

Multi-Dimensional Spatially-Coupled Code Design: Enhancing the Cycle Properties

Homa Esfahanizadeh¹, Student Member, IEEE, Lev Tauz, Student Member, IEEE,
and Lara Dolecek², Senior Member, IEEE

Abstract—A circulant-permutation-based spatially-coupled (SC) code is constructed by partitioning the circulant permutation matrices (CPMs) in the parity-check matrix of a block code into several components and piecing copies of these components in a diagonal structure. By connecting several SC codes, multi-dimensional SC (MD-SC) codes are constructed. In this paper, we present a systematic framework for constructing 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 CPMs. 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. Finally, we present a decoding algorithm that utilizes the structures of the MD-SC code to achieve lower decoding latency. Compared to the conventional SC codes, our MD-SC codes have a notably lower population of small cycles, and a dramatic BER improvement. The results of this work can be particularly beneficial in data storage systems, e.g., 2D magnetic recording and 3D Flash systems, as high-performance MD-SC codes are robust against various channel impairments and non-uniformity.

Index Terms—Circulant permutation matrix, cycles, error floor, finite-length, LDPC codes, multi-dimensional codes, relocation, spatially-coupled codes.

I. INTRODUCTION

Spatially-coupled (SC) codes are a family of graph-based codes that have attracted significant attention thanks to their capacity approaching performance. One-dimensional SC (1D-SC) codes are constructed by coupling a series of disjoint block codes into a single coupled chain [3]. Here, we use circulant-permutation-based LDPC codes [4] as the underlying block codes. 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 [7]–[9].

Multi-dimensional SC (MD-SC) codes can be constructed by coupling several 1D-SC codes together via rewiring the existing connections or by adding extra variable nodes (VNs) or check nodes (CNs) [10], [11]. 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 [10], [11]. MD-SC codes were introduced in [10], [11] and investigated more in [12]–[17].

In [10], [12], [13], constructions are presented for MD-SC codes that have specific structures, e.g., loops and triangles. The construction method for MD-SC codes presented in [11] involves connecting edges uniformly at random such that some criteria on the number of connections are satisfied. In [14], 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 [15], multiple SC codes are connected by random edge exchanges between adjacent chains to improve the iterative decoding threshold. In [16], [17], 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 focus on using 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. In [13], a finite-length analysis in the waterfall region for MD-SC codes with a loop structure is presented.

Finding the best connections to be rewired in order to connect constituent 1D-SC codes and construct MD-SC codes with outstanding finite-length performance is still an open problem. This paper is the first work to present a

Manuscript received April 9, 2019; revised August 7, 2019, October 29, 2019, and December 28, 2019; accepted January 23, 2020. Date of publication February 5, 2020; date of current version May 15, 2020. This work was supported in part by UCLA Dissertation Year Fellowship, a grant from ASRC-IDEMA, and NSF CCF-BSF:CI1718389. This article was presented in part at the 56th Annual Allerton Conference on Communication, Control, and Computing 2018, and in part at the 10th Annual Non-Volatile Memories Workshop 2019. The associate editor coordinating the review of this article and approving it for publication was Q. Huang. (Corresponding author: Homa Esfahanizadeh.)

The authors are with the Department of Electrical and Computer Engineering, University of California, Los Angeles (UCLA), Los Angeles, CA 90095 USA (e-mail: hesfahanizadeh@ucla.edu; levtau@ucla.edu; dolecek@ee.ucla.edu).

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

Digital Object Identifier 10.1109/TCOMM.2020.2971694

0090-6778 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

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. This paper is an extended version of our work published in [1]. We extend our previous work by: (1) connecting an arbitrary number of SC codes at a desired MD coupling depth to construct MD-SC codes; (2) converting the instances of the short cycles in the constituent SC codes to cycles of the largest possible length in the MD-SC code; and (3) presenting a low-latency decoder that exploits the structure of the constituent SC codes along with the structure of the final MD-SC code.

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 circulant permutation matrices (CPMs), corresponding 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.

The rest of the paper is organized as follows. In Section II, the necessary preliminaries are briefly reviewed. In Section III, the structure of our MD-SC codes is presented. In Section IV, our novel framework for constructing MD-SC codes with enhanced cycle properties is introduced. In Section V, a low-latency algorithm for decoding MD-SC codes is presented. In Section VI, our simulation results are given. Finally, the conclusion appears in Section VII.

II. PRELIMINARIES

Throughout this paper, each column (resp., row) in a parity-check matrix corresponds to a VN (resp., CN) in the equivalent graph of the matrix. Regular circulant-permutation-based codes are (γ, κ) LDPC codes, where γ is the column weight of the parity-check matrix (VN degree), and κ is the row weight (CN degree). The parity-check matrix $\mathbf{H}$ of a circulant-permutation-based code is constructed as follows:

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

Each CPM 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 σ is the $z \times z$ identity matrix cyclically shifted one unit to the left. The term $f_{i,j}$ specifies the power of the CPM at row group index i and column group index j . We use circulant-permutation-based codes as the underlying block codes of SC codes. We highlight that, in this paper, each CPM in (1) is a permutation of an identity matrix. Thus, each circulant matrix has weight 1. Circulants with larger weights have a negative

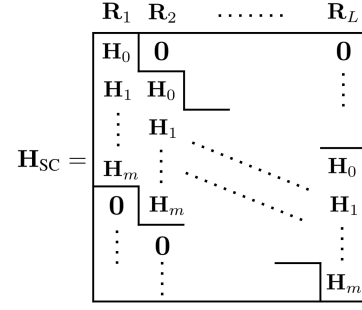


Fig. 1. The parity-check matrix of an SC code with parameters m and L .

impact on the girth [18], and we do not use them in our code construction since the ultimate goal is to improve the cycle properties.

