selected, the sets $\mathcal{O}_2^\ell$, $\forall \ell$, are properly selected. Consequently, $t_{\min}$ is achieved.

**Example 7.** *Let $\gamma = 4$, $\kappa = 12$, and $m = 1$. The length of each stripe is $\omega_{max} = \binom{4}{2} = 6$, and the number of stripes is $\kappa/\omega_{max} = 2$. As a result, $t_{\min} = 2$, and a partitioning matrix that achieves the minimum overlap is illustrated below:*

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

*For the first stripe (the first six columns in $\mathbf{H}^{\mathrm{P}}$), the parameters $\mathcal{O}_y^\ell$, $0 \leq y \leq 1$ and $0 \leq \ell \leq 5$, are listed below. The second stripe can be any permutation of the columns of the first stripe. In this example, we chose exactly the same order of columns for the second stripe as the first stripe.*

$$\begin{array}{llllll} \mathcal{O}_0^0 = \{0,1\} & \mathcal{O}_0^1 = \{0,2\} & \mathcal{O}_0^2 = \{1,2\} & \mathcal{O}_0^3 = \{0,3\} & \mathcal{O}_0^4 = \{1,3\} & \mathcal{O}_0^5 = \{2,3\} \\ \mathcal{O}_1^0 = \{2,3\} & \mathcal{O}_1^1 = \{1,3\} & \mathcal{O}_1^2 = \{0,3\} & \mathcal{O}_1^3 = \{1,2\} & \mathcal{O}_1^4 = \{0,2\} & \mathcal{O}_1^5 = \{0,1\} \end{array}$$

The partitioning that achieves the minimum overlap value is not unique. We suggest two methods for choosing the MO partitioning and constructing the final SC code. The first method is searching over all MO partitioning choices and finding the one that results in the minimum number of problematic objects in the lifted graph for a fixed arrangement of the circulant powers. The second method is choosing one MO partitioning and an initial set of circulant powers. Then, we apply the CPO algorithm to reduce the population of problematic objects by heuristically adjusting the circulant powers.

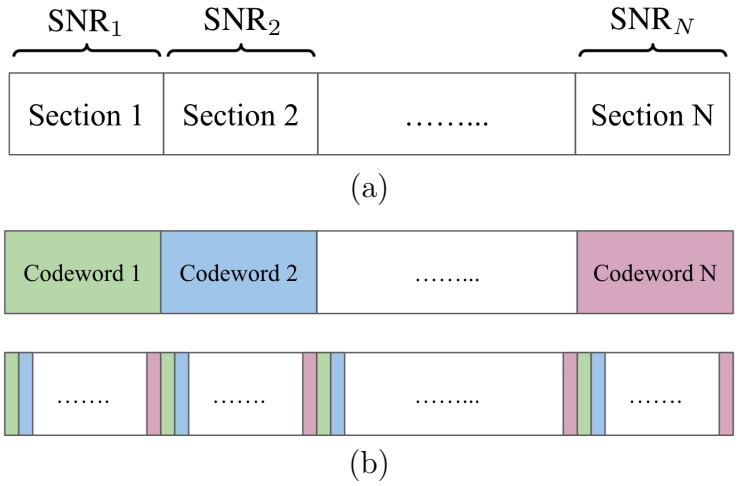



Figure 3.8: (a) A non-uniform channel with $N$ sections. (b) Original and interleaved sequence of encoded data; each color corresponds to one codeword.

### 3.4.2 Channels with SNR Variation

Our model for a channel with SNR variation is depicted in Figure 3.8 (a). It shows a channel with $N$ sections, and each section is considered as an AWGN channel with $\mathrm{SNR}_s$ ($s$ is the section index). For the $s$'th section, we state the SNR as $(\mathrm{SNR}_s)_{\mathrm{dB}} = (\mathrm{SNR}_{\mathrm{abs}})_{\mathrm{dB}} + (\Delta\mathrm{SNR}_s)_{\mathrm{dB}}$, where $\mathrm{SNR}_{\mathrm{abs}}$ is the absolute SNR, $\Delta\mathrm{SNR}_s$ is the variation from the absolute SNR for the $s$'th section, and $\mathrm{X}_{\mathrm{dB}} = 10\log_{10}\mathrm{X}$. We assume all sections have the same length. In a practical channel model, the value of $\Delta\mathrm{SNR}_i$ for different sections are dependent.

### 3.4.3 Interleaving to Mitigate Non-Uniformity

Based on the described model for the SNR variation, some sections of the channel have higher SNR than $\mathrm{SNR}_{\mathrm{abs}}$ while others do not. The interleaving technique is used to minimize the negative impact of SNR variation by introducing diversity [42]. The idea of interleaving is illustrated in Figure 3.8 (b). Interleaving is performed on the sequence of codewords before they pass through the channel, and de-interleaving is performed on the received sequence of data and before decoding.

First, a sequence of data is split into $N$ parts. Each part is coded individually by the same block code to generate $N$ codewords. Next, each codeword is partitioned into $N$ chunks,

and different chunks from all codewords are interleaved such that each $N$ consecutive chunks belong to $N$ different codewords. Finally, each $N$ consecutive chunks pass through a separate section of the channel. The interleaving helps to achieve a better error-correction capability since the average SNR that a codeword is affected by is the same for all codewords and it is equal to the average SNR of the channel, so each codeword has some reliable bits that can help recovering the less reliable bits.

### 3.4.4 SC Code Design to Alleviate SNR Variation

In this subsection, we first describe our framework to construct SC codes over channels with SNR variation, and explain why well-designed SC codes show a superior performance compared to block codes with interleaving. Then, we study the effect of parameter memory on the performance of SC codes over channels with SNR variation. Finally, we introduce an interleaving scheme for SC codes that further improves their performance for channels with non-uniform SNR.

#### 3.4.4.1 Code Design Machinery

Instead of using an individual block code for each section of the channel in Figure 3.8 (a), we use an SC code that spans several consecutive sections. By using SC codes, we can make more reliable sections help unreliable ones while keeping the decoding latency low. The decoding latency of an SC code is a function of the underlying code length and the window size of the decoder [4].

Parameters of our code design are illustrated in Figure 3.9. The length of the underlying block code is equal to the length of one section of the channel, thus each replica of an SC code spans one section of the channel. The coupling length $L$ determines how many sections are spanned by one SC codeword, i.e., $N = L$. The parameter $L$ must be chosen such that a variety of sections with different reliabilities are included. We use CB codes as underlying block codes. For partitioning the underlying block code, we use the MO partitioning scheme.

The memory $m$ of an SC code plays a critical role on its performance over channels with

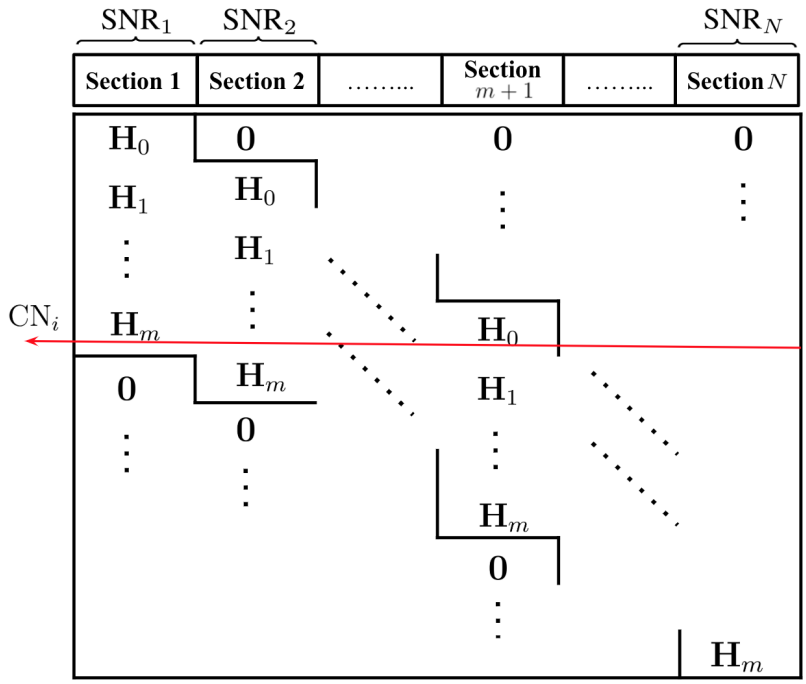



Figure 3.9: An SC code with memory $m$ constructed for a non-uniform channel. $\text{CN}_i$ spans $(m+1)$ consecutive sections.

SNR variation. The parameter $m$ determines how many different sections the VNs of a check equation span. All CNs of an SC code receive messages from VNs within $(m+1)$ consecutive sections (see $\text{CN}_i$ in Figure 3.9), except for the first and last group of $(m\gamma z)$ CNs which receive messages from a fewer number of sections. If a CN is connected to an unreliable VN, the other more reliable neighboring VNs of that CN can help correcting the VN error. If $m$ is chosen appropriately, each CN receives enough reliable messages to send improved messages to its neighboring VNs in an iterative decoding. However, there is a trade-off, and as we increase the memory, the decoding window size and consequently, the decoding latency also increase.

According to the simulation results, our SC codes outperform block codes (with interleaving) for channels with SNR variation. We also demonstrate that the performance gap over channels with SNR variation and corresponding uniform channels is lower for SC codes compared to block codes. This result verifies that the correlation that exists between different sections due to the structure of SC codes causes enough diversity to alleviate the negative

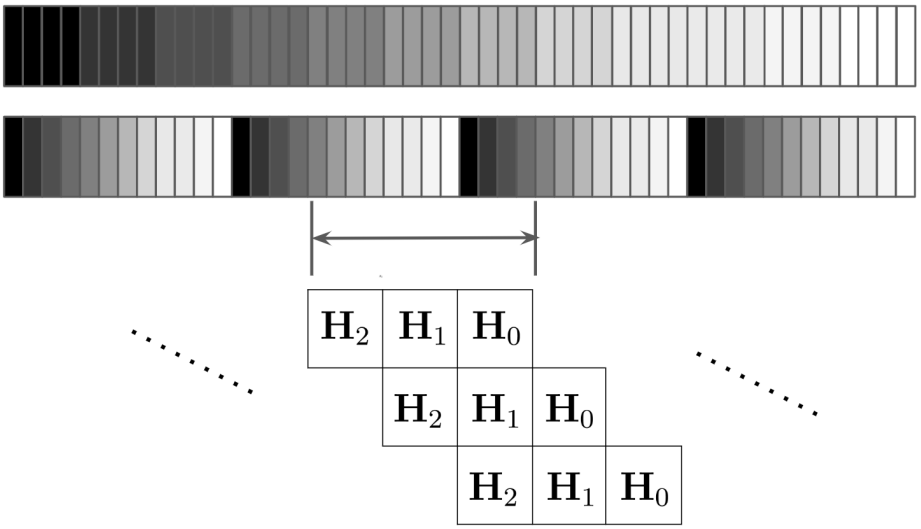



Figure 3.10: The original sequence of data (top panel) represents the worst case scenario where multiple consecutive sections are affected by a low SNR. Interleaved sequence of data (middle panel) for an SC code with $m = 2$ and $L = 12$: a darker gray represents a lower SNR. These chunks are interleaved such that each check equation receives messages with all $L$ different reliabilities.

impact of channel non-uniformity.

#### 3.4.4.2 Interleaving for SC Codes

In this part, we introduce a low-cost interleaving scheme specialized for SC codes in order to further improve the performance. In the presented code design machinery, CNs receive messages from $(m + 1)$ different sections. If most of these $(m + 1)$ sections have a low SNR, the decoder may fail to recover the message correctly. We use interleaving to avoid letting multiple unreliable VNs from consecutive sections participate in one check equation and degrade the performance.

Consider an SC code with the coupling length $L$ ($L$ is also equal to the number of sections covered by one SC codeword) and memory $m$. We assume that $(m + 1)$ divides $L$. We divide the SC codeword into $L$ groups, and we further divide each group of data into $L/(m + 1)$ chunks. Then, we rearrange them by taking one chunk from each group in order and placing them next to each other, see Figure 3.10. This interleaved data is passed through the channel, and the de-interleaving is performed on the received data from the channel and

before decoding. Due to interleaving, each CN receives equal number of messages from all $L$ different levels of reliabilities (except for the first and last groups of CNs).

**Claim 1.** *Consider an SC code with parameters $m$ and $L$. Let $\mu$ be the number of chunks that an SC codeword from this code is partitioned into for the interleaving step. Then, the smallest value of $\mu$ that provides the same average reliability for all CNs, except for the first and last few CNs that have smaller degrees, is $\mu_{\text{opt}} = L^2/(m+1)$.*

*Proof.* We prove Claim 1 by the contradiction. Suppose an SC codeword from this code is partitioned into $\mu < L^2/(m+1)$ chunks. In an iterative decoding of an SC codeword, except for the first and last few CNs, all CNs receive messages from a contiguous $(m+1)/L$ fraction of all VNs. In other words, they receive messages from $\nu = \mu(m+1)/L < L$ consecutive chunks. Suppose that the interleaving is perfect in the sense that all these $\nu$ chunks have different SNR values. Then, the number of different SNR values that a CN is affected by, i.e., $\nu$, is less than the total number of different SNR values, i.e., $L$. Consequently, some CNs may receive messages from more reliable VNs while others do not, and the average SNR need not be the same for check equations. □

Our presented interleaving scheme partitions an SC codeword into $\mu_{\text{opt}} = L^2/(m+1)$ chunks, and ensures that all check equations, except for the first and last few ones, experience the same average SNR. For example, an SC codeword with $m = 2$ and $L = 12$, shown in Figure 3.10, is partitioned into $\mu_{\text{opt}} = 48$ chunks. By using interleaving, we avoid letting multiple unreliable sections dominate the messages received by one CN. Our simulation results show that interleaving notably reduces the performance gap that exists between the error rates of SC codes over non-uniform and uniform channels. We note that using block codes and traditional interleaving requires dividing the data with the same length into $L^2$ chunks.

The performance of the SC code constructed using our technique is not sensitive to the perfect alignment of underlying codes and sections of the channel. In fact, a substantial mis-alignment resembles an uninformed interleaving that neither helps nor degrades the performance.

### 3.4.5 Simulation Results

First, we show the performance gap that exists over channels with uniform and non-uniform SNR for uncoupled block codes. Next, we compare the performance of SC codes possessing the introduced structure with block codes, and illustrate the effect of interleaving and increasing the memory. For simulations, we use a min-sum algorithm with a maximum of 50 iterations for decoding.

Our code parameters are as follows. Block Code 3.4 is a binary AB block code with $\kappa = z = 17$ and $\gamma = 3$. Uncoupled Block Codes 3.4 represents 30 uncoupled Block Codes 3.4, and it can be viewed as an SC code with $m = 0$, $\mathbf{H}_0 = \mathbf{H}$, $L = 30$, length 8,670 bits, and rate 0.824. SC Code 3.10 and SC Code 3.11 are binary SC codes with $\gamma = 3$, $\kappa = z = 17$, $L = 30$, MO partitioning, and AB circulant powers. Thus, these two codes have length 8,670 bits. The memory and rate for SC Code 3.10 are $m = 1$ and 0.818, and for SC Code 3.11 are $m = 2$ and 0.812, respectively. The partitioning matrix for SC Codes 3.10 is:

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix},$$

and the partitioning matrix for SC Codes 3.11 is:

$$\begin{bmatrix} 0 & 1 & 0 & 2 & 2 & 1 & 2 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 1 & 2 & 2 \\ 1 & 0 & 2 & 1 & 0 & 2 & 1 & 1 & 1 & 2 & 2 & 0 & 2 & 0 & 0 & 0 & 1 \\ 2 & 2 & 1 & 0 & 1 & 0 & 0 & 2 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 1 & 0 \end{bmatrix}.$$

SC Codes 3.12 is a binary SC code with $\gamma = 6$, $\kappa = 29$, $z = 61$, $m = 2$, $L = 6$, MO partitioning, and circulant powers attained by the CPO program. Thus, it has length 10,614 bits and rate 0.724. According to Theorem 3, the length of each stripe (except for the last stripe) is $\omega_{\max} = \binom{6}{2} = 15$, and the number of stripes is $\left\lceil \frac{\kappa}{\omega_{\max}} \right\rceil = 2$. As a result, $t_{\min} = 2$.