The parity-check matrix $\mathbf{H}_{SC}$ of a circulant-permutation-based SC code is constructed by partitioning the $\kappa\gamma$ CPMs of the underlying block code into $(m+1)$ component matrices $\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_m$ (with the same size as $\mathbf{H}$), and piecing L copies of the component matrices together as shown in Fig. 1. The parameter m is called the memory, and the parameter L is called the coupling length. Each component matrix $\mathbf{H}_l$, $0 \leq l \leq m$, has a subset of CPMs of $\mathbf{H}$ and zeros elsewhere so that $\sum_{l=0}^m \mathbf{H}_l = \mathbf{H}$. A replica $\mathbf{R}_\nu$, $1 \leq \nu \leq L$, is a submatrix of $\mathbf{H}_{SC}$ that has one submatrix $[\mathbf{H}_0^T \dots \mathbf{H}_m^T]^T$, Fig. 1.

Recently, a systematic framework for partitioning the underlying block code and optimizing the CPM powers, known as the optimal partitioning and circulant power optimizer (OO-CPO) technique, was proposed for constructing high-performance SC codes [7], [19]. In this paper, 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.

Short cycles have a negative impact on the performance 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., absorbing sets and trapping sets [20], [21], are formed of cycles with relatively short lengths [7], [19], [22], [23]. Finally, short cycles can have a negative impact on the code minimum distance. In [24], [25], some upper bounds on the minimum distance of circulant-based block and SC LDPC 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.

We present a systematic framework to construct MD-SC codes, which is based on an informed relocation of CPMs. 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.

Throughout this paper, the operator $\stackrel{p}{=}$ (resp., $\stackrel{p}{\neq}$) defines the congruence (resp., incongruence) modulo p , and the operator $(\cdot)_p$ defines modulo p of an integer.

III. MD-SC 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.

We intend to reduce the population of cycles with length k , or 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 [26], or the length of the cycle that is the common denominator of several problematic combinatorial objects for a specific channel, e.g., AWGN channels [7], partial response channels [27], or Flash channels [28]. For instance, a cycle-6 is the common denominator of problematic combinatorial objects for AWGN channels, and a cycle-8 is the common denominator of problematic combinatorial objects for partial response channels even if the girth is 6.¹

An *Auxiliary matrix* $\mathbf{A}_t$, $t \in \{1, \dots, L_2 - 1\}$, 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 (3). The auxiliary matrices are all-zero matrices at the beginning of the framework and are filled with CPMs during the construction process. A *relocation* is defined as moving a CPM 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 $\kappa\gamma$ CPMs in this replica. Out of these $\kappa\gamma$ CPMs, we choose $\mathcal{T}$ CPMs 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 CPMs to auxiliary matrices $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{d-1}$ such that a relocated CPM 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,

$$\mathbf{H}_{\text{SC}} = \mathbf{H}'_{\text{SC}} + \sum_{t=1}^{d-1} \mathbf{A}_t, \quad (2)$$

where $\mathbf{H}'_{\text{SC}}$ is derived from $\mathbf{H}_{\text{SC}}$ by removing the $\mathcal{T}$ chosen CPMs. We note that the middle replica $\mathbf{R}_\nu$ is considered for ranking the CPMs in order to include all possible cycles- k that a CPM in $\mathbf{H}_{\text{SC}}$ can contribute to. The parity-check matrix of the MD-SC code, $\mathbf{H}_{\text{SC}}^{\text{MD}}$, is constructed as follows, where $\mathbf{A}_d = \mathbf{A}_{d+1} = \dots = \mathbf{A}_{L_2-1} = \mathbf{0}$: (The non-zero auxiliary matrices are $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{d-1}$.)

$$\mathbf{H}_{\text{SC}}^{\text{MD}} = \begin{bmatrix} \mathbf{H}'_{\text{SC}} & \mathbf{A}_{L_2-1} & \cdots & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{H}'_{\text{SC}} & \cdots & \mathbf{A}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{L_2-1} & \mathbf{A}_{L_2-2} & \cdots & \mathbf{H}'_{\text{SC}} \end{bmatrix}. \quad (3)$$

¹We note that while all cycles of equal length may not be equally detrimental in general, we treat them the same to simplify our design. As we will show later in the paper, our presented algorithm works on a list of cycles to remove by multi-dimensional rewiring. This list can be further filtered by the user to only include the most detrimental cycles.

$\mathbf{H}_{\text{SC}}^{\text{MD}}$ 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 $t \in \{1, \dots, d-1\}$, and $\mathbf{S}_{(a+t)L_2,a} = \mathbf{0}$ for $t \in \{d, \dots, L_2-1\}$.² As such, the code does not reduce to a block LDPC code even when $d = L_2$.

Example 1. 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 CPM, and we relocate it to $\mathbf{A}_1$. This relocation is applied to all $L = 3$ instances of the problematic CPM. We remind that each CPM corresponds to a group of z connections in the graph of the SC code. Four constituent SC codes along with their problematic connections are depicted in Fig. 2(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, Fig. 2(b).

Next, we introduce some necessary definitions:

Definition 1.