The partitioning matrix that achieves the minimum overlap value is:

$$\begin{bmatrix}
1 & 1 & 2 & 1 & 0 & 1 & 2 & 2 & 0 & 2 & 1 & 0 & 2 & 0 & 0 & 1 & 1 & 2 & 1 & 0 & 1 & 2 & 2 & 0 & 2 & 1 & 0 & 2 & 0 \\
1 & 2 & 1 & 0 & 1 & 2 & 1 & 0 & 2 & 2 & 0 & 2 & 0 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 2 & 1 & 0 & 2 & 2 & 0 & 2 & 0 & 1 \\
2 & 1 & 0 & 2 & 1 & 2 & 0 & 2 & 1 & 0 & 2 & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 2 & 1 & 2 & 0 & 2 & 1 & 0 & 2 & 1 & 0 & 0 \\
2 & 0 & 1 & 1 & 2 & 0 & 2 & 1 & 2 & 0 & 0 & 0 & 1 & 2 & 1 & 2 & 0 & 1 & 1 & 2 & 0 & 2 & 1 & 2 & 0 & 0 & 0 & 1 & 2 \\
0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2
\end{bmatrix},$$

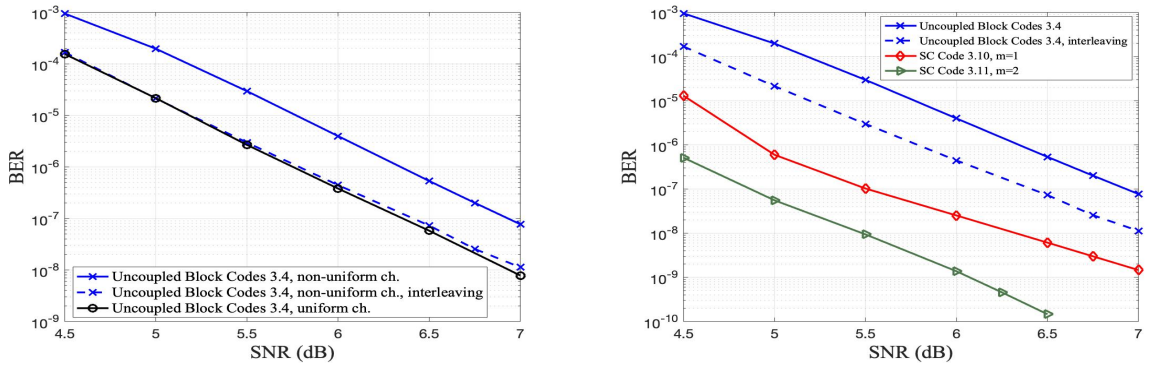
and the circulant power matrix obtained by the CPO program is given below:

$$\begin{bmatrix}
4 & 55 & 59 & 49 & 10 & 4 & 59 & 14 & 17 & 44 & 12 & 58 & 55 & 27 & 11 & 15 & 16 & 15 & 7 & 0 & 36 & 0 & 0 & 22 & 41 & 0 & 0 & 0 & 24 \\
20 & 28 & 28 & 3 & 15 & 5 & 54 & 7 & 8 & 9 & 49 & 11 & 12 & 13 & 14 & 15 & 16 & 59 & 18 & 52 & 30 & 21 & 22 & 23 & 24 & 25 & 26 & 50 & 34 \\
0 & 14 & 4 & 6 & 8 & 10 & 11 & 14 & 16 & 18 & 20 & 22 & 12 & 19 & 32 & 30 & 1 & 34 & 36 & 38 & 40 & 42 & 44 & 46 & 48 & 42 & 26 & 54 & 56 \\
8 & 3 & 6 & 9 & 12 & 15 & 57 & 32 & 24 & 27 & 24 & 55 & 57 & 39 & 42 & 45 & 48 & 51 & 17 & 57 & 60 & 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\
0 & 4 & 8 & 8 & 16 & 20 & 24 & 28 & 32 & 4 & 40 & 44 & 18 & 52 & 56 & 60 & 27 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 17 & 39 & 5 & 47 & 51 \\
0 & 5 & 10 & 17 & 20 & 25 & 30 & 37 & 40 & 13 & 50 & 55 & 57 & 4 & 3 & 14 & 19 & 36 & 49 & 28 & 20 & 47 & 49 & 31 & 59 & 3 & 8 & 13 & 18
\end{bmatrix}.$$

For Uncoupled Block Codes 3.4, SC Code 3.10, and SC Code 3.11, we use a non-uniform channel with $N = 30$ sections. For SC Code 3.12, we use a non-uniform channel with $N = 6$ sections. For uncoupled block codes, we use the interleaving technique that is studied in Section 3.4.3, and for SC codes, we use the interleaving scheme introduced in Section 3.4.4. In a practical channel model, the values of $\Delta\text{SNR}_s$ for different sections are dependent. We describe the correlation model among SNR values of different sections as follows. Let $\mathbf{a}$ be an $\alpha \times 1$ vector that describes the correlation coefficients among $\alpha$ consecutive sections:

$$(\Delta\text{SNR}_s)_{\text{dB}} = \mathbf{a}^{\text{T}}\mathbf{u}, \quad \mathbf{u} = [u_s \; u_{s-1} \; \cdots \; u_{s-\alpha+1}]^{\text{T}}, \quad u_i \sim \mathcal{N}(0, \sigma^2).$$

$\mathcal{N}(0, \sigma^2)$ defines a Gaussian distribution with mean 0 and variance $\sigma^2$. In our simulations,




(a) (b)

Figure 3.11: (a) BER curves for Uncoupled Block Codes 3.4 over uniform and non-uniform channels with/without interleaving. (b) BER curves over non-uniform channel for Uncoupled Block Codes 3.4 with/without interleaving versus SC code 3.10 and SC code 3.11.

we consider the following parameters for the channel:[3]

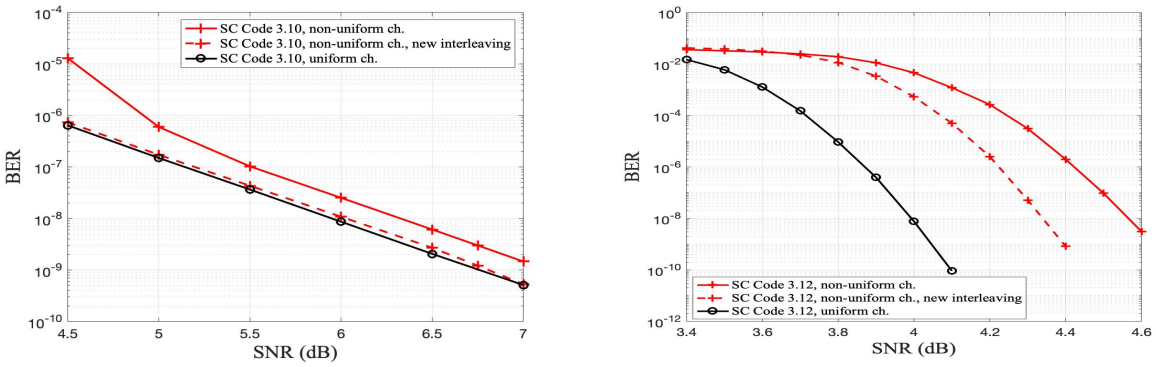
$$\mathbf{a} = [0.78\ 0.44\ 0.31\ 0.23\ 0.15\ 0.08\ 0.09\ 0.03\ 0.04\ 0.02 {-} 0.04]^{\mathrm{T}},\ \alpha = 11,\ \sigma = 0.15.$$

Figure 3.11 (a) shows the BER curves for Uncoupled Block Codes 3.4 over non-uniform and uniform channels with the same average SNR. Because of the SNR variation, the performance over non-uniform channel is around 1 order of magnitude worse in the error floor region. Then, we apply the traditional interleaving for the non-uniform channel. It can be seen, as expected, that interleaving compensates for the performance loss due to the SNR variation. (For the non-uniform channel, the horizontal axis represents $\mathrm{SNR}_{\mathrm{abs}}$.)

Next, we compare the error floor performance of Uncoupled Block Codes 3.4 with SC codes over channels with SNR variation. According to Figure 3.11 (b), SC Code 3.10 shows 1 and 2 orders of magnitude performance improvement compared to the Uncoupled Block Codes 3.4 with and without interleaving, respectively, while it has a comparable decoding latency. Moreover, SC Code 3.11 secures even further improvement by providing more diversity.

Figure 3.12 (a) demonstrates the performance loss due to SNR variation for SC Code 3.10.

[3]On a hard disk drive, under certain environmental and vibration conditions, BER for each of many consecutive sectors around a circular track can be measured. The autocorrelation terms gauge the coupling of SNR between sectors of various spacing. The autocorrelation model used in this paper come from experimental data provided by a Western Digital Company (WDC).




(a) (b)

Figure 3.12: BER curves over uniform and non-uniform channels for: (a) SC Code 3.10 with/without interleaving. (b) SC Code 3.12 with/without interleaving.

The performance loss at SNR = 6.5 dB is around 0.5 of an order of magnitude. Compared to the performance loss for Uncoupled Block Codes 3.4, we immediately see that well-designed SC codes are more robust against SNR variation. Furthermore, this loss can be compensated by the introduced interleaving, as Figure 3.12 (a) shows.

Figure 3.12 (b) demonstrates the similar analysis for SC Code 3.12. Since well-designed SC codes with $\gamma = 6$ have a very low error floor, we could not collect enough errors in the error floor region. As we see in this figure, our well-designed SC code with column weight $\gamma = 6$ constructed by the MO partitioning has a sharp waterfall region, and it achieves BER $= 10^{-10}$ at SNR $= 4.1$ dB. We also note that there is a notable performance gap between uniform and non-uniform channels for this code which is because of the high SNR variation of the channel with $N = 6$ sections and the relatively low coupling length of the SC code ($L = 6$). However, our efficient interleaving scheme remarkably reduces this gap.

## 3.5 Conclusion

In this chapter, we investigated non-AWGN channels and presented channel-aware SC constructions. In Section 3.2, we considered one-dimensional MR channels and presented a systematic design methodology for NB SC codes that exploits the structure of dominant errors for setting the partitioning, circulant powers, and edge weights. We demonstrated

significant performance advantage of our proposed optimized SC codes relative to their block counterparts as well as relative to other SC constructions.

In Section 3.3, we considered Flash memories and proposed a systematic approach for the design of NB SC codes optimized for practical Flash channels. The SC codes designed using our approach have reduced number of dominant GASTs as the detrimental objects for Flash channels, thus outperforming existing NB SC codes.

In Section 3.4, we presented an SC code design for channels with SNR variation. First, we introduced a new construction method for SC codes with a low computational complexity. Second, we demonstrated that our well-designed SC codes outperform block codes thanks to the diversity that is provided by the coupling, while they have comparable decoding latency as their underlying block codes. Finally, we demonstrated that the performance of SC codes can be further improved by increasing the memory and performing the introduced interleaving scheme that is specific to SC codes.

## Acknowledgment

The material of this chapter have been published in several papers [13–15, 40, 43]. The author would like to thank all the collaborators in these publications. The author would like to especially thank Dr. Ahmed Hareedy, who is also the first author in [14, 40], for the great collaboration on the materials that appeared in this chapter.

# CHAPTER 4

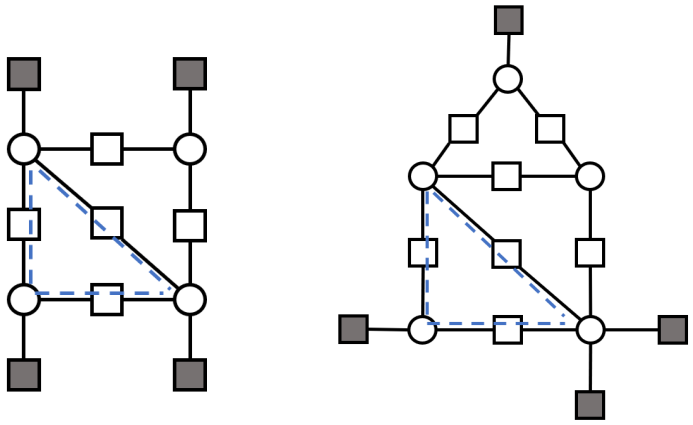
# Extending the Construction Framework for Irregular SC Codes

## 4.1 Introduction

It has long been known that graph-based LDPC codes with irregular node degree distribution offer performance advantage over their regular-degree counterparts [44]. This observation has led to the construction of many irregular block LDPC codes with excellent properties, e.g., [45], and design of capacity approaching performance of highly irregular LDPC codes using density evolution techniques [46]. While SC codes intrinsically possess a small amount of node degree irregularity due to the termination effects, which aides in performance improvement [6], it is nonetheless customary to use regular underlying block to construct finite-length SC codes.

In this chapter, we present a novel combinatorial framework for the finite-length analysis and design of irregular SC codes. Our irregular SC codes have the desirable properties of regular SC codes thanks to their structure while offering significant performance benefits that come with the degree irregularity. We still use CB codes as the underlying block codes. We present a systematic scheme for optimal partitioning of the underlying block codes and constructing irregular SC codes with a superior performance in the error floor area. We show that this scheme has significantly better performance than random code constructions with the same node degree distribution. The performance advantage is achieved by explicitly organizing and combining circulants such that the population of graphical objects that are problematic for decoding is minimized.

Certain structures in the graph of LDPC codes are responsible for most errors that occur



(a) (b)

Figure 4.1: Examples of ASs in graph of irregular LDPC codes along with their common denominator structure shown with dashed blue lines; (a) The $(4, 4)$ AS. (b) The $(5, 4)$ AS.

under the iterative decoding in the error floor region. Among the problematic graphical structures, the most harmful ones are ASs [23], see Definition 3. ASs are problematic objects in graph of irregular LDPC codes as well. Figure 4.1 shows two examples of ASs in the graph of an irregular LDPC code. Based on our empirical results, we have identified that cycles-6 are the common denominator instances of most problematic ASs over AWGN channels for irregular SC codes with girth 6.

We remind that focusing on minimizing the population of instances of a small common substructure, such as cycle-6, in the code design notably reduces the computational complexity and improves the performance as discussed in the previous chapters. This observation follows from noting that the common denominator has a simpler graphical structure, and it exists in the graph of several detrimental ASs.

The rest of this chapter is organized as follows. In Section 4.2, we show how irregular SC codes can be constructed by a new partitioning scheme that takes the irregularity into account. In Section 4.3, we present our new scheme for constructing irregular SC codes through optimal partitioning, which builds in part on the OO partitioning that was introduced in Chapter 2 for regular SC codes. Simulation results are presented in Section 4.4. Section 4.5 delivers conclusions.

## 4.2 Irregular SC Code Construction

As stated, regular SC codes are constructed by partitioning the circulants of a CB block code with non-zero circulants into $(m+1)$ component matrices. For constructing irregular SC codes, one can choose an irregular underlying block code, i.e., a CB code with zero/non-zero circulants, for the partitioning. The resulting SC code has the same VN and CN degree distribution as the underlying block code, except for the first and last few CNs that have lower degrees.

In this section, we present a new procedure for constructing irregular SC codes that creates degree irregularity in the partitioning stage. For this purpose, we introduce a new component matrix, which is called the *dummy component matrix* and is denoted by $\mathbf{H}_{\mathrm{d}}$. While $\mathbf{H}_{\mathrm{d}}$ is treated in the partitioning like the other component matrices,

$$\mathbf{H} = \mathbf{H}_{\mathrm{d}} + \sum_{y=0}^{m} \mathbf{H}_y, \tag{4.1}$$

the elements assigned to $\mathbf{H}_{\mathrm{d}}$ are discarded, and $\mathbf{H}_{\mathrm{d}}$ does not appear in the parity-check matrix $\mathbf{H}_{\mathrm{SC}}$ of the irregular SC code, see Figure 1.1.

We remind that the protograph of a CB matrix is denoted by the super-script p, i.e., $(.)^{\mathrm{p}}$. The degree distribution which indicates the ratio of VNs (resp., CNs) with a specific degree value, is the same for a CB block/SC code and its protograph. Therefore, we define and evaluate the degree distribution for the protograph. The VN (resp., CN) degree vector for the protograph of an irregular SC code is denoted by $\Lambda = [\lambda_0 \ \ldots \ \lambda_{\gamma-1}]$ (resp., $\Phi = [\rho_0 \ \ldots \ \rho_{\kappa-1}]$). Here, $\lambda_i$ (resp., $\rho_j$) is the portion of VNs (resp., CNs) with degree $i+1$ (resp., $j+1$). Parameter $\gamma$ (resp., $\kappa$) is the maximum VN (resp., CN) degree. The protograph of the dummy component matrix, i.e., $\mathbf{H}_{\mathrm{d}}^{\mathrm{p}}$, determines the degree distribution of the final SC code, as shown by Lemma 8.

**Lemma 8.** *Let $\alpha_u$, i.e., $0 \leq u \leq \gamma-1$, be the number of 1s in row $u$ of $\mathbf{H}_{\mathrm{d}}^{\mathrm{p}}$. Similarly, let $\beta_v$, i.e., $0 \leq v \leq \kappa-1$, be the number of 1s in column $v$ of $\mathbf{H}_{\mathrm{d}}^{\mathrm{p}}$. Then, $\forall i \in \{0,\ldots,\gamma-1\}$*

*and* $\forall j \in \{0, \ldots, \kappa - 1\}$,

$$\lambda_i = \frac{|\{\beta_v | 0 \leq v \leq \kappa - 1, \beta_v = \gamma - i - 1\}|}{\kappa}, \tag{4.2}$$

$$\rho_j \approx \frac{|\{\alpha_u | 0 \leq u \leq \gamma - 1, \alpha_u = \kappa - j - 1\}|}{\gamma}. \tag{4.3}$$

*Proof.* First, we derive the elements of the vector $\Lambda$. All replicas of $\mathbf{H}_{\text{SC}}^{\text{p}}$ have the same non-zero parts. Thus, we only need to consider one replica, say $\mathbf{R}_1^{\text{p}}$, to derive the VN degree distribution. The $v$'th column in $\mathbf{R}_1^{\text{p}}$, where $0 \leq v \leq \kappa - 1$, has degree $\gamma - \beta_v$, see Figure 1.1 and (4.1). Therefore, the number of columns in $\mathbf{R}_1^{\text{p}}$ with degree $i + 1$ is $|\{\beta_v | 0 \leq v \leq \kappa - 1, \beta_v = \gamma - i - 1\}|$. There are $\kappa$ columns in $\mathbf{R}_1^{\text{p}}$, and the expression in (4.2) follows. The elements of the vector $\Phi$, i.e., (4.3), can be derived similarly. The approximation sign in (4.3) is due to the fact that the first and the last group of CNs have lower degree due to the SC structure. □

**Example 8.** *Assume the code parameters* $\kappa = 7$ *and* $\gamma = 3$. *We seek to construct* $\mathbf{H}_{\text{d}}^{\text{p}}$ *to achieve* $\Lambda = [0\ 3/7\ 4/7]$ *and* $\Phi \approx [0\ 0\ 0\ 0\ 0\ 1\ 0]$. *Using (4.2) and (4.3),* $\mathbf{H}_{\text{d}}^{\text{p}}$ *has* 3 *columns with degree* 1, 4 *columns with degree* 0, *and* 3 *rows with degree* 1. *With these properties, one realization for* $\mathbf{H}_{\text{d}}^{\text{p}}$ *is:*

$$\mathbf{H}_{\text{d}}^{\text{p}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The node degree distributions, and consequently the matrix $\mathbf{H}_{\text{d}}^{\text{p}}$, can be derived using the density evolution techniques to attain better threshold performance. In the next section, we present a new scheme for optimal partitioning of the circulant of $\mathbf{H}$ into the $(m + 1)$ component matrices, considering the circulants that are already assigned to $\mathbf{H}_{\text{d}}$, in order to minimize the number of cycles-6 in the protograph of an irregular SC code.

## 4.3 Optimal Partitioning for Irregular SC Codes

In this section, we extend the OO partitioning scheme, introduced in Chapter 2, for constructing irregular SC codes. For simplicity, we consider $m = 1$ throughout this section. Thus, we intend to partition $\kappa\gamma$ circulants in matrix $\mathbf{H}$ of an underlying block code into three component matrices $\mathbf{H}_0$, $\mathbf{H}_1$, and $\mathbf{H}_{\rm d}$, and piece $L$ copies of $\mathbf{H}_0$ and $\mathbf{H}_1$ in a diagonal structure to construct the parity-check matrix $\mathbf{H}_{\rm SC}$ of an irregular SC code. We note that the circulants that are assigned to $\mathbf{H}_{\rm d}$ are fixed given the degree distribution, and we aim to find the best partitioning of the remaining circulants between $\mathbf{H}_0$ and $\mathbf{H}_1$ such that the number of cycles-6 in the protograph of the SC code is minimized.