- 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 CPM in $\mathbf{H}_{\text{SC}}$. We say $\mathcal{C}_{i,j}$ is relocated to $\mathbf{A}_t$, where $t \in \{1, \dots, d-1\}$, 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 $\mathcal{C}_{i,j}$ 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 $\mathbf{0}$ ($a \in \{0, \dots, L_2-1\}$ and $t \in \{1, \dots, d-1\}$).
- 3) The MD mapping $M : \{\mathcal{C}_{i,j}\} \rightarrow \{0, \dots, d-1\}$ is a mapping from a CPM in $\mathbf{H}_{\text{SC}}$ to an integer in $\{0, \dots, d-1\}$, 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 , denoted by $\mathcal{O}_k$, visits k CPMs in the parity-check matrix of the code. We list the k CPMs of $\mathcal{O}_k$, according to the order they are visited when the cycle is traversed in a clockwise direction, in a sequence as $\mathcal{C}_{\mathcal{O}_k} = \{\mathcal{C}_{i_1,j_1}, \mathcal{C}_{i_2,j_2}, \dots, \mathcal{C}_{i_k,j_k}\}$, where $i_1 = i_2, j_2 = j_3, \dots, i_{k-1} = i_k, j_k = j_1$. A CPM can be visited more than once, e.g., Fig. 3(b).
- 5) We denote the distance between $\mathcal{C}_{i_u,j_u}$ and $\mathcal{C}_{i_v,j_v}$ on $\mathcal{O}_k$, where $u, v \in \{1, \dots, k\}$, as $D_{\mathcal{O}_k}(\mathcal{C}_{i_u,j_u}, \mathcal{C}_{i_v,j_v}) \in \{0, \dots, k-1\}$. $D_{\mathcal{O}_k}(\mathcal{C}_{i_u,j_u}, \mathcal{C}_{i_v,j_v}) = |v - u|$.

In the new MD-SC code design framework, we effectively answer two questions: which CPMs to relocate, and where to relocate them.

²While $\mathbf{H}_{\text{SC}}^{\text{MD}}$ may look similar to a block LDPC code, we would like to note that the locality of connections is always bounded during relocations.

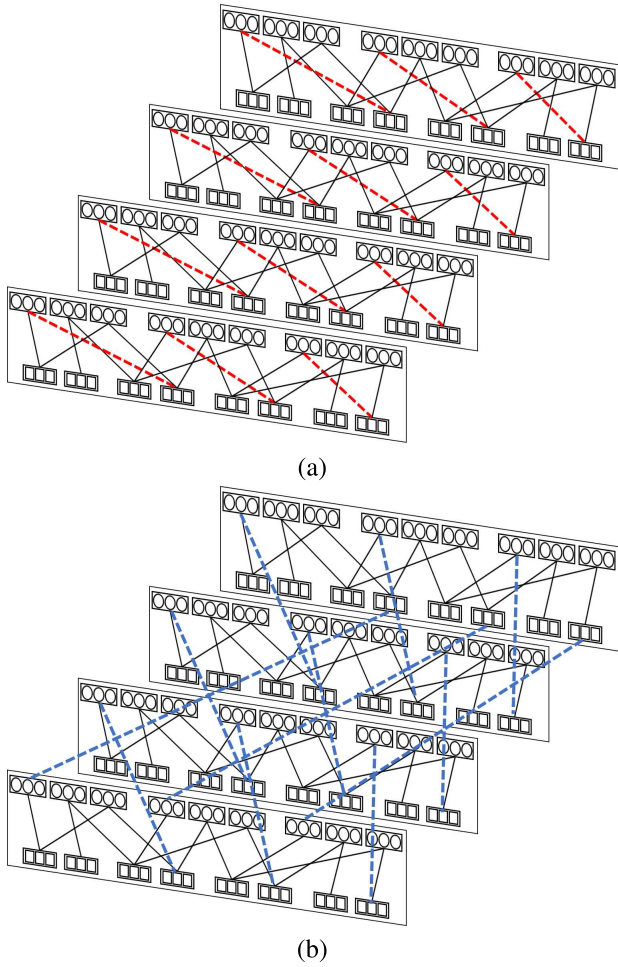


Fig. 2. (a) Four 1D-SC codes. Circles (resp., squares) represent VNs (resp., CNs). Each line represents a group of connections (defined by a CPM) from z VNs to z CNs. Problematic connections are shown in dashed red lines. (b) MD-SC code with $T = 1$, $d = 2$, and $L_2 = 4$. Rewired connections are shown in dashed blue lines.

IV. NOVEL 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 CPMs on the population of cycles. Then, we present our algorithm for constructing MD-SC codes which is based on a score voting policy.

A. The Effects of Relocation of CPMs on Cycles

Consider a cycle $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}$ with the sequence of CPMs $\mathcal{C}_{\mathcal{O}_k}$. 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 CPMs of $\mathcal{O}_k$, and we call this subset *targeted CPMs*. We show that, after relocations, L_2 instances of CPMs of $\mathcal{C}_{\mathcal{O}_k}$ can form L_2 cycles of length k , $L_2/2$ cycles of length $2k$, ..., or one cycle of length L_2k . 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 1. Let $\mathcal{C}_{\mathcal{O}_k} = \{\mathcal{C}_{i_1,j_1}, \mathcal{C}_{i_2,j_2}, \dots, \mathcal{C}_{i_k,j_k}\}$ be the sequence of CPMs in $\mathbf{H}_{\text{SC}}$ that are visited in a clock-