The overlap parameters are a set of integer-valued parameters that include all necessary information needed to find the population of combinatorial objects, e.g., cycles-6, in the protograph of a regular SC code, see Definition 8. Here, we extend the definition of the overlap parameters for irregular SC codes. A careful selection of the overlap parameters corresponds to constructing high performance irregular SC codes.

A cycle-6 in the graph of $\mathbf{H}_{\rm SC}^{\rm p}$ results in either $z$ or 0 cycles-6 in the graph of $\mathbf{H}_{\rm SC}$ depending on the powers of the circulants associated with that cycle [29, 30]. Moreover, a cycle-6 in the final (lifted) graph of an SC code can only be generated from a cycle-6 in the protograph. Motivated by the above fact, our optimal partitioning aims at deriving the overlap parameters that result in the minimum number of cycles-6 in the graph of $\mathbf{H}_{\rm SC}^{\rm p}$ with girth 6.

We establish a discrete optimization problem by expressing the number of cycles-6 in the graph of $\mathbf{H}_{\rm SC}^{\rm p}$ as a function of the overlap parameters and standard code parameters. We first review and extend the definition of overlap parameters for an irregular SC code.

**Definition 10.** *Define matrix* $\mathbf{\Pi}^{\rm p}$ *of size* $3\gamma \times \kappa$ *as follows:*

$$\mathbf{\Pi}^{\rm p} = \begin{bmatrix} \mathbf{H}_0^{\rm p} \\ \mathbf{H}_1^{\rm p} \\ \mathbf{H}_{\rm d}^{\rm p} \end{bmatrix}. \tag{4.4}$$

*A degree-d overlap parameter* $t_{\{i_1,\cdots,i_d\}}$, $0 \leq i_1, \cdots, i_d \leq 3\gamma - 1$, *is defined as the overlaps among* $d$ *distinct rows of* $\mathbf{\Pi}^{\mathrm{p}}$ *specified by the set* $\{i_1, \cdots, i_d\}$, *i.e., the number of positions in which all the* $d$ *rows simultaneously have* 1*s.*

Similar to the case of regular SC codes, for an irregular SC code with maximum VN degree $\gamma$, the maximum degree for an overlap parameter with non-zero value is $\gamma$. Additionally, if there is at least one pair of distinct row indices $(i_u, i_v)$ such that $i_u, i_v \in \{i_1, \cdots, i_d\}$ and $i_u = i_v \pmod{\gamma}$, then $t_{\{i_1,\cdots,i_d\}} = 0$.

**Lemma 9.** *The set of independent non-deterministic overlap parameters* $\mathcal{O}_{\mathrm{ind}}$ *is defined as follows:*

$$\begin{aligned}\mathcal{O}_{\mathrm{ind}} =&\{t_{\{i_1,\cdots,i_d\}} | 1 \leq d \leq \gamma, \gamma \leq i_1, \cdots, i_d \leq 3\gamma - 1, \\ &\forall \{i_u, i_v\} \subset \{i_1, \cdots, i_d\}\, i_u \neq i_v \pmod{\gamma}\} \backslash \\ &\{t_{\{i_1,\cdots,i_d\}} | 1 \leq d \leq \gamma, 2\gamma \leq i_1, \cdots, i_d \leq 3\gamma - 1\}.\end{aligned} \tag{4.5}$$

*The other overlap parameters that are not in* $\mathcal{O}_{\mathrm{ind}}$ *are either deterministic (zero or determined by the desired node degree distribution) or functions of the overlap parameters in* $\mathcal{O}_{\mathrm{ind}}$, *as follows:*

1. *Let* $2\gamma \leq i_1, \cdots, i_d \leq 3\gamma - 1$. *Then,* $t_{\{i_1,\cdots,i_d\}}$ *is determined based on the desired degree distribution.*

2. *Let* $0 \leq i_1, \cdots, i_{d_1} \leq \gamma - 1$, $\gamma \leq j_1, \cdots, j_{d_2} \leq 3\gamma - 1$, *and* $1 \leq (d_1 + d_2) \leq \gamma$. *Then,* $t_{\{i_1,\cdots,i_{d_1},j_1,\cdots,j_{d_2}\}}$ *is a linear function of the overlap parameters in* $\mathcal{O}_{\mathrm{ind}}$*:*

$$t_{\{i_1,\ldots,i_{d_1},j_1,\ldots,j_{d_2}\}} = t_{\mathcal{J}} + \sum_{\alpha=1}^{d_1} (-1)^{\alpha} \sum_{\substack{\{i'_1,\ldots,i'_\alpha\} \subset \mathcal{I} \\ [x_1 \ \ldots \ x_\alpha] \in \{1,2\}^\alpha}} t_{\mathcal{J} \cup \{x_1\gamma + i'_1, \ldots, x_\alpha\gamma + i'_\alpha\}}, \tag{4.6}$$

   *where* $\mathcal{I} = \{i_1, \ldots, i_{d_1}\}$, $\mathcal{J} = \{j_1, \ldots, j_{d_2}\}$, *and in the case of* $\mathcal{J} = \varnothing$, $t_{\mathcal{J}} = \kappa$.

*Proof.*

1. Given the degree distribution, the dummy component matrix $\mathbf{H}_{\mathrm{d}}$, its protograph, and its overlap parameters are determined.

2. To have an overlap at position (column index) $y \in \{1, \dots \kappa\}$ among the rows $\mathcal{I} \cup \mathcal{J}$ of $\mathbf{\Pi}^{\mathrm{p}}$: a) the rows in $\mathcal{J}$ of $\mathbf{\Pi}^{\mathrm{p}}$ must have 1s at position $y$, b) the rows in the second and third component matrices of $\mathbf{\Pi}^{\mathrm{p}}$, i.e., $\mathbf{H}_1^{\mathrm{p}}$ and $\mathbf{H}_{\mathrm{d}}^{\mathrm{p}}$, that correspond to the rows in $\mathcal{I}$ must have 0s at position $y$. In other words, the rows in $\{x_1\gamma + i_1, \dots, x_{d_1}\gamma + i_{d_1}\}$ of $\mathbf{\Pi}^{\mathrm{p}}$ must have 0s at position $y$, where $[x_1 \ \dots \ x_{d_1}] \in \{1, 2\}^{d_1}$. Aided by the principle of inclusion and exclusion, (4.6) follows.

□

**Example 9.** *For an irregular SC code with $m = 1$, $\gamma = 3$, and $\kappa = 7$,*

$$\mathcal{O}_{\mathrm{ind}} = \{t_3, t_4, t_5, t_{\{3,4\}}, t_{\{3,5\}}, t_{\{4,5\}}, t_{\{3,7\}}, t_{\{3,8\}}, t_{\{4,6\}},$$
$$t_{\{4,8\}}, t_{\{5,6\}}, t_{\{5,7\}}, t_{\{3,4,5\}}, t_{\{3,4,8\}}, t_{\{3,5,7\}}, t_{\{3,7,8\}}, t_{\{4,5,6\}}, t_{\{4,6,8\}}, t_{\{5,6,7\}}\}.$$

*Let $\Lambda = [0\ 3/7\ 4/7]$ and $\Phi \approx [0\ 0\ 0\ 0\ 0\ 1\ 0]$ as in Example 8. According to Lemma 9, the overlap parameters that are not in $\mathcal{O}_{\mathrm{ind}}$ are deterministic or functions of the overlap parameters in $\mathcal{O}_{\mathrm{ind}}$. For example,*

$$\begin{aligned} t_{\{6\}} &= 1, & t_{\{1\}} &= 6 - t_4, \\ t_{\{7,8\}} &= 0, & t_{\{0,2\}} &= 5 - t_{\{3\}} - t_{\{5\}} + t_{\{3,5\}} + t_{\{3,8\}} + t_{\{5,6\}}, \\ t_{\{6,7,8\}} &= 0, & t_{\{1,3\}} &= t_{\{3\}} - t_{\{3,4\}} - t_{\{3,7\}}. \end{aligned}$$

**Lemma 10.** *The number of independent non-deterministic overlap parameters is $\mathcal{N}_{\mathrm{ind}} = |\mathcal{O}_{\mathrm{ind}}| = 3^\gamma - 2^\gamma$*

*Proof.* Let the sets $T_1$ and $T_2$ be defined as follows:

$$\begin{aligned} T_1 =& \{t_{\{i_1,\cdots,i_d\}} | 1 \le d \le \gamma, \gamma \le i_1, \cdots, i_d \le 3\gamma - 1, \\ & \forall \{i_u, i_v\} \subset \{i_1, \cdots, i_d\}\ i_u \ne i_v \pmod{\gamma}\}, \\ T_2 =& \{t_{\{i_1,\cdots,i_d\}} | 1 \le d \le \gamma, 2\gamma \le i_1, \cdots, i_d \le 3\gamma - 1\}. \end{aligned}$$

Since $\mathcal{O}_{\text{ind}} = T_1 \setminus T_2$, see (4.5), and $T_2 \subset T_1$, $\mathcal{N}_{\text{ind}} = |\mathcal{O}_{\text{ind}}| = |T_1| - |T_2|$. $|T_1|$ is the number of non-empty subsets of the set $S = \{\gamma, \ldots, 3\gamma - 1\}$ with maximum size $\gamma$ such that no two elements in a subset have the same value mod $\gamma$. We partition the set $S$ into $\gamma$ disjoint sets $\{\gamma, 2\gamma\}, \{\gamma + 1, 2\gamma + 1\}, \ldots, \{2\gamma - 1, 3\gamma - 1\}$. The two elements in each of these partitions have the same value mod $\gamma$. Thus, we need to pick at most one element from each partition to form the set $T_1$, and there are $3^\gamma - 1$ choices for this (selection of the first element, second element, or neither of elements for each partition; the case where none of the partitions lends an element to the subset, i.e., empty subset, must be excluded). $|T_2|$ is the number of non-empty subsets of the set $\{2\gamma, \ldots, 3\gamma - 1\}$, i.e., $2^\gamma - 1$. As a result, $\mathcal{N}_{\text{ind}} = 3^\gamma - 2^\gamma$. $\square$

For example for $\gamma = 3$, $\mathcal{N}_{\text{ind}} = 19$. Next, we show that the number of cycles-6 in the protograph of an irregular SC code can be expressed as a function of the parameters in $\mathcal{O}_{\text{ind}}$.

**Theorem 4.** *The number of cycles-*6 *in the protograph of an irregular SC code with parameters* $m = 1$*,* $L$*,* $\kappa$*,* $\gamma$*, and* $\mathcal{O}_{\text{ind}}$ *is:*

$$F = LF_1^1 + (L-1)F_1^2, \tag{4.7}$$

*and* $F_1^1$ *and* $F_1^2$ *are:*

$$F_1^1 = \sum_{\substack{\{i_1,i_2,i_3\} \subset \{0,\ldots,2\gamma-1\} \\ \bar{i}_1 \neq \bar{i}_2, \bar{i}_1 \neq \bar{i}_3, \bar{i}_2 \neq \bar{i}_3}} \mathcal{A}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2,i_3\}}),$$

$$F_1^2 = \sum_{\substack{i_1 \in \{0,\ldots,2\gamma-1\} \\ \{i_2,i_3\} \subset \{\gamma,\ldots,2\gamma-1\} \\ \bar{i}_1 \neq \bar{i}_2, \bar{i}_1 \neq \bar{i}_3, \bar{i}_2 \neq \bar{i}_3}} \mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2-\gamma,i_3-\gamma\}})$$

$$+ \sum_{\substack{i_1 \in \{0,\ldots,2\gamma-1\} \\ \{i_2,i_3\} \subset \{0,\ldots,\gamma-1\} \\ \bar{i}_1 \neq \bar{i}_2, \bar{i}_1 \neq \bar{i}_3, \bar{i}_2 \neq \bar{i}_3}} \mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2+\gamma,i_3+\gamma\}}),$$

*where* $\bar{i} = (i \bmod \gamma)$*. The functions* $\mathcal{A}$ *and* $\mathcal{B}$ *are are defined in (2.12) and (2.13), and are*

*reminded here:*

$$\begin{aligned}
&\mathcal{A}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2,i_3\}}) \\
&\qquad = \big(t_{\{i_1,i_2,i_3\}}[t_{\{i_1,i_2,i_3\}} - 1]^+[t_{\{i_2,i_3\}} - 2]^+\big) \\
&\qquad + \big(t_{\{i_1,i_2,i_3\}}(t_{\{i_1,i_3\}} - t_{\{i_1,i_2,i_3\}})[t_{\{i_2,i_3\}} - 1]^+\big) \\
&\qquad + \big((t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})t_{\{i_1,i_2,i_3\}}[t_{\{i_2,i_3\}} - 1]^+\big) \\
&\qquad + \big((t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})(t_{\{i_1,i_3\}} - t_{\{i_1,i_2,i_3\}})t_{\{i_2,i_3\}}\big)\,.
\end{aligned}$$

$$\begin{aligned}
&\mathcal{B}(t_{\{i_1,i_2,i_3\}}, t_{\{i_1,i_2\}}, t_{\{i_1,i_3\}}, t_{\{i_2\pm\gamma,i_3\pm\gamma\}}) \\
&\qquad = \big(t_{\{i_1,i_2,i_3\}}[t_{\{i_1,i_3\}} - 1]^+ t_{\{i_2\pm\gamma,i_3\pm\gamma\}}\big) \\
&\qquad + \big((t_{\{i_1,i_2\}} - t_{\{i_1,i_2,i_3\}})t_{\{i_1,i_3\}}t_{\{i_2\pm\gamma,i_3\pm\gamma\}}\big)\,.
\end{aligned}$$

*Proof.* The proof has the same flow as the proof of Theorem 2 for the case $m = 1$ because the construction shown in Figure 1.1 is the same for both regular and irregular SC codes. The set of overlap parameters needed to calculate $F$ in (4.7) is a subset of $\mathcal{O}_{\text{ind}}$. The overlap parameters that are related to the dummy component matrix $\mathbf{H}_{\text{d}}^{\text{p}}$ do not directly play a role in finding $F$, and they only help to exploit the dependency between the overlap parameters for the protograph of $\mathbf{\Pi}_1^1 = [\mathbf{H}_0^T\ \mathbf{H}_1^T]^T$. Once the overlap parameters for $\mathbf{\Pi}_1^{1,\text{p}}$ are determined, $F$ can be found using (4.7). □

Let $F^*$ be the minimum number of cycles-6 in $\mathbf{H}_{\text{SC}}^{\text{p}}$. Thus, our discrete optimization problem is:

$$F^* = \min_{\mathcal{O}_{\text{ind}}} F. \tag{4.8}$$

Consider an underlying block code with parameters $\kappa$ and $\gamma$. In the partitioning, each circulant of the matrix of the underlying block code, i.e., $\mathbf{H}$, that is not assigned to the dummy component matrix $\mathbf{H}_{\text{d}}$ can be assigned to $\mathbf{H}_0$ or $\mathbf{H}_1$, resulting in roughly $2^{\kappa\gamma}$ possible options. The goal is to choose a partitioning that results in the lowest number of cycles-6 in the protograph of an irregular SC code. We reduced the problem of finding the optimal partitioning for irregular SC codes into an optimization problem over $\mathcal{N}_{\text{ind}} = 3^\gamma - 2^\gamma$ integer-

valued overlap parameters.

As the second (optional) step of constructing irregular SC codes, we run the heuristic CPO program, introduced in Section 2.3.2, for adjusting the powers of non-zero circulants in one replica of $\mathbf{H}_{\mathrm{SC}}$. We remind that each change that is made to one replica is also applied to all other replicas to preserve the structure described in Figure 1.1. The CPO program adjusts the powers of the problematic circulants, i.e., those that are involved in the most cycles-6, to break the necessary condition for as many remaining cycles-6 as possible without creating cycles-4.

## 4.4 Simulation Results

In this section, we compare the performance of our irregular SC codes with arbitrarily constructed irregular SC codes. All codes have the same length, rate, and degree distribution. First, we describe the code parameters. SC Code 4.1, SC Code 4.2, and SC Code 4.3 are binary SC codes with parameters $\gamma = 4$, $\kappa = z = 13$, $m = 1$, $L = 10$, length 1,690 bits, and design rate 0.662. All these SC codes are irregular with $\Lambda = [0\ 0\ 8/13\ 5/13]$ and $\Phi \approx [0\ \ldots\ 0\ 1\ 0\ 0]$.

SC Code 4.1 is constructed by the previous method of partitioning by cutting vectors and AB circulant powers (CV-AB). SC Code 4.2 is constructed by the OO partitioning and AB circulant powers (OO-AB). SC Code 4.3 is constructed by the OO partitioning and circulant powers obtained by the CPO program (OO-CPO). The optimal partitioning is obtained by using the new systematic framework that we presented in this chapter. Moreover, in order to reduce the computational complexity, the balanced partitioning choices of circulants of $\mathbf{H}$, between $\mathbf{H}_0$ and $\mathbf{H}_1$, are considered in (4.8), and the optimal ones are chosen to construct SC Code 4.2 and SC Code 4.3. The balanced partitioning also gives more freedom to the CPO program to adjust the circulant powers, and thus it results in a better performance.

We need to generalize the definition of the partitioning matrix, introduced in Definition 9, to be able to use it for irregular SC codes as well. The partitioning matrix $\mathbf{PM} = [g_{i,j}]$ of size $\kappa \times \gamma$ is defined as follows: A circulant with row group index $i$ and column group index

$j$ in $\mathbf{H}$ is assigned to $\mathbf{H}_{\mathrm{d}}$, resp., $\mathbf{H}_0$, and $\mathbf{H}_1$, if $g_{i,j}$ is $X$, resp., 0, and 1. The circulant power matrix $\mathbf{CM} = [f_{i,j}]$, with dimension $\gamma \times \kappa$, stores the powers of non-zero circulants as before. The partitioning matrix for SC Code 4.1 is:

$$\begin{bmatrix} X & 0 & 0 & 1 & X & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & X & 0 & 0 & 0 & X & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & X & 0 & 0 & 0 & X & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & X & 0 & 0 & 0 & X & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

The partitioning matrix for SC Code 4.2 and SC Code 4.3 is:

$$\begin{bmatrix} X & 1 & 1 & 1 & X & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & X & 1 & 0 & 1 & X & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & X & 1 & 0 & 1 & X & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & X & 0 & 0 & 0 & X & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

and the circulant power matrix for SC Code 4.3 is given below:

$$\begin{bmatrix} X & 0 & 0 & 0 & X & 0 & 0 & 0 & 0 & 5 & 12 & 0 & 0 \\ 0 & X & 1 & 4 & 4 & X & 6 & 7 & 8 & 3 & 3 & 8 & 12 \\ 0 & 4 & X & 6 & 8 & 4 & X & 1 & 3 & 5 & 7 & 9 & 10 \\ 7 & 0 & 4 & X & 12 & 10 & 5 & X & 3 & 1 & 4 & 11 & 8 \end{bmatrix}.$$

In our simulations, we consider AWGN channels, and we use block min-sum algorithm with 15 iterations for the decoding. Figure 4.2 shows the FER performance for SC Codes 4.1, 4.2, and 4.3. As we see, SC Code 4.3 shows 1.3 and 0.7 orders of magnitude performance improvement compared to SC Code 4.1 and SC Code 4.2 at SNR = 5.8 dB, respectively. In terms of the number of cycles, SC Code 4.1, resp., SC Code 4.2, and SC Code 4.3, has 9,754, resp., 4,397, and 4,397, cycles-6 in its protograph, and 12,896, resp., 5,278, and 1,469, cycles-6 in its final graph.