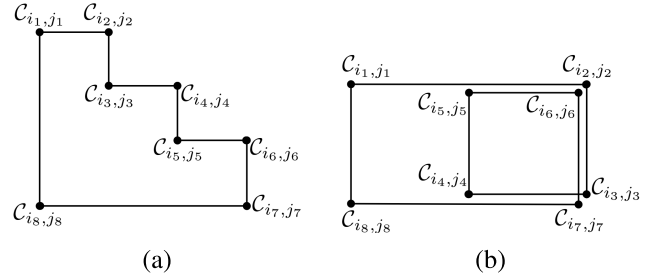


Fig. 3. Cycles-8 with $\mathcal{C}_{\mathcal{O}_8} = \{\mathcal{C}_{i_1,j_1}, \dots, \mathcal{C}_{i_8,j_8}\}$. Each line represents a connection between two CPMs. (a) All CPMs are unique. (b) $\mathcal{C}_{i_6,j_6} = \mathcal{C}_{i_2,j_2}$ and $\mathcal{C}_{i_7,j_7} = \mathcal{C}_{i_3,j_3}$.

wise order by $\mathcal{O}_k$. If the following equation holds, the L_2 instances of CPMs of $\mathcal{C}_{\mathcal{O}_k}$ form L_2 cycles- k in $\mathbf{H}_{\text{SC}}^{\text{MD}}$,

$$\sum_{u=1}^k (-1)^u M(\mathcal{C}_{i_u,j_u}) \stackrel{L_2}{=} 0. \quad (4)$$

Otherwise, the instances of the targeted CPMs do not result in cycles- k in $\mathbf{H}_{\text{SC}}^{\text{MD}}$.³ We call (4) the *Ineffective Relocation Condition*, or *IRC*, in the rest of this paper.

Proof: Let $(\mathcal{C}_{i_u,j_u}, \mathcal{C}_{i_{u+1},j_{u+1}})$ be a pair of consecutive CPMs in $\mathcal{C}_{\mathcal{O}_k}$, where $u \in \{1, \dots, k\}$ and $\mathcal{C}_{i_{k+1},j_{k+1}} = \mathcal{C}_{i_1,j_1}$. By definition, two CPMs have the same row (resp., column) group index, i.e., $i_u = i_{u+1}$ (resp., $j_u = j_{u+1}$), when $u \stackrel{2}{=} 1$ (resp., $u \stackrel{2}{=} 0$). 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 $\mathcal{C}_{i_u,j_u} \in \mathcal{C}_{\mathcal{O}_k}$, $a, b \in \{0, \dots, L_2 - 1\}$, 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 CPMs in $\mathcal{C}_{\mathcal{O}_k}$ do not all belong to the same segment.

Here, a unit of an 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 CPMs of the cycle in a clock wise order (with the same order they appear in $\mathcal{C}_{\mathcal{O}_k}$), 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 $\{\mathbf{H}'_{\text{SC}}, \mathbf{A}_{L_2-1}, \dots, \mathbf{A}_1\}$, with the MD mapping $\{0, L_2 - 1, \dots, 1\}$, from left to right. These segments appear in the cyclic order $\{\mathbf{H}'_{\text{SC}}, \mathbf{A}_1, \dots, \mathbf{A}_{L_2-1}\}$, with the MD mapping $\{0, 1, \dots, L_2 - 1\}$, from top to bottom, see (3). Thus, the MD horizontal shift, when we go from $\mathcal{C}_{i_u,j_u}$ to $\mathcal{C}_{i_{u+1},j_{u+1}}$, $u \in \{1, 3, \dots, k-1\}$, is $(M(\mathcal{C}_{i_u,j_u}) - M(\mathcal{C}_{i_{u+1},j_{u+1}}))_{L_2}$ units, see Definition 1.3. Similarly, the MD vertical shift, when we go from $\mathcal{C}_{i_u,j_u}$ to $\mathcal{C}_{i_{u+1},j_{u+1}}$, $u \in \{0, 2, \dots, k\}$, 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 $(\cdot)_p$ defines modulo p of an integer. The total MD horizontal and vertical shifts when we traverse the CPMs of

³Equation (4) resembles Fosserier's condition on CPM powers of a circulant-permutation-based code that makes a cycle in the protograph result in multiple cycles in the lifted graph of the code [29].

$\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$ are δ_H and δ_V , respectively:

$$\begin{aligned}\delta_H &= \left(\sum_{u \in \{1, 3, \dots, k-1\}} [M(\mathcal{C}_{i_u, j_u}) - M(\mathcal{C}_{i_{u+1}, j_{u+1}})] \right)_{L_2} \\ &= \left(- \sum_{u=1}^k [(-1)^u M(\mathcal{C}_{i_u, j_u})] \right)_{L_2}, \\ \delta_V &= \left(\sum_{u \in \{2, 4, \dots, k\}} [M(\mathcal{C}_{i_{u+1}, j_{u+1}}) - M(\mathcal{C}_{i_u, j_u})] \right)_{L_2} \\ &= \left(- \sum_{u=1}^k [(-1)^u M(\mathcal{C}_{i_u, j_u})] \right)_{L_2}.\end{aligned}\quad (5)$$

The relocations are ineffective if and only if the start and end segments are the same when we traverse the k CPMs of $\mathcal{O}_k$. For this to happen, the total MD horizontal and vertical shifts (δ_H and δ_V) need to be zero, which results in (4). $\square$

If equation (4), or IRC, holds for the CPMs of $\mathcal{O}_k$, L_2 instances of CPMs of $\mathcal{C}_{\mathcal{O}_k}$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$ form L_2 cycles- k in the MD-SC code (not preferable). Theorem 2 investigates the situation when IRC does not necessarily hold.