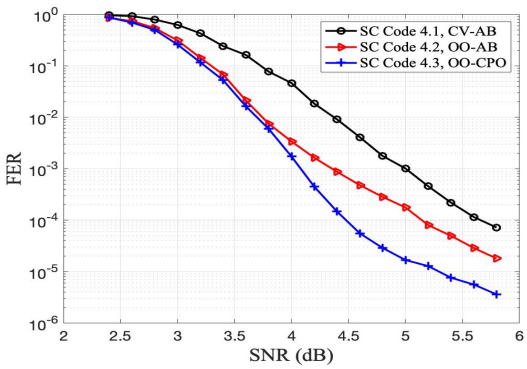



Figure 4.2: FER curves over AWGN channel for irregular SC codes of the same length, rate, and degree distribution.

## 4.5 Conclusion

In this chapter, we presented a novel combinatorial construction of finite-length irregular SC codes. Our code optimization is based on careful organization of circulants in the underlying component matrices. We showed on a representative example that our proposed codes offer performance improvement over comparable irregular SC codes. Results from this work contribute to the growing portfolio of construction methods for finite-length SC codes.

## Acknowledgment

The majority of the material in this chapter is published in [16]. The author would like to thank the collaborators in this publication.

# CHAPTER 5

# Multi-Dimensional Spatially-Coupled Code Construction

## 5.1 Introduction

One-dimensional SC (1D-SC) codes are constructed by coupling a series of disjoint block LDPC codes into a single coupled chain [3]. The 1D-SC codes have been well studied from the asymptotic perspective and the finite-length perspective. From the asymptotic perspective, density evolution techniques have been used to study the decoding threshold, e.g., [5, 6]. From the finite-length perspective, via the evaluation and optimization of the number of problematic combinatorial objects, it has been shown how an informed coupling strategy can notably improve the performance, e.g., see Chapter 2 and [10, 47].

Multi-dimensional SC (MD-SC) codes can be constructed by coupling several 1D-SC codes together via rewiring their existing connections or by adding extra VNs or CNs [48, 49]. MD-SC codes are more robust against burst erasures and channel non-uniformity, and they have improved iterative decoding thresholds, compared to 1D-SC codes [48, 49]. MD-SC codes were introduced in [48, 49] and investigated more in [50–55].

In [48, 50, 51], construction methods are presented for MD-SC codes that have specific structures, e.g., loops and triangles. The construction method for MD-SC codes presented in [49] involves connecting edges uniformly at random such that some criteria on the number of connections are satisfied. In [52], a framework is presented for constructing MD-SC codes by randomly and sparsely introducing additional CNs to connect VNs at the same positions of different chains. In [53], multiple SC codes are connected by random edge exchanges

between adjacent chains to improve the iterative decoding threshold. In [54, 55], MD-SC codes are presented to improve the error-correction performance against the severe burst errors in wireless channels.

Previous works on MD-SC codes, while promising, have some limitations. In particular, they either consider random constructions or are limited to specific topologies. As a result, they do not effectively use the added degree of freedom achieved by the multi-dimensional (MD) coupling in order to improve particular properties of the code, e.g., girth and minimum distance. They also use the density evolution technique for the performance analysis. This technique is dedicated to the asymptotic regime and is based on some assumptions, e.g., being cycle-free, that cannot be readily translated to the practical finite-length case.

Short cycles have a negative impact on the error-correction performance of graph-based codes under iterative decoding. They affect the independence of the extrinsic information exchanged in the iterative decoder. Moreover, problematic combinatorial objects that cause the error floor phenomenon, e.g., ASs, BASs, GASTs, are formed of cycles with relatively short lengths as we discussed in Chapters 2 and 3. Finally, short cycles can have a negative impact on the code minimum distance. In [56, 57], some upper bounds on the minimum distance of circulant-based block and SC codes are derived, and it is shown that the smaller the girth of the graph, the smaller the minimum distance upper bound will be. Thus, improving the girth can result in a larger minimum distance.

In this chapter, we present a comprehensive systematic framework for constructing MD-SC codes by coupling individual SC codes together to attain fewer short cycles. For connecting the constituent SC codes, we do not add extra VNs or CNs, and we only rewire some existing connections. For exchanging the connections, we follow three rules:

1. The connections that are involved in the highest number of short cycles are targeted for rewiring.

2. The neighboring constituent SC codes to which the targeted connections are rewired are chosen such that the associated short cycles convert to cycles of the largest possible length in the MD setting

3. The targeted connections are rewired to the same positions in the other constituent SC codes in order to preserve the low-latency decoding property.

From an algebraic viewpoint, problematic circulants (which correspond to groups of connections) that contribute to the highest number of short cycles in the constituent SC codes are relocated to connect these codes together. Thus, We present a systematic framework to construct MD-SC codes, which is based on an informed relocation of circulants. MD-SC codes constructed using our proposed framework enjoy a notably lower population of short cycles, and consequently better performance, compared to 1D-SC codes.

In this chapter, the operator $\overset{p}{=}$ (resp., $\overset{p}{\neq}$) defines the congruence (resp., incongruence) modulo $p$, and the operator $(.)_p$ defines modulo $p$ of an integer.

The rest of the chapter is organized as follows. In Section 5.2, the structure of our MD-SC codes is presented. In Section 5.3, our novel framework for constructing MD-SC codes with enhanced cycle properties is introduced. In Section 5.4, our simulation results are given. Finally, the conclusion appears in Section 5.5.

## 5.2 Multi-dimensional Spatially-Coupled Code Structure

In this section, we demonstrate the structure of our MD-SC codes. Our MD-SC codes have two main parameters: *MD coupling depth* $d$ and *MD coupling length* $L_2$. The parameter $L_2$ of an MD-SC code shows the number of SC codes that are connected together to form the MD-SC code. Each constituent SC code is connected to at most $(d-1)$ following SC codes, sequenced in a cyclic order. Thus, $1 \leq d \leq L_2$, and $d = 1$ corresponds to $L_2$ disjoint 1D-SC codes.

In Chapter 2, a systematic framework for partitioning the underlying block code and optimizing the circulant powers, known as the optimal overlap partitioning and circulant power optimizer (OO-CPO) technique, was introduced for constructing high performance SC codes. In this chapter, we use the OO-CPO technique for designing the constituent SC codes that are then used to construct MD-SC codes. We note that choosing high performance

1D-SC codes as constituent SC codes is not necessary in our MD-SC construction, and it only results in a better start point in a framework that further improves the performance via MD coupling.

We intend to reduce the population of cycles with length $k$, i.e., cycles-$k$, in our MD-SC code construction, and the parameter $k$ is an input to our scheme. A wise choice for $k$ is the girth [58], or the length of the cycle that is the common denominator of several problematic combinatorial objects for a specific channel, e.g., AWGN channels, MR channels, or Flash channels, see Chapters 2 and 3. For instance, a cycle-6 is the common denominator of problematic combinatorial objects for AWGN channels, and a cycle-8 (with no interconnections) is the common denominator of problematic combinatorial objects for MR channels even if the girth is 6.

An *auxiliary matrix* $\mathbf{A}_t$, $t \in \{1, \cdots, L_2 - 1\}$, has the same size as the parity-check matrix of the constituent 1D-SC code , i.e., $\mathbf{H}_{\text{SC}}$, and appears in the parity-check matrix of the final MD-SC code, see (5.2). The auxiliary matrices are all-zero matrices at the beginning of the framework and are filled with non-zero circulants during the construction process. A *relocation* is defined as moving a non-zero circulant of $\mathbf{H}_{\text{SC}}$ to the same position in one of the auxiliary matrices.

Consider an SC code with parity-check matrix $\mathbf{H}_{\text{SC}}$, memory $m$, and coupling length $L$ as the constituent 1D-SC code, and let $\mathbf{R}_\nu$ be the middle replica of $\mathbf{H}_{\text{SC}}$, i.e., $\nu = \lceil L/2 \rceil$. There are $\kappa\gamma$ non-zero circulants in this replica. Out of these $\kappa\gamma$ circulants, we choose $\mathcal{T}$ circulants that are the most problematic, i.e., that contribute to the highest number of cycles-$k$. The parameter $\mathcal{T}$ is called the *MD coupling density*. We relocate the chosen circulants to auxiliary matrices $\mathbf{A}_1$, $\mathbf{A}_2$, $\ldots$, $\mathbf{A}_{d-1}$ such that a relocated circulant from $\mathbf{H}_{\text{SC}}$ is moved to the same position in one of the auxiliary matrices. The same relocations are repeated for all the $(L-1)$ remaining replicas. As a result,

$$\mathbf{H}_{\text{SC}} = \mathbf{H}'_{\text{SC}} + \sum_{t=1}^{d-1} \mathbf{A}_t, \tag{5.1}$$

where $\mathbf{H}'_{\text{SC}}$ is derived from $\mathbf{H}_{\text{SC}}$ by removing the $\mathcal{T}$ chosen circulants.

We note that the middle replica $\mathbf{R}_\nu$ is considered for ranking the circulants in order to include all possible cycles-$k$ that a non-zero circulant in $\mathbf{H}_{\text{SC}}$ can contribute to. The parity-check matrix $\mathbf{H}_{\text{SC}}^{\text{MD}}$ of the MD-SC code is constructed as follows, where $\mathbf{A}_d = \mathbf{A}_{d+1} = \cdots = \mathbf{A}_{L_2-1} = \mathbf{0}$: (The non-zero auxiliary matrices are $\mathbf{A}_1$, $\mathbf{A}_2$, $\ldots$, $\mathbf{A}_{d-1}$.)

$$\mathbf{H}_{\text{SC}}^{\text{MD}} = \begin{bmatrix} \mathbf{H}'_{\text{SC}} & \mathbf{A}_{L_2-1} & \cdots & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{H}'_{\text{SC}} & \cdots & \mathbf{A}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{L_2-1} & \mathbf{A}_{L_2-2} & \cdots & \mathbf{H}'_{\text{SC}} \end{bmatrix}. \tag{5.2}$$

$\mathbf{H}_{\text{SC}}^{\text{MD}}$ can be viewed as a collection of $L_2$ rows and $L_2$ columns of *segments* $\mathbf{S}_{a,b}$, where $0 \leq a \leq L_2 - 1$ and $0 \leq b \leq L_2 - 1$. Each segment $\mathbf{S}_{a,b}$ is a matrix with the same dimension as $\mathbf{H}_{\text{SC}}$, $\mathbf{S}_{a,a} = \mathbf{H}'_{\text{SC}}$, $\mathbf{S}_{(a+t)_{L_2},a} = \mathbf{A}_t$ for $t \in \{1, \cdots, d-1\}$, and $\mathbf{S}_{(a+t)_{L_2},a} = \mathbf{0}$ for $t \in \{d, \cdots, L_2 - 1\}$. In Example 10, we show the graphical illustration of an MD-SC code having the presented structure.

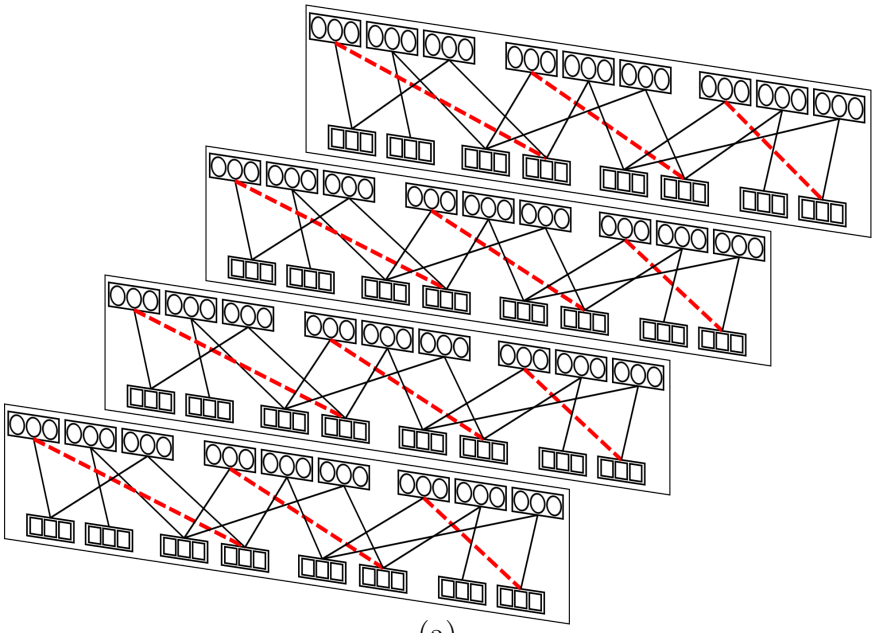
**Example 10.** *Consider an SC code with $\gamma = 2$, $\kappa = 3$, $z = 3$, $m = 1$, and $L = 3$. The matrix $\mathbf{H}$ of the underlying block code and the component matrices are given below:*

$$\mathbf{H} = \begin{bmatrix} \sigma^{f_{0,0}} & \sigma^{f_{0,1}} & \sigma^{f_{0,2}} \\ \sigma^{f_{1,0}} & \sigma^{f_{1,1}} & \sigma^{f_{1,2}} \end{bmatrix}, \ \mathbf{H}_0 = \begin{bmatrix} \sigma^{f_{0,0}} & \mathbf{0} & \sigma^{f_{0,2}} \\ \mathbf{0} & \sigma^{f_{1,1}} & \mathbf{0} \end{bmatrix}, \ \mathbf{H}_1 = \begin{bmatrix} \mathbf{0} & \sigma^{f_{0,1}} & \mathbf{0} \\ \sigma^{f_{1,0}} & \mathbf{0} & \sigma^{f_{1,2}} \end{bmatrix}.$$

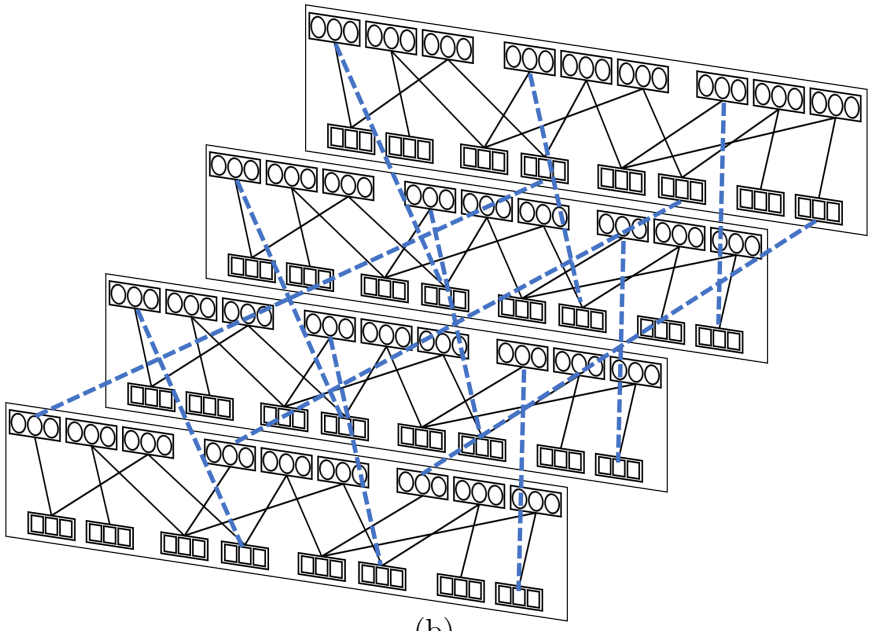
*We intend to construct an MD-SC code with parameters $\mathcal{T} = 1$, $d = 2$, and $L_2 = 4$. Assume $\sigma^{f_{1,0}}$ is the most problematic circulant, and we relocate it to $\mathbf{A}_1$. This relocation is applied to all $L = 3$ instances of the problematic circulant. We remind that each circulant corresponds to a group of $z$ connections in the graph of the SC code. The four constituent SC codes along with their problematic connections are depicted in Figure 5.1 (a). The problematic connections are rewired to the same positions in the next SC codes, in a cyclic order, to construct the MD-SC code, Figure 5.1 (b).*

**Definition 11.** *We introduce some necessary definitions:*

1. *Let $\mathcal{C}_{i,j}$, where $0 \leq i \leq (L+m)\gamma - 1$ and $0 \leq j \leq L\kappa - 1$, be a non-zero circulant in*



(a)



(b)

Figure 5.1: (a) Four 1D-SC codes. Each line represents a group of connections (defined by a circulant) from $z$ VNs to $z$ CNs. Problematic connections are shown in dashed red lines. (b) MD-SC code with $\mathcal{T} = 1$, $d = 2$, and $L_2 = 4$. Rewired connections are shown in dashed blue lines.

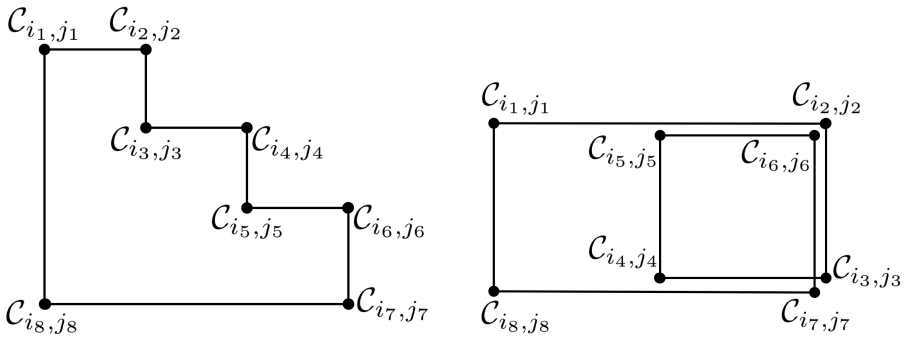



Figure 5.2: Cycles-8 with $C_{\mathcal{O}_8} = \{\mathcal{C}_{i_1,j_1}, \ldots, \mathcal{C}_{i_8,j_8}\}$. Each line represents a connection between two circulants. (a) All circulants are unique. (b) $\mathcal{C}_{i_6,j_6} = \mathcal{C}_{i_2,j_2}$ and $\mathcal{C}_{i_7,j_7} = \mathcal{C}_{i_3,j_3}$.

$\mathbf{H}_{\text{SC}}$. *We say* $\mathcal{C}_{i,j}$ *is relocated to* $\mathbf{A}_t$*, where* $t \in \{1, \cdots, d-1\}$*, if it is moved from* $\mathbf{H}_{\text{SC}}$ *to* $\mathbf{A}_t$*. We denote this relocation as* $\mathcal{C}_{i,j} \rightarrow \mathbf{A}_t$*.*

2. $\mathcal{C}_{i,j}@\mathbf{S}_{a,b}$ *refers to the circulant* $\mathcal{C}_{i,j}$ *in segment* $\mathbf{S}_{a,b}$*. When* $\mathcal{C}_{i,j} \rightarrow \mathbf{A}_t$*, the value of* $\mathcal{C}_{i,j}@\mathbf{S}_{a,a}$ *is copied to* $\mathcal{C}_{i,j}@\mathbf{S}_{(a+t)_{L_2},a}$*, and* $\mathcal{C}_{i,j}@\mathbf{S}_{a,a}$ *becomes zero* ($a \in \{0, \cdots, L_2 - 1\}$ *and* $t \in \{1, \cdots, d-1\}$)*.*

3. *The MD mapping* $M : \{\mathcal{C}_{i,j}\} \rightarrow \{0, \cdots, d-1\}$ *is a mapping from a non-zero circulant in* $\mathbf{H}_{\text{SC}}$ *to an integer in* $\{0, \cdots, d-1\}$*, and it is defined as follows:*

   (a) *If* $\mathcal{C}_{i,j} \rightarrow \mathbf{A}_t$*,* $M(\mathcal{C}_{i,j}) = t$*.*

   (b) *If* $\mathcal{C}_{i,j}$ *is kept in* $\mathbf{H}'_{\text{SC}}$ *(no relocation),* $M(\mathcal{C}_{i,j}) = 0$*.*

4. *A cycle-k, or* $\mathcal{O}_k$*, visits k circulants in the parity-check matrix of the code. We list the k circulants of* $\mathcal{O}_k$*, according to the order they are visited when the cycle is traversed in a clockwise direction, in a sequence as* $C_{\mathcal{O}_k} = \{\mathcal{C}_{i_1,j_1}, \mathcal{C}_{i_2,j_2}, \ldots, \mathcal{C}_{i_k,j_k}\}$*, where* $i_1 = i_2, j_2 = j_3, \ldots, i_{k-1} = i_k, j_k = j_1$*. A circulant can be visited more than once, e.g., Figure 5.2.*

5. *We denote the distance between two circulants* $\mathcal{C}_{i_u,j_u}$ *and* $\mathcal{C}_{i_v,j_v}$ *on a cycle* $\mathcal{O}_k$*, where* $u, v \in \{1, \ldots, k\}$*, as* $D_{\mathcal{O}_k}(\mathcal{C}_{i_u,j_u}, \mathcal{C}_{i_v,j_v}) \in \{0, \ldots, k-1\}$*. By definition,* $D_{\mathcal{O}_k}(\mathcal{C}_{i_u,j_u}, \mathcal{C}_{i_v,j_v}) = |v - u|$*. For example,* $D_{\mathcal{O}_8}(\mathcal{C}_{i_1,j_1}, \mathcal{C}_{i_4,j_4}) = 3$ *in Figure 5.2 (a).*

Because of the structure of MD-SC codes, when a non-zero circulant in one replica of

$\mathbf{H}_{\text{SC}}$ is relocated, the same relocation is applied to the $(L-1)$ other replicas as well.

$$M(\mathcal{C}_{i,j}) = M(\mathcal{C}_{i-\rho\gamma,j-\rho\kappa}), \text{ where } \rho = \lfloor j/\kappa \rfloor. \tag{5.3}$$

In the new MD-SC code design framework, we effectively answer two questions: which circulants to relocate, and where to relocate them. We note that the relocations of circulants to the same positions in the auxiliary matrices preserve the special structure of SC codes, which makes them suitable for applications that require low decoding latency.

## 5.3 Framework for MD-SC Code Design

In this section, we present a new framework for constructing MD-SC codes. First, we investigate the effects of relocating a subset of circulants on the population of cycles. Then, we present our algorithm for constructing MD-SC codes which is based on a score voting policy.

### 5.3.1 The Effects of Relocation of Circulants on Cycles

Consider a cycle $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}$ with sequence of circulants $C_{\mathcal{O}_k}$. Prior to any relocation, there are $L_2$ instances of this cycle in the MD-SC code with parameter $L_2$, one per each constituent SC code. We investigate the effect of relocating a subset of circulants of $\mathcal{O}_k$, and we call this subset *targeted circulants*. We show that, after relocations, $L_2$ instances of circulants of $C_{\mathcal{O}_k}$ can form $L_2$ cycles of length $k$, $L_2/2$ cycles of length $2k$, ..., or one cycle of length $L_2 k$. The first case is a result of bad choices for relocations, and the rest are more preferable. In fact, we opt for the relocations that result in larger cycles (with smaller cardinality as a result).

**Theorem 5.** *Let $C_{\mathcal{O}_k} = \{\mathcal{C}_{i_1,j_1}, \mathcal{C}_{i_2,j_2}, \ldots, \mathcal{C}_{i_k,j_k}\}$ be the sequence of circulants in $\mathbf{H}_{\text{SC}}$ that are visited in a clockwise order by $\mathcal{O}_k$. If the following equation holds, the $L_2$ instances of circulants of $C_{\mathcal{O}_k}$ form $L_2$ cycles-$k$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$,*

$$\sum_{u=1}^{k} (-1)^u M(\mathcal{C}_{i_u,j_u}) \overset{L_2}{=} 0. \tag{5.4}$$

*Otherwise, the instances of the targeted circulants do not result in cycles-k in* $\mathbf{H}_{\text{SC}}^{\text{MD}}$ [1]. *We call (5.4) the Ineffective Relocation Condition, or IRC, in the rest of this chapter.*

*Proof.* Let $(\mathcal{C}_{i_u,j_u}, \mathcal{C}_{i_{u+1},j_{u+1}})$ be a pair of consecutive circulants in $C_{\mathcal{O}_k}$, where $u \in \{1, \ldots, k\}$ and $\mathcal{C}_{i_{k+1},j_{k+1}} = \mathcal{C}_{i_1,j_1}$. By definition, two circulants have the same row (resp., column) group index, i.e., $i_u = i_{u+1}$ (resp., $j_u = j_{u+1}$), when $u \overset{2}{=} 1$ (resp., $u \overset{2}{=} 0$). Before relocations, $\mathcal{C}_{i_u,j_u}@\mathbf{S}_{a,a} \neq \mathbf{0}$ and $\mathcal{C}_{i_u,j_u}@\mathbf{S}_{a,b} = \mathbf{0}$, where $\mathcal{C}_{i_u,j_u} \in C_{\mathcal{O}_k}$, $a, b \in \{0, \cdots, L_2 - 1\}$, and $a \neq b$. This results in $L_2$ instances of $\mathcal{O}_k$, one per each segment $\mathbf{S}_{a,a}$. After relocations, the circulants in $C_{\mathcal{O}_k}$ do not all belong to the same segment. We remind that the matrix $\mathbf{H}_{\text{SC}}^{\text{MD}}$ is formed of $(L_2)^2$ segments, see (5.2).

Here, a unit of a MD horizontal (resp., MD vertical) shift is defined as cyclically going one segment right (resp., down) when we go from $\mathcal{C}_{i_u,j_u}$ to $\mathcal{C}_{i_{u+1},j_{u+1}}$. The cycle $\mathcal{O}_k$ reflects in the MD-SC code as cycles with the same length $k$ if and only if when we start from $\mathcal{C}_{i_1,j_1} \neq \mathbf{0}$ from one segment and traverse the circulants of the cycle in a clock wise order (with the same order they appear in $C_{\mathcal{O}_k}$), we end up at the same segment that we started with.

The segments of $\mathbf{H}_{\text{SC}}^{\text{MD}}$ appear in the cyclic order $\{\mathbf{H}'_{\text{SC}}, \mathbf{A}_{L_2-1}, \cdots, \mathbf{A}_1\}$, with the MD mapping $\{0, L_2 - 1, \cdots, 1\}$, from left to right. These segments appear in the cyclic order $\{\mathbf{H}'_{\text{SC}}, \mathbf{A}_1, \cdots, \mathbf{A}_{L_2-1}\}$, with the MD mapping $\{0, 1, \cdots, L_2 - 1\}$, from top to bottom, see (5.2). Thus, the MD horizontal shift, when we go from $\mathcal{C}_{i_u,j_u}$ to $\mathcal{C}_{i_{u+1},j_{u+1}}$, $u \in \{1, 3, \ldots, k-1\}$, is $(M(\mathcal{C}_{i_u,j_u}) - M(\mathcal{C}_{i_{u+1},j_{u+1}}))_{L_2}$ units. Similarly, the MD vertical shift, when we go from $\mathcal{C}_{i_u,j_u}$ to $\mathcal{C}_{i_{u+1},j_{u+1}}$, $u \in \{0, 2, \ldots, k\}$, is $(M(\mathcal{C}_{i_{u+1},j_{u+1}}) - M(\mathcal{C}_{i_u,j_u}))_{L_2}$ units. We remind that the operator $(.)_p$ defines modulo $p$ of an integer. The total MD horizontal and vertical shifts when we traverse the circulants of $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$ are $\delta_H$ and $\delta_V$, respectively:

$$\begin{aligned}
\delta_H &= \Big(\sum_{u \in \{1,3\ldots,k-1\}} [M(\mathcal{C}_{i_u,j_u}) - M(\mathcal{C}_{i_{u+1},j_{u+1}})]\Big)_{L_2} = \Big(-\sum_{u=1}^{k} [(-1)^u M(\mathcal{C}_{i_u,j_u})]\Big)_{L_2}, \\
\delta_V &= \Big(\sum_{u \in \{2,4\ldots,k\}} [M(\mathcal{C}_{i_{u+1},j_{u+1}}) - M(\mathcal{C}_{i_u,j_u})]\Big)_{L_2} = \Big(-\sum_{u=1}^{k} [(-1)^u M(\mathcal{C}_{i_u,j_u})]\Big)_{L_2}.
\end{aligned} \tag{5.5}$$

[1] Equation (5.4) resembles Fossorier's condition on circulant powers of a CB code that makes a cycle in the protograph result in multiple cycles in the lifted graph of the code [29].

The relocations are ineffective if and only if the start and end segments are the same when we traverse the $k$ circulants of $\mathcal{O}_k$. For this to happen, the total MD horizontal and vertical shifts ($\delta_H$ and $\delta_V$) need to be zero, which results in (5.4). □

If equation (5.4), or IRC, holds for the circulants of $\mathcal{O}_k$, $L_2$ instances of circulants of $C_{\mathcal{O}_k}$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$ form $L_2$ cycles-$k$ in the MD-SC code (unpreferable). Theorem 6 investigates the situation when IRC does not necessarily hold.

**Theorem 6.** *Each cycle $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}$ results in $\tau$ cycles with length $L_2k/\tau$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$, where*

$$\tau = \gcd(L_2, \Delta_{\mathcal{O}_k}), \text{ and} \qquad \Delta_{\mathcal{O}_k} = (-\sum_{u=1}^{k}[(-1)^u M(\mathcal{C}_{i_u,j_u})])_{L_2}. \tag{5.6}$$

*The operator* gcd *outputs the greatest common divisor of its two operands.*

*Proof.* Consider a cycle $\mathcal{O}_k$ with $C_{\mathcal{O}_k} = \{\mathcal{C}_{i_1,j_1}, \ldots, \mathcal{C}_{i_k,j_k}\}$ in $\mathbf{H}_{\text{SC}}$. There are $(L_2)^2$ instances of $\mathcal{C}_{i_u,j_u}$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$, $u \in \{1, \ldots, k\}$, one per each segment, and only $L_2$ of them can be non-zero. We traverse the circulants of $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$ according to the order they appear in $C_{\mathcal{O}_k}$ starting from a non-zero instance of $\mathcal{C}_{i_1,j_1}$. After traversing all $k$ circulants, we reach circulant $\mathcal{C}_{i_1,j_1}$ in a segment that is (cyclically) $\Delta_{\mathcal{O}_k}$ units right and $\Delta_{\mathcal{O}_k}$ units down from the segment we started.

If $\Delta_{\mathcal{O}_k} = 0$, the cycle is complete after traversing the $k$ circulants. In this case, there are $L_2$ instances of $C_{\mathcal{O}_k}$, one per each non-zero instance of $\mathcal{C}_{i_1,j_1}$. If $\Delta_{\mathcal{O}_k} \neq 0$, the cycle cannot be complete after traversing $k$ circulants. We proceed traversing the circulants until we reach $\mathcal{C}_{i_1,j_1}$ that is in the same segment that we started from.

We define the parameter $\lambda$ as follows:

$$\lambda = \min\{g | g \in \{1, 2, \cdots\}, g\Delta_{\mathcal{O}_k} \overset{L_2}{=} 0\}. \tag{5.7}$$

Then, we complete the cycle after traversing $\lambda k$ circulants. The parameter $\lambda$ is the minimum integer value such that $\lambda\Delta_{\mathcal{O}_k} \overset{L_2}{=} 0$, i.e., $\lambda = L_2/\gcd(L_2, \Delta_{\mathcal{O}_k})$. The $L_2$ non-zero instances of the $k$ circulants in $C_{\mathcal{O}_k}$ form $\tau = L_2k/\lambda k = \gcd(L_2, \Delta_{\mathcal{O}_k})$ cycles of the length $\lambda k = L_2k/\tau$. □

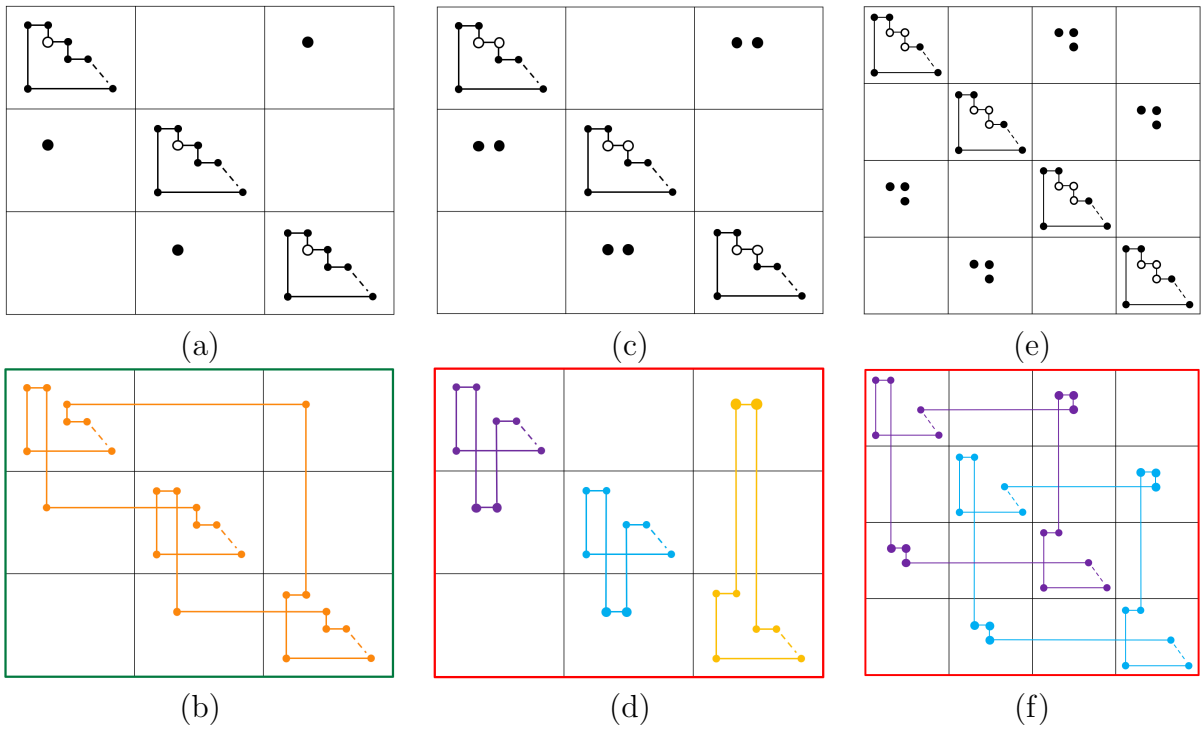



Figure 5.3: (a) $\mathcal{C}_{i_a,j_a} \rightarrow \mathbf{A}_1$. The white circles show original locations of the relocated circulant. (b) A cycle-$3k$ is formed. (c) $\{\mathcal{C}_{i_a,j_a}, \mathcal{C}_{i_b,j_b}\} \rightarrow \mathbf{A}_1$. (d) Three cycles-$k$ are formed. (e) $\{\mathcal{C}_{i_a,j_a}, \mathcal{C}_{i_b,j_b}, \mathcal{C}_{i_c,j_c}\} \rightarrow \mathbf{A}_2$. (f) Two cycles-$2k$ are formed.

For example, when $L_2$ and $\Delta_{\mathcal{O}_k}$ are relatively prime, there is a cycle with length $L_2 k$ that traverses all non-zero instances of the circulants of $C_{\mathcal{O}_k}$. When $\tau = \gcd(L_2, \Delta_{\mathcal{O}_k}) = L_2$, the non-zero instances of the circulants of $C_{\mathcal{O}_k}$ form $L_2$ cycles with length $k$. In our algorithm for the MD-SC code construction, the relocations that result in smaller $\tau$ are more preferred as they result in larger cycles.

**Remark 10.** *Review some properties of* gcd *that are used in the rest of this chapter:*

- $\gcd(a, 0) = |a|$ *for any non-zero* $a$,
- $\gcd(a + yb, b) = \gcd(a, b)$ *for any integer* $y$,
- $\gcd(-a, b) = \gcd(a, b)$.