Theorem 2. Each cycle $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$ results in τ cycles with length $L_2 k / \tau$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$, where

$$\tau = \gcd(L_2, \Delta_{\mathcal{O}_k}), \quad (6)$$

and

$$\Delta_{\mathcal{O}_k} = \left(- \sum_{u=1}^k [(-1)^u M(\mathcal{C}_{i_u, j_u})] \right)_{L_2}. \quad (7)$$

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

Proof: Consider a cycle $\mathcal{O}_k$ with $\mathcal{C}_{\mathcal{O}_k} = \{\mathcal{C}_{i_1, j_1}, \dots, \mathcal{C}_{i_k, j_k}\}$ in $\mathbf{H}_{\text{SC}}$. There are $(L_2)^2$ instances of $\mathcal{C}_{i_u, j_u}$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$, $u \in \{1, \dots, k\}$, one per each segment, and only L_2 of them can be non-zero. We traverse the CPMs of $\mathcal{O}_k$ in $\mathbf{H}_{\text{SC}}^{\text{MD}}$ according to the order they appear in $\mathcal{C}_{\mathcal{O}_k}$ starting from a non-zero instance of $\mathcal{C}_{i_1, j_1}$. After traversing all k CPMs, we reach $\mathcal{C}_{i_1, j_1}$ 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 CPMs. In this case, there are L_2 instances of $\mathcal{C}_{\mathcal{O}_k}$, one per each instance of $\mathcal{C}_{i_1, j_1}$. If $\Delta_{\mathcal{O}_k} \neq 0$, the cycle cannot be complete after traversing k CPMs. We proceed traversing the CPMs until we reach $\mathcal{C}_{i_1, j_1}$ that is in the same segment that we started from.

We define the parameter λ as follows:

$$\lambda = \min\{g | g \in \{1, 2, \dots\}, g\Delta_{\mathcal{O}_k} \stackrel{L_2}{=} 0\}. \quad (8)$$

Then, we complete the cycle after traversing λk CPMs. The parameter λ is the minimum integer value such that $\lambda\Delta_{\mathcal{O}_k} \stackrel{L_2}{=} 0$, i.e., $\lambda = L_2 / \gcd(L_2, \Delta_{\mathcal{O}_k})$. The L_2 instances of the k CPMs in $\mathcal{C}_{\mathcal{O}_k}$ form $\tau = L_2 k / \lambda k = \gcd(L_2, \Delta_{\mathcal{O}_k})$ cycles of the length $\lambda k = L_2 k / \tau$. $\square$

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 instances of the CPMs of $\mathcal{C}_{\mathcal{O}_k}$. When $\tau = \gcd(L_2, \Delta_{\mathcal{O}_k}) = L_2$, the instances of the CPMs of $\mathcal{C}_{\mathcal{O}_k}$ form L_2 cycles with length k . In our

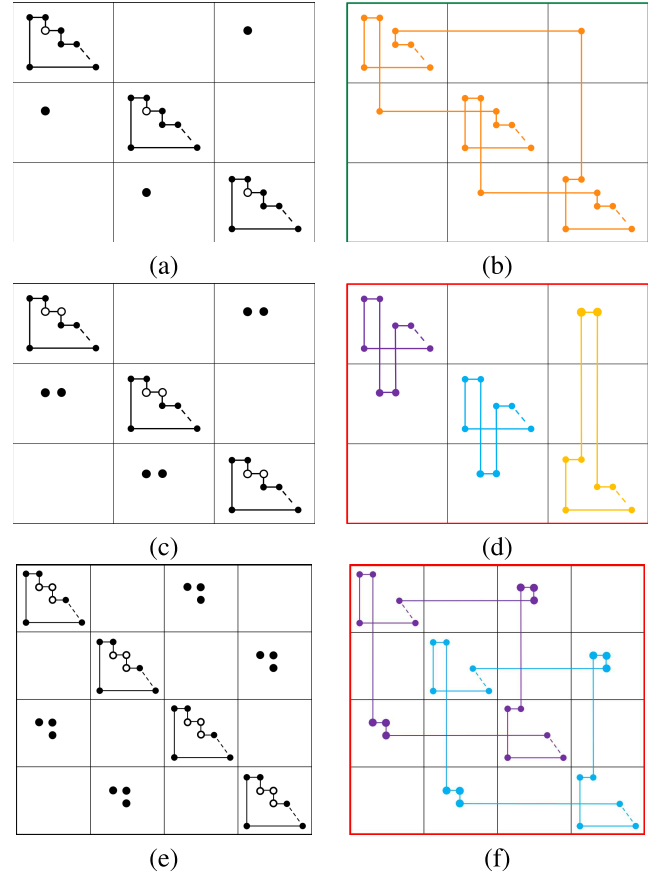


Fig. 4. (a) $\mathcal{C}_{i_a, j_a} \rightarrow \mathbf{A}_1$. The white circles show original locations of the relocated CPM. (b) A cycle-3k is formed. (c) $\{\mathcal{C}_{i_a, j_a}, \mathcal{C}_{i_b, j_b}\} \rightarrow \mathbf{A}_1$. (d) Three cycles- k are formed. (e) $\{\mathcal{C}_{i_a, j_a}, \mathcal{C}_{i_b, j_b}, \mathcal{C}_{i_c, j_c}\} \rightarrow \mathbf{A}_2$. (f) Two cycles-2k are formed.

algorithm for the MD-SC code construction, the relocations that result in smaller τ are more preferred as they result in larger cycles.