**Example 11.** *Let* $C_{\mathcal{O}_k} = \{\mathcal{C}_{i_1,j_1}, \ldots, \mathcal{C}_{i_k,j_k}\}$ *be the sequence of circulants of* $\mathcal{O}_k$*, and* $n$ *be the number of its relocated circulants.*

1. *Let* $n = 1$, $\mathcal{C}_{i_a,j_a} \rightarrow \mathbf{A}_1$, *and* $L_2 = 3$. *Then,* $\Delta_{\mathcal{O}_k} = ((-1)^a)_3$ *and* $\tau = 1$. *Figure 5.3 (a)*

*shows* $\mathcal{C}_{i_a,j_a} \to \mathbf{A}_1$. *Figure 5.3 (b) shows that a cycle-*$3k$ *(shown in orange) is formed. The green border represents that this relocation is preferable.*

2. *Let* $n = 2$, $\mathcal{C}_{i_a,j_a}, \mathcal{C}_{i_b,j_b} \to \mathbf{A}_1$, *and* $L_2 = 3$. *Suppose* $D_{\mathcal{O}_k}(\mathcal{C}_{i_a,j_a}, \mathcal{C}_{i_b,j_b}) = 1$. *Then,* $\Delta_{\mathcal{O}_k} = ((-1)^a - (-1)^a)_3 = 0$ *and* $\tau = L_2 = 3$. *Figure 5.3 (c) shows* $\mathcal{C}_{i_a,j_a}, \mathcal{C}_{i_b,j_b} \to \mathbf{A}_1$. *Figure 5.3 (d) shows that three cycles-*$k$ *are formed. The red border represents that these relocations are not preferable.*

3. *Let* $n = 3$, *and* $\mathcal{C}_{i_a,j_a}$, $\mathcal{C}_{i_b,j_b}$, $\mathcal{C}_{i_c,j_c} \to \mathbf{A}_2$, *and* $L_2 = 4$. *Suppose these three circulants are consecutive in* $C_{\mathcal{O}_k}$. *Then,* $\Delta_{\mathcal{O}_k} = ((-1)^a(2 - 2 + 2))_4 = 2$ *and* $\tau = 2$. *Figure 5.3 (e) shows* $\mathcal{C}_{i_a,j_a}, \mathcal{C}_{i_b,j_b}, \mathcal{C}_{i_c,j_c} \to \mathbf{A}_2$. *Figure 5.3 (f) shows that two cycles-*$2k$ *are formed. The red border represents that these relocations are less preferred. We note that if we relocated the targeted circulants to* $\mathbf{A}_1$ *instead, the result would be one cycle-*$4k$ *which is more preferred.*

4. *If all* $n$ *targeted circulants are relocated to the same auxiliary matrix* $\mathbf{A}_t$, *where* $t \in \{1, \cdots, d-1\}$, *and each pair of consecutive relocated circulants have even distance on* $\mathcal{O}_k$, *see Definition 11, the relocations are ineffective when* $nt \overset{L_2}{=} 0$. *Let* $\mathcal{C}_{i_v,j_v}$ *be the first relocated circulant in* $C_{\mathcal{O}_k}$. *Then,*

$$\begin{aligned}\sum_{u=1}^{k}(-1)^u M(\mathcal{C}_{i_u,j_u}) &= \sum_{M(\mathcal{C}_{i_u,j_u})>0}(-1)^u M(\mathcal{C}_{i_u,j_u}) = M(\mathcal{C}_{i_v,j_v}) \sum_{M(\mathcal{C}_{i_u,j_u})>0}(-1)^u \\ &= M(\mathcal{C}_{i_v,j_v})(-1)^v(1 + \cdots + 1) = nt(-1)^v,\end{aligned} \tag{5.8}$$

*which* $\overset{L_2}{=} 0$ *only if* $nt \overset{L_2}{=} 0$.

5. *If all* $n$ *targeted circulants are relocated to the same auxiliary matrix* $\mathbf{A}_t$, *where* $t \in \{1, \cdots, d-1\}$, *and each pair of consecutive relocated circulants have odd distance on* $\mathcal{O}_k$, *see Definition 11, which can only happen if* $n \overset{2}{=} 0$ *since* $k \overset{2}{=} 0$, *the relocations are always ineffective. Let* $\mathcal{C}_{i_v,j_v}$ *be the first relocated circulant in* $C_{\mathcal{O}_k}$. *Then,*

$$\sum_{u=1}^{k}(-1)^u M(\mathcal{C}_{i_u,j_u}) = M(\mathcal{C}_{i_v,j_v}) \sum_{M(\mathcal{C}_{i_u,j_u})>0}(-1)^u = t(-1)^v(1 - 1 + \cdots - 1), \tag{5.9}$$

*which* $\stackrel{L_2}{=} 0$ *always since* $n \stackrel{2}{=} 0$.

**Remark 11.** *A circulant can appear more than once in* $C_{\mathcal{O}_k}$*, e.g., see Figure 5.2 (b). A circulant that is repeated* $r$ *times in the sequence can be interpreted in our analysis as* $r$ *different circulants; every two circulants from this group have an even distance on* $\mathcal{O}_k$*. The relocation of a circulant that appears* $r$ *times is equivalent to the relocation of* $r$ *circulants with the above property to the same auxiliary matrix.*

### 5.3.2 Score Voting Algorithm for MD-SC Code Design

Our framework is based on a score voting policy and aims at minimizing the population of short cycles. As stated in Section 5.2, the MD coupling with depth $d$ is performed via relocating problematic circulants to auxiliary matrices $\mathbf{A}_t$, $t \in \{1, \cdots, d-1\}$. After relocating one circulant, the ranking of the problematic circulants (with respect to the number of cycles each of them is visited by) changes. Thus, the relocations are performed sequentially.

In our framework, we use a tree-based strategy for constructing MD-SC codes, by identifying a proper sequence of relocations such that as many as possible designated cycles are removed in the constituent SC codes, while as few as possible short cycles are formed in the multi-dimensional configuration. To assign scores to the branches of the tree, we use the results of Section 5.3.1.

Consider a targeted circulant $\mathcal{C}_{i_v,j_v}$. There are $d$ possible relocation options for this circulant: relocate to one of the $(d-1)$ auxiliary matrices or keep in $\mathbf{H}'_{\text{SC}}$, i.e., $M(\mathcal{C}_{i_v,j_v}) = t$ and $t \in \{0, 1, \cdots d-1\}$. Each cycle $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}$ that has the targeted circulant in its sequence gives a score for each of these options, and the collective scoring results are considered for making a decision. The score $R(\mathcal{O}_k, t)$ is proportional to the length of the cycles that the non-zero instances of the circulants of $C_{\mathcal{O}_k}$ form after applying the corresponding option (after performing a relocation or keeping the targeted circulant in $\mathbf{H}'_{\text{SC}}$):

$$R(\mathcal{O}_k, t) = \frac{L_2}{\gcd(L_2, \Delta_{\mathcal{O}_k})}, \quad \Delta_{\mathcal{O}_k} = ((-1)^{v+1} rt - \sum_{\mathcal{C}_{i_u,j_u} \in C_{\mathcal{O}_k} \setminus \mathcal{C}_{i_v,j_v}} [(-1)^u M(\mathcal{C}_{i_u,j_u})])_{L_2}. \quad (5.10)$$

Here, we assumed $\mathcal{C}_{i_v,j_v}$ is repeated $r$ times in $C_{\mathcal{O}_k}$, and $v$ is the index of one of the repetitions.

In fact, there might be several options for a targeted circulant such that IRC (i.e., (5.4)) does not hold. However, the options that result in larger cycles (with smaller cardinality as a consequence) are preferable. We use a scoring system in our algorithm for constructing MD-SC codes in order to convert short cycles in the constituent SC codes into cycles with lengths as large as possible. Example 12 studies several scenarios, and in each one, $\mathcal{O}_k$ gives different scores regarding a targeted circulant.

**Example 12.** *Consider the cycle $\mathcal{O}_k$ and a targeted circulant $\mathcal{C}_{i_v,j_v} \in C_{\mathcal{O}_k}$.*
*Scenario 1: No circulants of $\mathcal{O}_k$ are previously relocated, and $\mathcal{C}_{i_v,j_v}$ appears once in $C_{\mathcal{O}_k}$, i.e., $r = 1$. Thus, IRC does not hold after a relocation, regardless of the auxiliary matrix that $\mathcal{C}_{i_v,j_v}$ is relocated to. For the option $M(\mathcal{C}_{i_v,j_v}) = t$, $R(\mathcal{O}_k, t) = L_2/\gcd(L_2, t)$. For instance, $\mathcal{O}_k$ gives score 1 to the option "keep in $\mathbf{H}'_{\mathrm{SC}}$", and gives score $L_2$ to the option "relocate to $\mathbf{A}_1$".*
*Scenario 2: No circulants of $\mathcal{O}_k$ are relocated before, and $\mathcal{C}_{i_v,j_v}$ appears $L_2$ times in $C_{\mathcal{O}_k}$, i.e., $r = L_2$. Then, $\mathcal{O}_k$ cannot be removed via any relocation of $\mathcal{C}_{i_v,j_v}$ (Refer to the relocation of $L_2$ circulants with even mutual distances to the same auxiliary matrix in Example 11.), and $\mathcal{O}_k$ gives the lowest score, i.e., 1, to all the $d$ options:*

$$R(\mathcal{O}_k, t) = \frac{L_2}{\gcd(L_2, 0)} = 1.$$

*Scenario 3: Circulant $\mathcal{C}_{i_w,j_w} \in C_{\mathcal{O}_k}$ is already relocated to $\mathbf{A}_1$, $D_{\mathcal{O}_k}(\mathcal{C}_{i_v,j_v}, \mathcal{C}_{i_w,j_w}) = 2$, and both circulants appear once in $C_{\mathcal{O}_k}$, i.e., $r = 1$. Then, IRC does not hold for options "no relocation" and "relocation to $\mathbf{A}_t$", when $t \neq L_2 - 1$. In fact, for the option $M(\mathcal{C}_{i_u,j_u}) = t$, $t \in \{0, \cdots, d-1\}$, $\mathcal{O}_k$ gives score $R(\mathcal{O}_k, t) = L_2/\gcd(L_2, t+1)$. For instance, $\mathcal{O}_k$ gives score 1 to "relocate to $\mathbf{A}_{L_2-1}$", and gives score $L_2$ to "keep in $\mathbf{H}'_{\mathrm{SC}}$" and "relocate to $\mathbf{A}_{t'}$" where $(t'+1)$ and $L_2$ are relatively prime.*

The relocation options are {relocate to $\mathbf{A}_1$,..., relocate to $\mathbf{A}_{d-1}$, keep in $\mathbf{H}'_{\mathrm{SC}}$}. We identify the best options for a targeted circulant as follows: Each cycle-$k$ in $\mathbf{H}_{\mathrm{SC}}$ that has the targeted circulant in its sequence gives a score to each possible relocation option in {relocate

**Algorithm 1** Score Voting Algorithm for Relocation

**inputs:** targeted circulant $\mathcal{C}_{i_v,j_v}$, $k$, $[M(\mathcal{C}_{i,j})]$, $d$, and $L_2$.
Find $\Psi$, the set of all active/inactive cycles-$k$ that have $\mathcal{C}_{i_v,j_v}$ in their sequences.
**for** each $\mathcal{O}_k \in \Psi$ **do**
    **for** $t \leftarrow 0$ to $d-1$ **do**
        $M(\mathcal{C}_{i_v,j_v}) = t$.
        $\Delta_{\mathcal{O}_k} = (-\sum_{u=1}^{k}[(-1)^u M(\mathcal{C}_{i_u,j_u})])_{L_2}$.
        $R(\mathcal{O}_k, t) = L_2/\gcd(L_2, \Delta_{\mathcal{O}_k})$.
$\Phi = \{0, \cdots, d-1\}$.
**for** $x \leftarrow 1$ to $\lfloor L_2/2 \rfloor$ **do**
    **if** $L_2 \overset{x}{=} 0$ **then**
        $\Phi \leftarrow \arg\min_{t\in\Phi} |\{\mathcal{O}_k | \mathcal{O}_k \in \Psi, R(\mathcal{O}_k, t) = x\}|$.
**output:** relocation options $\Phi$.

to $\mathbf{A}_1$,..., relocate to $\mathbf{A}_{d-1}$, keep in $\mathbf{H}'_{\text{SC}}\}$. The decision is made based on the collected scores. We first identify and keep the options that receive the least number of scores with value $x = 1$, as these options result in fewer cycle-$k$ in the MD-SC code. Among these options, we keep the ones that that receive the least number of scores with value $x = 2$, as these options result in fewer cycle-$2k$ in the MD-SC code. We continue until we reach $x = \lfloor L_2/2 \rfloor$ or there is only one option left for the targeted circulant. Then, all survived options are recorded as branches of a tree, and the next targeted circulant is chosen and similarly evaluated for each branch.

**Remark 12.** *The score value is by definition a divisor of $L_2$. Thus, $x$ is only considered for the above analysis if $L_2 \overset{x}{=} 0$. Moreover, we do not continue the procedure until reaching $x = L_2$. This is because two options that receive the same number of scores with value $x$, $x \in \{x | x \in \{1, 2, \ldots, L_2/2\}, L_2 \overset{x}{=} 0\}$, receive the same number of scores with value $x = L_2$.*

Algorithm 1 shows the procedure to find the best relocation options. We consider all cycles-$k$ in $\mathbf{H}_{\text{SC}}$ that visit circulants in the middle replica. We call the cycles for which IRC holds the active cycles and the rest as the inactive cycles. We highlight three points here: (1) The targeted circulant $\mathcal{C}_{i_v,j_v}$ is chosen from $\{\mathcal{C}_{i,j} | \mathcal{C}_{i,j} \neq \mathbf{0}$ and $M(\mathcal{C}_{i,j}) = 0\}$ to increase the MD coupling. (2) The most problematic circulant is the one that is visited by the most active cycles. (3) Each active/inactive cycle that visits $\mathcal{C}_{i_v,j_v}$ (has $\mathcal{C}_{i_v,j_v}$ in its sequence) gives a score to each relocation option, since the status of cycles-$k$ (being active or inactive) changes by

relocations.

Now, we are ready to describe our algorithm for constructing MD-SC codes. A solution for constructing an MD-SC code is a sequence of up to $\mathcal{T}$ relocations. Our algorithm for constructing MD-SC codes is greedy in the sense that, at each step, it chooses the relocation options that result in the least number of short cycles. The solutions for constructing an MD-SC code are recorded in a tree structure. The root of the tree corresponds to the initial state, where $\mathbf{H}'_{\text{SC}} = \mathbf{H}_{\text{SC}}$ and $\mathbf{A}_t = \mathbf{0}$ for $t \in \{1, \cdots, L_2 - 1\}$. Other nodes correspond to one relocation each, and the path from the root node to a node at level $l$, $l \in \{1, \ldots, \mathcal{T}\}$, describes a solution with $l$ relocations for constructing the MD-SC code. At each iteration of our algorithm, we expand the tree by one level and trim the solutions that do not result in MD-SC codes with the best cycle properties amongst the solutions at that level.

**Expanding:** At iteration $l$, we consider all nodes at level $l-1$, individually. For each node at level $l-1$, we perform the relocations described by the path from the root to the node, and form matrix $\mathbf{H}'_{\text{SC}}$ and the auxiliary matrices, accordingly. Next, all non-zero circulants in the middle replica of $\mathbf{H}'_{\text{SC}}$ are ranked, in a decreasing order, based on the number of active cycles-$k$ that they are visited by. Then, we target one circulant from the top of the list and find its best relocation options, by Algorithm 1. If the option "keep in $\mathbf{H}'_{\text{SC}}$" is among the best options, the next problematic circulant in the sorted list is targeted. We continue this process until the most problematic circulant, such that its relocation reduces the population of short cycles, is found. Then, its best relocation options are added as children of the current node. If there is no circulant in the list such that its relocation reduces the population of short cycles, the node is not expanded.

**Trimming:** At the end of each iteration, all solutions (there is one solution per leaf node) that do not result in MD-SC codes with the least number of active cycles are trimmed. If all children of a node are trimmed, that node is trimmed as well.

**Termination:** We proceed with expanding and trimming the tree of solutions, until no node is expanded in an iteration (the relocation process does not help anymore) or the maximum

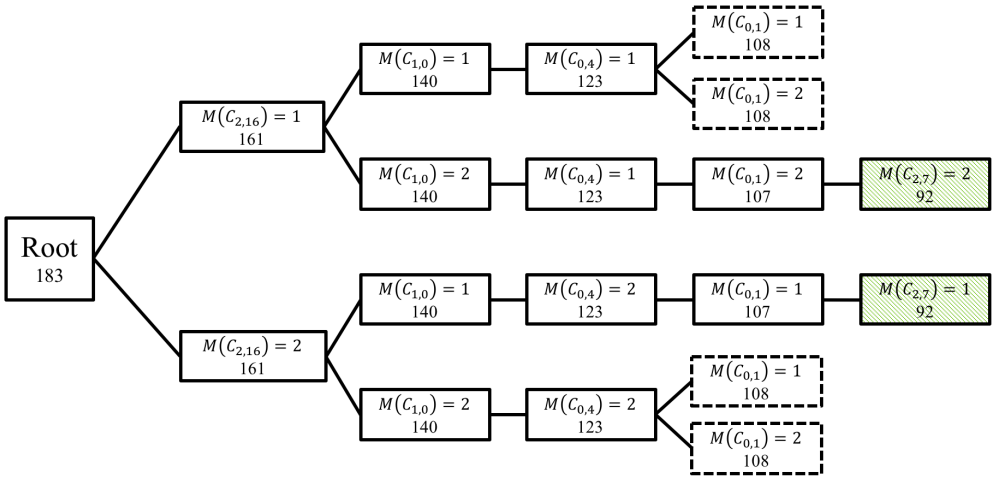



Figure 5.4: An illustration for a tree of solutions. The information associated with each node are the relocation option and the number of cycles-6 for the solution described by the path from the root up to this node. The nodes with dashed borders show the trimmed solutions. The nodes with hatch background show the winning solutions.

density is achieved (it happens at the end of iteration $\mathcal{T}$). Then, we construct the MD-SC code according to the relocations suggested by the nodes on the path from the root to a randomly chosen, non-trimmed, leaf.

**Example 13.** *Figure 5.4 illustrates an example for the tree of solutions to construct an MD-SC code with parameters* $L_2 = 3$, $d = 3$, *and* $\mathcal{T} = 5$ [2]. *At iteration* 1, *there are two winning relocation options for the targeted circulant, and they both result in* 161 *cycles*-6. *At iteration* 2, *each node at level* 1 *is expanded to two nodes. All* 4 *solutions result in* 140 *active cycles*-6. *At iteration* 3, *each node at level* 2 *is expanded to one node. All* 4 *solutions result in* 123 *active cycles*-6. *At iteration* 4, *the* $1^{st}$ *and* $4^{th}$ *nodes at level* 3 *are expanded to two nodes each, and the* $2^{nd}$ *and* $3^{rd}$ *nodes at level* 3 *are expanded to one node each. Among the* 6 *solutions, two of them result in* 107 *active cycles*-6, *and the remaining result in* 108 *active cycles*-6 *and are trimmed. At iteration* 5 *(the last iteration), each (non-trimmed) node at level* 4 *is expanded to one node. The two solutions (shown with leaves that have hatch backgrounds) result in* 92 *active cycles*-6, *and one of them can be chosen randomly for constructing the MD-SC code.*

[2]The remaining code parameters that result in this realization are $\kappa = z = 17$, $\gamma = 4$, $m = 1$, $L = 10$, girth 6, and OO-CPO technique for constructing the constituent SC codes.

Algorithm 2 shows the procedure for constructing MD-SC codes.

**Algorithm 2** Algorithm for Constructing MD-SC Codes

**inputs:** $\mathbf{H}_{\text{SC}}$, $k$, $L_2$, $d$, and $\mathcal{T}$.
**initialize:** A tree with one node (*root*), $l = 1$.
Find $\Gamma$, i.e., the set of all cycles-$k$ in $\mathbf{H}_{\text{SC}}$ that visit the circulants in the middle replica of $\mathbf{H}_{\text{SC}}$.
**while** $l \leq \mathcal{T}$ **and** there are nodes at level $l-1$ **do**
    **for** each *node* at level $l-1$ **do**
        Set $[M(\mathcal{C}_{i,j})]$ according to the relocations suggested by the path from *root* to *node*
        Find status (active/inactive) of cycles-$k$ in $\Gamma$ using IRC (i.e., (5.4)).
        $S = \{\mathcal{C}_{i,j} | \mathcal{C}_{i,j} \in \mathbf{R}_{\lceil L/2 \rceil} \text{ and } M(C_{i,j}) = 0\}$.
        Sort $S$ in a decreasing order according to the number of times they are visited by active cycles in $\Gamma$.
        Flag= 0.
        **while** $|S| > 0$ **and** Flag= 0 **do**
            Select the first circulant $\mathcal{C}_{i_v,j_v}$ in $S$ for relocation.
            Find best relocation options $\Phi$ for $\mathcal{C}_{i_v,j_v}$ by Algorithm 1.
            **if** $0 \in \Phi$ **then** $S = S \setminus \mathcal{C}_{i_v,j_v}$
            **else**
                Flag= 1.
                **for** $\forall t \in \Phi$ **do**
                    Add a child to *node* with content $M(C_{i_v,j_v}) = t$.
    Count the number of active cycles for each solution suggested by the nodes at level $l$.
    Trim all leaves (and their parents if needed) that do not result in minimum active cycles-$k$.
    $l = l + 1$.
Pick a random solution, set $M(C_{i,j})$ accordingly, and construct $\mathbf{H}_{\text{SC}}^{\text{MD}}$ using (5.2).
**output:** $\mathbf{H}_{\text{SC}}^{\text{MD}}$.

## 5.4 Simulation Results

Our simulation results demonstrate the outstanding performance of our new framework for constructing MD-SC codes. Sections 5.4.1 and 5.4.2 are dedicated to the analysis of MD-SC codes with girths 6 and 8, respectively. In each subsection, we study the effect of parameters $\mathcal{T}$, $d$, and $L_2$ on the performance of MD-SC codes. Additionally, we compare the MD-SC codes constructed by our new framework with their 1D-SC counterparts (1D-SC codes having the same length and nearly the same rate as the MD-SC codes). In Section 5.4.3, we compare the performance of our well-designed MD-SC codes with random constructions. In our

simulations, we consider the AWGN channel, and we use quantized min-sum algorithm with 4 bits and 15 iterations for the decoding.

### 5.4.1 Analysis for MD-SC Codes with Girth 6

We first describe the code parameters of SC Code 5.1 with girth 6, which is used as the constituent SC code in the rest of this subsection. SC Code 5.1 has parameters $\kappa = z = 17$, $\gamma = 4$, $m = 1$, $L = 10$, length 2,890 bits, and rate 0.741, and it is constructed by the OO-CPO technique. For SC Code 5.1, the partitioning matrix is:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

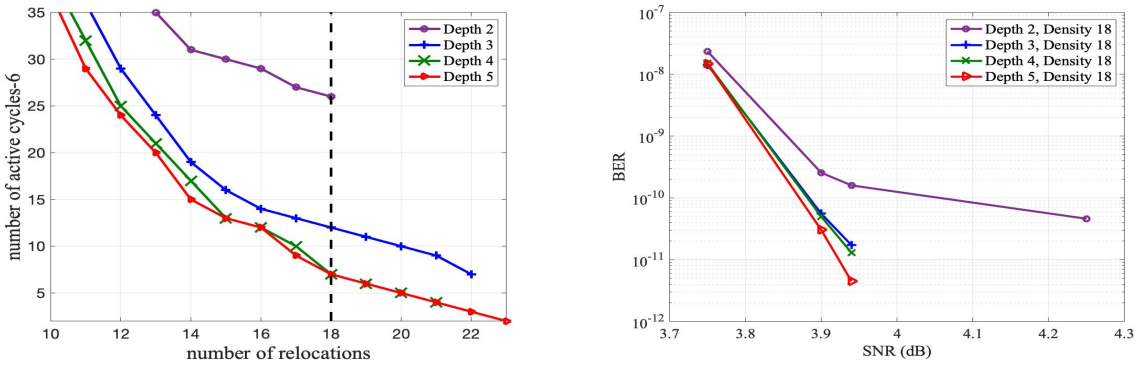
and the circulant power matrix is:

$$\begin{bmatrix} 0 & 10 & 2 & 8 & 2 & 0 & 5 & 7 & 15 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 11 & 15 & 2 & 14 & 10 & 3 & 6 & 7 & 8 & 9 & 4 & 11 & 12 & 8 & 14 & 10 & 16 \\ 11 & 2 & 4 & 12 & 8 & 11 & 12 & 9 & 15 & 4 & 13 & 5 & 6 & 1 & 11 & 13 & 15 \\ 11 & 3 & 6 & 9 & 2 & 16 & 8 & 4 & 7 & 10 & 13 & 16 & 2 & 5 & 8 & 6 & 14 \end{bmatrix}.$$

The cycles of interest here have length 6, i.e., $k = 6$.

First, we consider MD-SC codes with $L_2 = 5$ constructed by Algorithm 2. Figure 5.5 (a) shows the effect of increasing the MD coupling density $\mathcal{T}$ on the population of cycles-6 for various MD coupling depths. The horizontal axis shows $\mathcal{T}$, and the vertical axis shows the number of active cycles-6. We remind that an active cycle-$k$ is a cycle-$k$ that visits circulants of the middle replica of the constituent SC code and IRC (i.e., (5.4)) holds for it. As we see, increasing $\mathcal{T}$ does not decrease the population of active cycles-6 after 18 (resp. 23) relocations for depth 2 (resp., 5), resulting in an earlier termination for the smaller depth.

Table 5.1 shows the number of cycles-6 for MD-SC codes with $L_2 = 5$, MD coupling




(a) (b)

Figure 5.5: MD-SC codes with SC Code 5.1 as the constituent SC code and $L_2 = 5$: (a) The number of active cycles-6 for various densities and depths. (b) The BER curves over AWGN channel at density 26.47% and for various depths.

Table 5.1: Population of cycles-6 for MD-SC codes with SC Code 5.1 as the constituent SC code, $L_2 = 5$, and density 26.47%.

| MD coupling depth $d$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Number of active cycles-6 | 26 | 12 | 7 | 7 |
| Total number of cycles-6 | 20,825 | 9,775 | 5,695 | 5,610 |

density 18 (26.47% of circulants), and for various MD coupling depths. As we see, increasing the depth improves the cycle properties of the MD-SC codes. According to Table 5.1, MD-SC codes with depths 4 and 5 have similar number of active cycles-6, and the small difference in the total number of cycles-6 is due to the different multiplicity of the active cycles-6 in the final MD-SC codes. Figure 5.5 (b) shows a similar comparison in terms of the BER performance. For example, at SNR = 3.94 dB, the MD-SC code with depth 5 shows more than 1.5 orders of magnitude improvement in the BER performance compared to MD-SC code with depth 2.

Next, we study the effect of increasing the MD coupling length $L_2$ on the performance of MD-SC codes. We first describe the MD-SC codes and their 1D counterparts. MD-SC Code 5.1 has $L_2 = 1$, and it is in fact one instance of SC Code 5.1. MD-SC Code 5.2 has $L_2 = 3$, $d = 3$ (maximum depth), and $\mathcal{T} = 23$ (maximum density). After reaching the maximum density, relocation does not decrease the population of the cycles of interest. SC Code 5.2 is an SC code similar to SC Code 5.1 but with $L = 30$ (three times the coupling length of

Table 5.2: Population of cycles-6 for MD-SC codes and their 1D counterparts.

| code name | $L_2$ | length | rate | cycles-6 |
|---|---|---|---|---|
| MD-SC Code 5.1 (SC Code 5.1) | 1 | 2,890 | 0.741 | 29,274 |
| SC Code 5.2 | 1 | 8,670 | 0.757 | 91,494 |
| MD-SC Code 5.2 | 3 | 8,670 | 0.741 | 9,078 |
| SC Code 5.3 | 1 | 14,450 | 0.760 | 153,714 |
| MD-SC Code 5.3 | 5 | 14,450 | 0.741 | 1,700 |

SC Code 5.1), thus it has comparable length and rate to MD-SC Code 5.2. MD-SC Code 5.3 has $L_2 = 5$, $d = 5$ (maximum depth), and $\mathcal{T} = 23$ (maximum density). SC Code 5.3 is an SC code similar to SC Code 5.1 but with $L = 50$; thus it has comparable length and rate to MD-SC Code 5.3. The MD mapping matrix, i.e., $\mathbf{M} = [M(\mathcal{C}_{i,j})]$, for MD-SC Code 5.2 is:

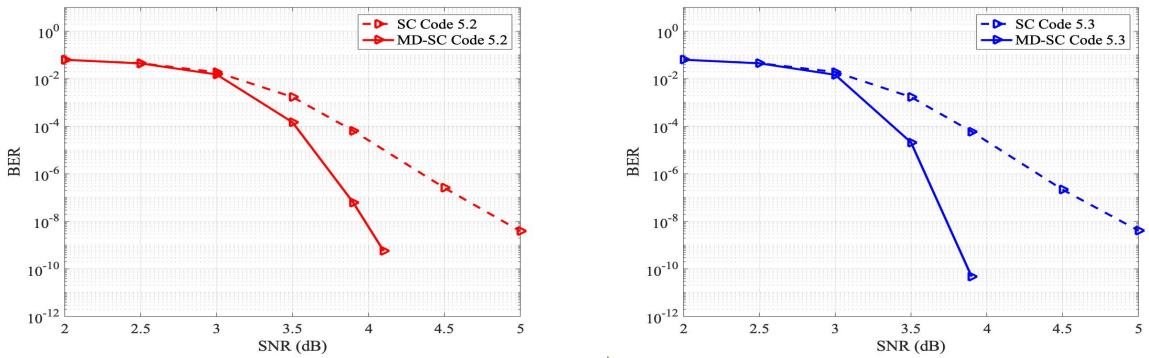
$$\begin{bmatrix} 0 & 1 & 2 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

and for MD-SC Code 5.3 is:

$$\begin{bmatrix} 0 & 3 & 3 & 0 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 3 & 0 & 0 & 0 & 0 & 3 & 0 & 3 & 4 & 0 & 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 3 & 4 & 3 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Table 5.2 shows the number of cycles-6 for this set of MD-SC codes and their 1D counterparts. MD-SC Code 5.2 has nearly 90% fewer cycles-6 compared to SC Code 5.2, and MD-SC Code 5.3 has nearly 99% fewer cycles-6 compared to SC Code 5.3. Furthermore, by increasing the number of constituent SC codes, although the overall code length increases, the number of cycles-6 decreases thanks to the higher amount of the MD coupling.

Figure 5.6 compares the BER performance of our MD-SC codes and their 1D-SC counterparts. MD-SC Code 5.2 shows about 4 orders of magnitude performance improvement




(a) (b)

Figure 5.6: The BER curves over AWGN channel for MD-SC codes compared to their 1D counterparts: (a) $L_2 = 3$, (b) $L_2 = 5$.

compared to SC Code 5.2 at SNR = 4.10 dB. This improvement is very pronounced for MD-SC Code 5.3 compared to SC Code 5.3 (about 6 orders of magnitude at SNR = 3.85 dB). These results demonstrate that the freedom offered by MD-SC codes is thoroughly exploited by our efficient construction framework, resulting in a large improvement in the BER performance. One interesting observation here is that although increasing the coupling length improves the BER performance for 1D-SC codes, the improvement becomes incremental for large values of $L$. Therefore, adding the MD coupling to achieve an even better error-correction is a promising choice.

### 5.4.2 Analysis for MD-SC Codes with Girth 8

We first describe the code parameters of SC Code 5.4 with girth 8, which is used as the constituent SC code in the rest of this subsection. SC Code 5.4 has parameters $\kappa = 19$, $z = 23$, $\gamma = 3$, $m = 2$, $L = 10$, length 4,370 bits, and rate 0.811, and it is constructed by the OO-CPO technique. For SC Code 5.4, the partitioning matrix is:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 2 & 0 & 2 & 2 & 0 & 1 & 1 & 0 & 1 & 2 & 0 & 2 & 2 & 2 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 & 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

and the circulant power matrix is:

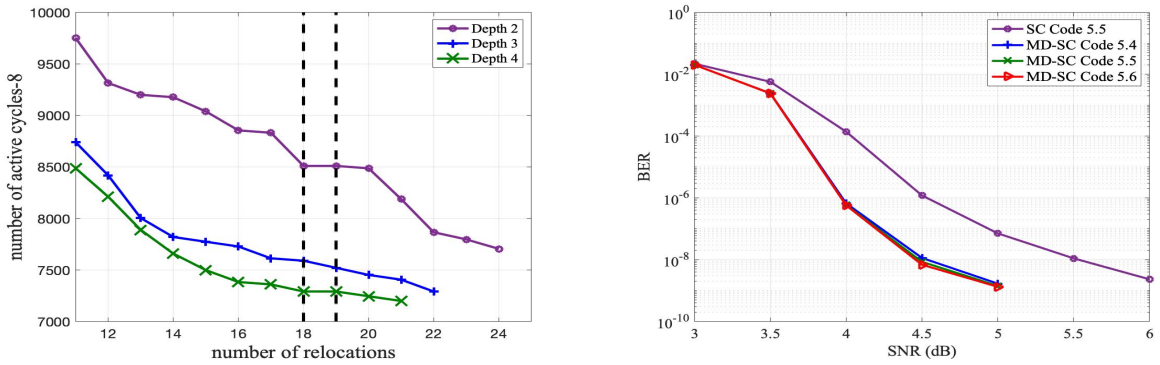
$$\begin{bmatrix} 21 & 0 & 16 & 3 & 19 & 1 & 0 & 0 & 21 & 5 & 0 & 0 & 1 & 0 & 9 & 0 & 16 & 1 & 0 \\ 0 & 11 & 7 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \\ 0 & 17 & 0 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 & 22 & 1 & 3 & 5 & 19 & 9 & 11 & 13 \end{bmatrix}.$$

The cycles of interest here have length 8, i.e., $k = 8$.

First, we consider MD-SC codes with $L_2 = 4$ constructed by Algorithm 2. Figure 5.7 (a) shows the effect of increasing the MD coupling density $\mathcal{T}$ on the population of cycles-8 for various MD coupling depths. We have two interesting observations here: First, increasing $\mathcal{T}$ does not decrease the population of active cycles-8 after 24 (resp. 22 and 21) relocations for depth 2 (resp., 3 and 4), implying that a larger depth does not necessarily result in an earlier termination. Second, for some relocations, although the population of active cycles-8 does not decrease, Algorithm 2 proceeds with relocations (for example, see relocations 18'th and 19'th in Figure 5.7 (a)). This is because although these relocations do not reduce the population of the shortest cycles (cycles with length 8 here), they reduce the population of cycles with length $2k$ (cycles with length 16 here).

Next, we study the BER performance of MD-SC codes with various depths and their 1D-SC counterpart. We first describe the code parameters. MD-SC Code 5.4, MD-SC Code 5.5, and MD-SC Code 5.6 have $L_2 = 4$, $\mathcal{T} = 19$, length 17,480 bits, and rate 0.811. MD-SC Code 5.4, resp. MD-SC Code 5.5 and MD-SC Code 5.6, has depth 2, resp., 3 and 4. SC Code 5.5 is an SC code similar to SC Code 5.4 but with $L = 40$ (four times the coupling length of SC Code 5.4); thus it has comparable length and rate to MD-SC Codes 5.4, 5.5, and 5.6 (length 17,480 and rate 0.834). The MD mapping matrix for MD-SC Code 5.4 is:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$




(a) (b)

Figure 5.7: MD-SC codes with SC Code 5.4 as the constituent SC code and $L_2 = 4$: (a) The number of active cycles-8 for various densities and depths. (b) The BER curves over AWGN channel at density 25% and for various depths along with the BER performance for the 1D-SC counterpart (SC Code 5.5).

the MD mapping matrix for MD-SC Code 5.5 is:

$$\begin{bmatrix} 0 & 2 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 2 & 0 & 0 \end{bmatrix},$$

and the MD mapping matrix for MD-SC Code 5.6 is:

$$\begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 0 & 3 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 & 0 & 1 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 \end{bmatrix}.$$

According to Figure 5.7 (b), MD-SC Codes 5.4, 5.5, and 5.6 show about 2 orders of magnitude performance improvement compared to SC Code 5.5 at SNR = 4.50 dB. Table 5.3 shows the number of cycles-8 for SC Code 5.5 and MD-SC Codes 5.4, 5.5, and 5.6. MD-SC Code 5.6 has nearly 82% fewer cycles-8 compared to SC Code 5.5 and nearly 15% fewer cycles-8 compared to MD-SC Code 5.4. As we see, the MD coupling considerably improves the performance of the SC codes; however, the improvement by increasing the MD coupling depth is small in this case, and thus, using a lower depth is sufficient to achieve a good

Table 5.3: Population of cycles-8 for MD-SC codes and their 1D counterparts.

| code name | number of active cycles-8 | total number of cycles-8 |
|---|---|---|
| SC Code 5.5 | – | 1,397,319 |
| MD-SC Code 5.4 | 8,510 | 292,560 |
| MD-SC Code 5.5 | 7,521 | 258,060 |
| MD-SC Code 5.6 | 7,291 | 249,320 |

error-correction performance.