Remark 1. Review some properties of $\gcd$ that are used in the rest of this paper:

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

Example 2. Let $\mathcal{C}_{\mathcal{O}_k} = \{\mathcal{C}_{i_1, j_1}, \dots, \mathcal{C}_{i_k, j_k}\}$ be the sequence of CPMs of $\mathcal{O}_k$, and n be the number of its relocated CPMs.

- 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 $\tau = 1$. Fig. 4(a) shows $\mathcal{C}_{i_a, j_a} \rightarrow \mathbf{A}_1$. Fig. 4(b) shows that a cycle-3k (shown in orange) is formed. The green border represents that this relocation is preferable.
- 2) Let $n = 2$, $\mathcal{C}_{i_a, j_a}, \mathcal{C}_{i_b, j_b} \rightarrow \mathbf{A}_1$, and $L_2 = 3$. Suppose $D_{\mathcal{O}_k}(\mathcal{C}_{i_a, j_a}, \mathcal{C}_{i_b, j_b}) = 1$. Then, $\Delta_{\mathcal{O}_k} = ((-1)^a - (-1)^b)_3 = 0$ and $\tau = L_2 = 3$. Fig. 4(c) shows $\mathcal{C}_{i_a, j_a}, \mathcal{C}_{i_b, j_b} \rightarrow \mathbf{A}_1$. Fig. 4(d) shows that three cycles- k are formed. The red border represents that these relocations are not preferable.
- 3) Let $n = 3$, and $\mathcal{C}_{i_a, j_a}, \mathcal{C}_{i_b, j_b}, \mathcal{C}_{i_c, j_c} \rightarrow \mathbf{A}_2$, and $L_2 = 4$. Suppose these three CPMs are consecutive in $\mathcal{C}_{\mathcal{O}_k}$. Then, $\Delta_{\mathcal{O}_k} = ((-1)^a(2 - 2 + 2))_4 = 2$ and $\tau = 2$.

Fig. 4(e) shows $C_{i_a, j_a}, C_{i_b, j_b}, C_{i_c, j_c} \rightarrow \mathbf{A}_2$. Fig. 4(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 CPMs to $\mathbf{A}_1$ instead, the result would be one cycle- $4k$ which is more preferred.

Remark 2. A CPM can appear more than once in C_{O_k} , e.g., see Fig. 3(b). A CPM that is repeated r times in the sequence can be interpreted in our analysis as r different CPMs; every two CPMs from this group have an even distance on O_k . The relocation of a CPM that appears r times is equivalent to the relocation of r CPMs to the same auxiliary matrix.

B. 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 III, the MD coupling with depth d is performed via relocating problematic CPMs to auxiliary matrices $\mathbf{A}_t$, $t \in \{1, \dots, d-1\}$. After relocating one CPM, the ranking of the problematic CPMs (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 IV (A). A tree-based strategy has also been recently applied to find a good partitioning to construct 1D-SC codes with a reduced population of problematic objects [30].

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

$$R(O_k, t) = \frac{L_2}{\gcd(L_2, \Delta_{O_k})},$$

$$\Delta_{O_k} = ((-1)^{v+1}rt - \sum_{C_{i_u, j_u} \in C_{O_k} \setminus C_{i_v, j_v}} [(-1)^u M(C_{i_u, j_u})])_{L_2}. \quad (9)$$

Here, we assumed C_{i_v, j_v} is repeated r times in C_{O_k} , and v is the index of one of the repetitions.

In fact, there might be several options for a targeted CPM such that IRC (i.e., (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 3. Consider the cycle O_k and a target CPM $C_{i_v, j_v} \in C_{O_k}$.

Algorithm 1 Score Voting Algorithm for Relocation

```

1: inputs: targeted CPM  $C_{i_v, j_v}$ ,  $k$ ,  $[M(C_{i, j})]$ ,  $d$ , and  $L_2$ .
2: Find  $\Psi$ , the set of all active/inactive cycles- $k$  that have  $C_{i_v, j_v}$  in their sequences.
3: for each  $O_k \in \Psi$  do
4:   for  $t \leftarrow 0$  to  $d-1$  do
5:      $M(C_{i_v, j_v}) = t$ .
6:      $\Delta_{O_k} = (-\sum_{u=1}^k [(-1)^u M(C_{i_u, j_u})])_{L_2}$ .
7:      $R(O_k, t) = L_2 / \gcd(L_2, \Delta_{O_k})$ .
8:    $\Phi = \{0, \dots, d-1\}$ .
9:   for  $x \leftarrow 1$  to  $\lfloor L_2/2 \rfloor$  do
10:    if  $L_2 \stackrel{x}{=} 0$  then
11:       $\Phi \leftarrow \arg \min_{t \in \Phi} |\{O_k | O_k \in \Psi, R(O_k, t) = x\}|$ .
12: output: relocation options  $\Phi$ .
```

Scenario 1. No CPMs of O_k are previously relocated, and C_{i_v, j_v} appears once in C_{O_k} (i.e., $r = 1$). Thus, IRC does not hold after a relocation, regardless of the auxiliary matrix that C_{i_v, j_v} is relocated to. For the option $M(C_{i_v, j_v}) = t$, $R(O_k, t) = L_2 / \gcd(L_2, t)$. For instance, O_k gives score 1 to the option “keep in $\mathbf{H}'_{SC}$ ”, and gives score L_2 to the option “relocate to $\mathbf{A}_1$ ”.

Scenario 2. $C_{i_w, j_w} \in C_{O_k}$ is already relocated to $\mathbf{A}_1$, $D_{O_k}(C_{i_v, j_v}, C_{i_w, j_w}) = 2$, and both CPMs appear once in C_{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 $M(C_{i_u, j_u}) = t$, $t \in \{0, \dots, d-1\}$, O_k gives score $R(O_k, t) = L_2 / \gcd(L_2, t+1)$. For instance, O_k gives score 1 to “relocate to $\mathbf{A}_{L_2-1}$ ”, and gives score L_2 to “keep in $\mathbf{H}'_{SC}$ ” and “relocate to $\mathbf{A}_{t'}$ ” where $(t'+1)$ and L_2 are relatively prime.

The relocation options are {relocate to $\mathbf{A}_1, \dots$, relocate to $\mathbf{A}_{d-1}$, keep in $\mathbf{H}'_{SC}$ }. We identify the best options for a targeted CPM as follows: 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 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 CPM. Then, all survived options are recorded as branches of a tree, and the next targeted CPM is chosen and similarly evaluated for each branch.

Remark 3. The score value is by definition a divisor of L_2 . Thus, x is only considered for the above analysis if $L_2 \stackrel{x}{=} 0$. 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 , $x \in \{x | x \in \{1, 2, \dots, L_2/2\}, L_2 \stackrel{x}{=} 0\}$, 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 $\mathbf{H}_{SC}$ that visit CPMs 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 CPM C_{i_v, j_v} is chosen from $\{C_{i, j} | C_{i, j} \neq \mathbf{0} \text{ and } M(C_{i, j}) = 0\}$ to increase the MD

coupling. (2) The most problematic CPM is the one that is visited by the most active cycles. (3) Each active/inactive cycle that visits C_{i_v, j_v} (has 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 $t \in \{1, \dots, L_2 - 1\}$. Other nodes correspond to one relocation each, and the path from the root node to a node at level ℓ , $\ell \in \{1, \dots, \mathcal{T}\}$, 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 $\ell - 1$, individually. For each node at level $\ell - 1$, 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 CPMs 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 CPM 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 CPM in the sorted list is targeted. We continue this process until the most problematic CPM, 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 CPM 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 also trimmed.

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 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 4. Fig. 5 illustrates an example for the tree of solutions to construct an MD-SC code with parameters $L_2 = 3$, $d = 3$, and $\mathcal{T} = 5$.⁴ At iteration 1, there are two winning relocation options for the targeted CPM, 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

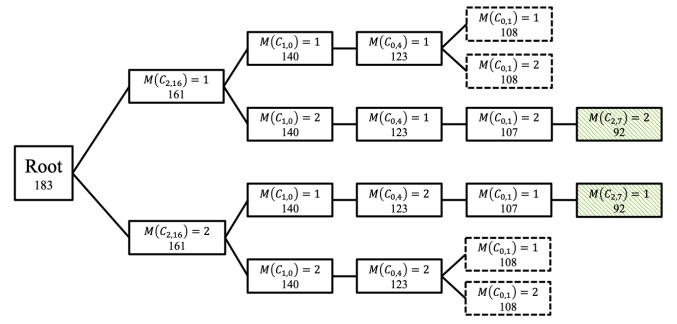


Fig. 5. An illustration for a tree of solutions. The information associated with each node are the relocation option and the number of cycles-6 for the solution described by the path from the root up to this node. The nodes with dashed borders show the trimmed solutions. The nodes with hatch background show the winning solutions.

4 solutions result in 123 active cycles-6. At iteration 4, the 1st and 4th nodes at level 3 are expanded to two nodes each, and the 2nd and 3rd 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 nodes that have hatch backgrounds) result in 92 active cycles-6, and one of them can be chosen randomly for constructing the MD-SC code.

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

V. LOW-LATENCY DECODING OF MD-SC CODES

In this section, we implement and analyze a low-latency windowed decoding for MD-SC codes. First, we describe the decoding method. Then, we provide the latency analysis of our decoder.

A. Multi-Dimensional Windowed Decoding

In this subsection, we describe a low-latency windowed decoder for our MD-SC codes. Our decoder extends the well-studied windowed decoder of the conventional SC codes, [31], [32], to allow for low-latency decoding across multiple constituent SC codes. Such a decoder was briefly introduced in [14]. In our paper, we thoroughly define and analyze the multi-dimensional windowed decoder which, to our knowledge, has not been done before.

First, note that each segment of an MD-SC code has the staircase structure of an SC code, see Definition 1.2. Therefore, if two VNs do not to share CNs within one constituent SC code before MD coupling, any instance of these two VNs across different constituent SC codes also do not share CNs after MD coupling. This observation motivates the low-latency windowed decoding, similar to 1D-SC codes.

We define an MD window as a collection of several smaller (local) windows that are each defined over one segment of $\mathbf{H}_{\text{SC}}^{\text{MD}}$. Let W_D be the size of the local windows and l_D be the MD window index, where $m + 1 \leq W_D \leq L$ and $1 \leq l_D \leq L$. Let $\mathbf{R}_k @ \mathbf{S}_{a,b}$ refer to the k^{th} replica in segment $\mathbf{S}_{a,b}$ of $\mathbf{H}_{\text{SC}}^{\text{MD}}$. Recall that $\mathbf{S}_{(a+t)L_2, a} = \mathbf{0}$ for $t \in \{d, \dots, L_2 - 1\}$