### 5.4.3 Comparison with Random Constructions

Previous works on MD-SC codes, while promising, either consider random constructions or are limited to specific topologies. In this subsection, we compare our new MD-SC code construction with random constructions for connecting several SC codes together. Random constructions are inspired by [48, 49, 52, 53], where the purpose of random constructions is performing an ensemble asymptotic analysis over a family of the MD-SC codes. In order to perform a fair comparison, all MD-SC codes in this subsection have the same constituent SC code, i.e., SC Code 5.6. SC Code 5.6 has parameters $\kappa = z = 17$, $\gamma = 3$, $m = 1$, $L = 15$, length 4,335 bits, and rate 0.812, and it is constructed by the OO-CPO technique. For SC Code 5.6, the partitioning matrix is:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix},$$

and the circulant power matrix is:

$$\begin{bmatrix} 0 & 0 & 2 & 9 & 0 & 7 & 4 & 16 & 2 & 4 & 2 & 9 & 0 & 4 & 13 & 1 & 1 \\ 13 & 1 & 2 & 6 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 13 & 12 & 0 & 14 & 8 & 16 \\ 0 & 2 & 0 & 0 & 8 & 10 & 8 & 14 & 16 & 1 & 3 & 5 & 7 & 15 & 5 & 5 & 2 \end{bmatrix}.$$

The cycles of interest here have length 6, i.e., $k = 6$.

MD-SC Codes 5.7, 5.8, 5.9, 5.10 have $L_2 = 3$, $\mathcal{T} = 9$, length 13,005 bits, and rate 0.812. MD-SC Codes 5.7 and 5.8, have depths 2 and 3, respectively, and they are constructed by Algorithm 2. The MD mapping matrix for MD-SC Codes 5.7 is:

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

and the MD mapping matrix for MD-SC Codes 5.8 is:

$$\begin{bmatrix} 2 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \end{bmatrix}.$$

MD-SC Codes 5.9 and 5.10 are constructed by random relocations, and they both have depth 2. For MD-SC Code 5.9, the relocated circulants are chosen uniformly at random, and similar relocations are applied to all replicas of one constituent SC code. However, different constituent SC codes can have different relocations. MD-SC Code 5.10 is constructed in a similar way to MD-SC Code 5.9, but the same relocations are applied to all constituent SC codes. The later random construction has the benefit of avoiding the creation of cycles-4 if the constituent SC codes do not have cycles-4.

Table 5.4 shows the population of short cycles for MD-SC Codes 5.7, 5.8, 5.9, and 5.10. As we see, MD-SC Code 5.7 has 65% fewer cycles-6 compared to MD-SC Code 5.10, and they both have zero cycles-4. These two codes have they same structure, but the relocated circulants are chosen randomly for MD-SC Code 5.7, while they are chosen to specifically reduce the number of cycles-6 for MD-SC Code 5.10. MD-SC Code 5.8, which is similar to MD-SC Code 5.7 but with depth 3, has zero cycles-6 and 6.1% fewer cycles-8 compared to MD-SC Code 5.7. MD-SC Code 5.9 is similar to MD-SC Code 5.10, but without the constraint of similar relocations for all constituent SC codes, thus it could not preserve the girth of the constituent SC codes and has cycles-4. Figure 5.8 shows the BER performance

Table 5.4: Population of short cycles for MD-SC codes constructed by various policies.

| code name | cycles-4 | cycles-6 | cycles-8 |
|---|---|---|---|
| MD-SC Code 5.7 | 0 | 2,856 | 685,032 |
| MD-SC Code 5.8 | 0 | 0 | 643,110 |
| MD-SC Code 5.9 | 255 | 9,010 | 585,820 |
| MD-SC Code 5.10 | 0 | 8,211 | 606,543 |

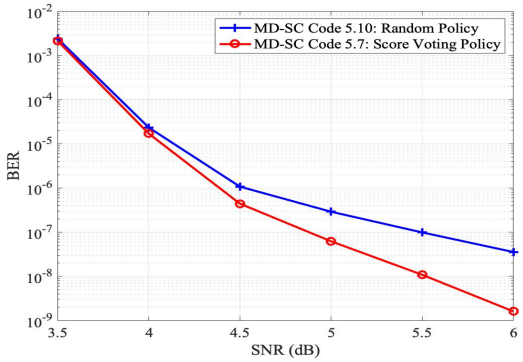



Figure 5.8: The BER curves over AWGN channel for MD-SC codes with SC Code 5.6 as the constituent SC code, $L_2 = 3$, density 18%, and constructed based on a random policy and our new score-voting policy.

for MD-SC Code 5.7 and MD-SC Code 5.10. These two codes both have depth 2 and have the MD structure described in (5.2). At SNR = 6.0 dB, MD-SC Code 5.7 shows nearly 1.3 orders of magnitude BER improvement compared to MD-SC Code 5.10.

## 5.5 Conclusion

We expanded the repertoire of SC codes by establishing a framework for MD-SC code construction with an arbitrary number of constituent SC codes and an arbitrary multi-dimensional coupling depth. For MD coupling, we rewire connections (relocate circulants) that are most problematic within each SC code. Our framework encompasses a systematic way to sequentially identify and relocate problematic circulants, thus utilizing them to connect the constituent SC codes. Our MD-SC codes show a notable reduction in the population of the small cycles and a significant improvement in the BER performance compared to the 1D setting.

## Acknowledgment

Parts of the materials presented in this chapter is published in [59], and the majority of the materials will appear in [17]. The author would like to thank the collaborators in these manuscripts.

# CHAPTER 6

# Conclusion

## 6.1 Summary of Our Results

We are living in an era where a huge amount of data is generated everyday, with a rapidly growing rate, that need to be stored and transmitted with extremely high reliability. This necessitates the design of high performance ECCs that operate at very low error rate and thus can compensate for the channel imperfections. The channel can be the medium that the (encoded) date is passed through to be received at the destination, or it can be an storage device. The focus of this dissertation is more on the latter case.

This dissertation dealt with the analysis and design of an attractive class of graph-based ECCs, called spatially-coupled (SC) codes, which have recently received significant attention due to their capacity approaching performance and low-latency decoding. We deftly analyzed the structure of SC codes and presented low-complexity tools to evaluate the performance of SC codes in terms of the number of problematic objects in their graphs. We first presented a systematic approach to efficiently exploit the available degrees of freedom and design SC codes that offer superior performance over a variety of channel setting. Next, we expanded our presented SC code machinery by adding irregularity to the design. Last but not the least, we increased the coupling dimensionality and presented the first comprehensive multi-dimensional SC design framework with enhanced finite-length properties.

An unlabeled circulant-based SC code can be designed in two steps: partitioning and adjusting circulant powers. In Chapter 2, we presented the first optimal partitioning scheme that results in the minimum number of problematic objects in protographs of SC codes. We also presented a low-complexity heuristic program to adjust circulant powers to further reduce

the population of problematic objects in the lifted graph. Our simulation results demonstrate a dramatic performance gain attained using our two-stage program over AWGN channels.

In Chapter 3, we investigated the practical non-AWGN settings, and demonstrated how our SC design can be tailored to incorporate the properties of the underlying channel model. Especially, for magnetic recording channels and Flash channels, we identified the problematic objects that are responsible for most errors in the error floor region. We then derived the common denominator of these problematic objects and customized the stages of our SC design framework to reduce the population of the common denominator instances in the graph of SC codes. We also considered non-uniform channels with SNR variation and presented SC construction along with a novel interleaving scheme that alleviate the negative impacts of the channel non-uniformity.

In Chapter 4, we expanded our design framework into the realm of irregular SC codes. We presented a novel scheme for optimal partitioning of irregular underlying block codes. We introduced a dummy component matrix that can be representative for the irregularity of the underlying block code, and incorporated the new component matrix into our optimal partitioning scheme to design irregular SC protographs with minimum number of problematic objects.

In Chapter 5, we presented the first finite-length systematic framework for constructing multi-dimensional (MD)-SC codes with enhanced cycle properties. Short cycles are known to negatively affect the performance of graph-based codes under iterative decoding. For constructing MD-SC codes, we connect several constituent SC codes together by rewiring their most problematic connections. We rewire connections such that the minimum number of short cycles are formed in the MD design. Our MD-SC codes can still be decoded using a (customized) low-latency windowed decoding thanks to the constraints that we considered in our MD design.

Our simulation results demonstrate a dramatic performance gain attained using our SC design framework over various channel models. The materials presented in this dissertation can be considered as a comprehensive study of finite-length SC codes, and can be used for

a variety of settings: regular or irregular, one-dimensional or multi-dimensional, and over different channel models.

## 6.2 Future Directions

In this dissertation, we presented a comprehensive study of finite-length SC codes. We presented a code design methodology that in a channel-aware way exploits the structure of dominant errors to systematically set the partitioning choice and circulant powers of an SC code. We also expanded the repertoire of SC codes by establishing a framework for MD-SC code construction. A fruitful future direction is to customize the proposed code design methodology for 3D Flash memories and 2D magnetic recording applications. Another promising research direction is to present an efficient low-latency decoding by incorporating MD locality into the decoder implementation. Moreover, one can incorporate the decoder setting, e.g., the window size, in the code design to present decoder-aware MD-SC codes that are adjusted to perform well for specific decoding properties.

Moreover, all algorithms presented in this dissertation, either those that identify the optimal solutions or those that are heuristic, can be improved by reducing their computational complexity. The improvement can be realized by further incorporating the code structure into the design, or by borrowing tools and techniques from the Combinatorics, Probability theory, Graph theory, etc. Any progress in this direction is significantly valuable as it brings the opportunity for designing SC codes with arbitrary memories, column weights, rates, etc; thus, it provides further design freedom.

# REFERENCES


[1] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, p. 1645, Aug. 1996.

[3] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.

[4] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 2277–2292, Apr. 2013.

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

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

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

[8] D. G. M. Mitchell, L. Dolecek, and D. J. Costello, "Absorbing set characterization of array-based spatially coupled LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Honolulu, HI, USA, Jun. 2014, pp. 886–890.

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

[10] D. G. M. Mitchell and E. Rosnes, "Edge spreading design of high rate array-based SC-LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 2940–2944.

[11] L. Chen, S. Mo, D. J. Costello, D. G. M. Mitchell, and R. Smarandache, "A protograph-based design of quasi-cyclic spatially coupled LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1683–1687.

[12] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Finite-length construction of high performance spatially-coupled codes via optimized partitioning and lifting," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 3–16, Jan. 2019.

[13] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Spatially coupled codes optimized for magnetic recording applications," *IEEE Transactions on Magnetics*, vol. 53, no. 2, pp. 1–11, Feb. 2017.

[14] A. Hareedy, H. Esfahanizadeh, and L. Dolecek, "High performance non-binary spatially-coupled codes for flash memories," in *Proc. IEEE Information Theory Workshop (ITW)*, Kaohsiung, Taiwan, Nov. 2017, pp. 229–233.

[15] H. Esfahanizadeh, A. Hareedy, R. Wu, R. Galbraith, and L. Dolecek, "Spatially-coupled codes for channels with SNR variation," *IEEE Transactions on Magnetics*, vol. 54, no. 11, pp. 1–5, Nov. 2018.

[16] H. Esfahanizadeh, R. Wu, and L. Dolecek, "A finite-length construction of irregular spatially-coupled codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Visby, Gotland, Sweden, Aug. 2019.

[17] H. Esfahanizadeh, L. Tauz, and L. Dolecek, "Multi-dimensional spatially-coupled code design: Enhancing the cycle properties," *IEEE Transactions on Communications*, To be published 2020.

[18] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

[19] J. L. Fan, "Array codes as low-desity parity-check codes," in *Proc. International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, Sep. 2000, pp. 543–546.

[20] Y. Wang, S. C. Draper, and J. S. Yedidia, "Hierarchical and high-girth QC LDPC codes," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4553–4583, Jul. 2013.

[21] T. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, Oct. 2003, pp. 1426–1435.

[22] M. Karimi and A. H. Banihashemi, "On characterization of elementary trapping sets of variable-regular LDPC codes," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5188–5203, Sep. 2014.

[23] 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 Transactions on Information Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[24] A. Hareedy, B. Amiri, R. Galbraith, and L. Dolecek, "Non-binary LDPC codes for magnetic recording channels: Error floor analysis and optimized code design," *IEEE Transactions on Communications*, vol. 64, no. 8, pp. 3194–3207, Aug. 2016.

[25] T. Parnell, N. Papandreou, T. Mittelholzer, and H. Pozidis, "Modelling of the threshold voltage distributions of sub-20nm NAND flash memory," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Austin, TX, USA, Dec. 2014, pp. 2351–2356.

[26] A. Hareedy, C. Lanka, and L. Dolecek, "A general non-binary LDPC code optimization framework suitable for dense flash memory and magnetic storage," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2402–2415, Sep. 2016.

[27] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[28] J. Wang, L. Dolecek, and R. D. Wesel, "The cycle consistency matrix approach to absorbing sets in separable circulant-based LDPC codes," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 2293–2314, Apr. 2013.

[29] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[30] A. Bazarsky, N. Presman, and S. Litsyn, "Design of non-binary quasi-cyclic LDPC codes by ACE optimization," in *Proc. IEEE Information Theory Workshop (ITW)*, Seville, Spain, Sep. 2013, pp. 1–5.

[31] K. Huang, D. G. M. Mitchell, L. Wei, X. Ma, and D. J. Costello, "Performance comparison of LDPC block and spatially coupled codes over GF($q$)," *IEEE Transactions on Information Theory*, vol. 63, no. 3, pp. 592–604, Mar. 2015.

[32] N. ul Hassan, M. Lentmaier, and G. P. Fettweis, "Comparison of LDPC block and LDPC convolutional codes based on their decoding latency," in *Proc. International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Gothenburg, Sweden, Aug. 2012, pp. 225–229.

[33] S. V. Maiya, D. J. Costello, T. E. Fuja, and W. Fong, "Coding with a latency constraint: The benefits of sequential decoding," in *Proc. Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Allerton, IL, USA, Sep. 2010, pp. 201–207.

[34] Y. Fang, P. Chen, L. Wang, and F. C. M. Lau, "Design of protograph LDPC codes for partial response channels," *IEEE Transactions on Communications*, vol. 60, no. 10, pp. 2809–2819, Oct. 2012.

[35] Y. Han and W. E. Ryan, "Low-floor detection/decoding of LDPC-coded partial response channels," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, pp. 252–260, Feb. 2010.

[36] H. Zhong, T. Zhong, and E. F. Haratsch, "Quasi-cyclic LDPC codes for the magnetic recording channel: Code design and VLSI implementation," *IEEE Transactions on Magnetics*, vol. 43, no. 3, pp. 1118–1123, Mar. 2007.

[37] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF($q$)," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[38] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[39] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 4, pp. 730–743, Apr. 2001.

[40] A. Hareedy, H. Esfahanizadeh, A. Tan, and L. Dolecek, "Spatially-coupled code design for partial-response channels: Optimal object-minimization approach," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.

[41] Y. Maeda and H. Kaneko, "Error control coding for multilevel cell flash memories using nonbinary low-density parity-check codes," in *proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, Chicago, IL, USA, Oct. 2009, pp. 367–375.

[42] N. Varnica, G. Burd, and Z. Wu, "Interleaved error correction coding for channels with non-uniform SNRs," Nov. 2012, US Patent 8,312,341. [Online]. Available: https://www.google.ch/patents/US8312341

[43] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "A novel combinatorial framework to construct spatially-coupled codes: Minimum overlap partitioning," in *proc. IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1693–1697.

[44] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[45] X. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[46] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[47] A. Beemer and C. A. Kelley, "Avoiding trapping sets in SC-LDPC codes under windowed decoding," in *Proc. International Symposium on Information Theory and Its Applications (ISITA)*, Monterey, CA, USA, Oct. 2016, pp. 206–210.

[48] D. Truhachev, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "New codes on graphs constructed by connecting spatially coupled chains," in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, USA, Feb. 2012, pp. 392–397.

[49] R. Ohashi, K. Kasai, and K. Takeuchi, "Multi-dimensional spatially-coupled codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 2448–2452.

[50] D. Truhachev, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Connecting spatially coupled LDPC code chains," in *Proc. IEEE International Conference on Communications (ICC)*, Ottawa, Canada, Jun. 2012, pp. 2176–2180.

[51] P. M. Olmos, D. G. M. Mitchell, D. Truhachev, and D. J. Costello, "A finite length performance analysis of LDPC codes constructed by connecting spatially coupled chains," in *Proc. IEEE Information Theory Workshop (ITW)*, Sevilla, Spain, Sep. 2013, pp. 1–5.

[52] L. Schmalen and K. Mahdaviani, "Laterally connected spatially coupled code chains for transmission over unstable parallel channels," in *Proc. International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Bremen, Germany, Aug. 2014, pp. 77–81.

[53] Y. Liu, Y. Li, and Y. Chi, "Spatially coupled LDPC codes constructed by parallelly connecting multiple chains," *IEEE Communications Letters*, vol. 19, no. 9, pp. 1472–1475, Sep. 2015.

[54] R. Tanaka and K. Ishibashi, "Robust coded cooperation based on multi-dimensional spatially-coupled repeat-accumulate codes," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[55] I. Ali, H. Lee, A. Hussain, and S. Kim, "Protograph-based folded spatially coupled LDPC codes for burst erasure channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 516–519, Apr. 2019.

[56] R. Smarandache and P. O. Vontobel, "Quasi-cyclic LDPC codes: Influence of proto- and tanner-graph structure on minimum hamming distance upper bounds," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 585–607, Feb. 2012.

[57] M. Battaglioni, M. Baldi, and G. Cancellieri, "Connections between low-weight codewords and cycles in spatially coupled LDPC convolutional codes," *IEEE Transactions on Communications*, vol. 66, no. 8, pp. 3268–3280, Aug. 2018.

[58] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[59] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Multi-dimensional spatially-coupled code design through informed relocation of circulants," in *Proc. Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, Oct. 2018, pp. 695–701.