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

Algorithm 2 Algorithm for Constructing MD-SC Codes

```

1: inputs:  $\mathbf{H}_{\text{SC}}, k, L_2, d$ , and  $\mathcal{T}$ .
2: initialize: A tree with one node (root),  $\ell = 1$ .
3: Find  $\Gamma$ , i.e., the set of all cycles- $k$  in  $\mathbf{H}_{\text{SC}}$  that visit CPMs
   in the middle replica of  $\mathbf{H}_{\text{SC}}$ .
4: while  $\ell \leq \mathcal{T}$  and there are nodes at level  $\ell - 1$  do
5:   for each node at level  $\ell - 1$  do
6:     Set  $[M(C_{i,j})]$  according to the relocations suggested
       by the path from root to node.
7:     Find status (active/inactive) of cycles- $k$  in  $\Gamma$  using
       IRC or (4).
8:      $S = \{C_{i,j} | C_{i,j} \in \mathbf{R}_{\lceil L/2 \rceil} \text{ and } M(C_{i,j}) = 0\}$ .
9:     Sort  $S$  in a decreasing order according to the number
       of times they are visited by active cycles in  $\Gamma$ .
10:    Flag = 0.
11:    while  $|S| > 0$  and Flag = 0 do
12:      Select the first CPM  $C_{i_v, j_v}$  in  $S$  for relocation.
13:      Find best relocation options  $\Phi$  for  $C_{i_v, j_v}$  by Algo-
        rithm 1.
14:      if  $0 \in \Phi$  then  $S = S \setminus C_{i_v, j_v}$ 
15:      else
16:        Flag = 1.
17:        for  $\forall t \in \Phi$  do
18:          Add a child to node with content
             $M(C_{i_v, j_v}) = t$ .
19:    Count the number of active cycles for each solution
       suggested by the nodes at level  $\ell$ .
20:    Trim all leaves (and their parents if needed) that do not
       result in minimum active cycles- $k$ .
21:     $\ell = \ell + 1$ .
22: Pick a random solution, set  $M(C_{i,j})$  accordingly, and
       construct  $\mathbf{H}_{\text{SC}}^{\text{MD}}$  using (3).
23: output:  $\mathbf{H}_{\text{SC}}^{\text{MD}}$ .

```

and $a \in \{0, \dots, L_2 - 1\}$, which results in $\mathbf{R}_k$ being zero for these segments. Therefore, the local windows are only defined for the non-zero segments, and the number of local windows is $L_2 d$.

Consider the l_D^{th} MD window. For this MD window, we define a local window $W_{a,b}^{l_D}$, where $0 \leq a \leq L_2 - 1$ and $0 \leq b \leq d - 1$, as the edges between VNs and CNs that exclusively belong to replicas $\{\mathbf{R}_k @ \mathbf{S}_{(a+b)L_2, a}, l_D \leq k \leq \min(l_D + W_D - 1, L)\}$. As such, the l_D^{th} MD window is defined as the collection of local windows $W_{a,b}^{l_D}$, and we call it the *multi-dimensional window configuration*. We assume that the VNs corresponding to replicas $\{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{l_D-1}\}$ in all segments have already been decoded, and their decoded values contribute to the decoding of VNs in later MD windows. The l_D^{th} window performs decoding on its own MD window configuration and aims to decode the VNs corresponding to replica $\mathbf{R}_{l_D}$ of all segments, known as the *MD targeted VNs*. This operation is performed sequentially from the first to the L^{th} MD window until all the VNs are decoded. For example in Fig. 6(a), the small rectangles represent an MD window configuration. The green columns are VNs that have already been decoded, and the blue columns are the targeted VNs.

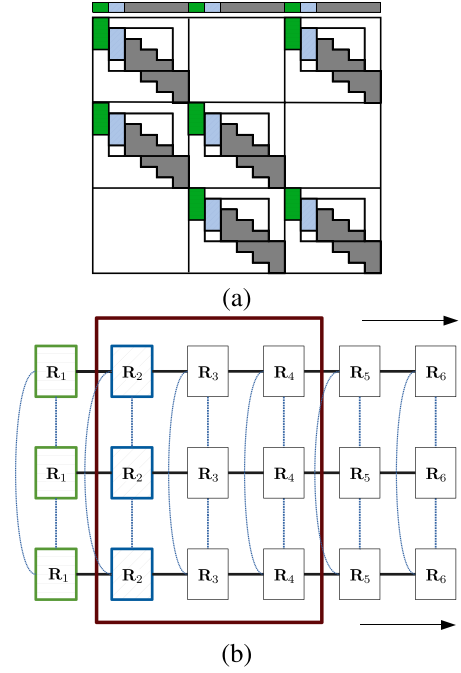


Fig. 6. In both figures, the color green with horizontal lines represents the decoded VNs and the color blue with diagonal lines represents the MD targeted VNs: (a) MD window configuration for an MD-SC code with parameters $L = 6, m = 2, L_2 = 3, d = 2, W_D = 4$, and $l_D = 2$. (b) Each smaller rectangle represents a replica and each horizontal chain is a constituent SC code. The bigger rectangle shows the MD window configuration.

One can view each constituent SC code as a chain of replicas. In the MD-SC code, constituent SC chains can only be connected together through their similar replicas. MD windowed decoding exploits this limited connectivity to allow for lower decoding latency, Fig. 6(b). The structure of our MD-SC codes allows for a simpler decoder implementation. For all MD window configurations, the graphs have the same structure (edge connectivity). Thus, the same, small decoder can be used for all MD windows and the only change across MD windows is the initial values of the VNs. This is another advantage of our deterministic construction compared to the random constructions, e.g., [11], [14], where the MD window configurations vary.

We now briefly mention a viable variant of the MD windowed decoding that is an interesting direction for future research. To further reduce the complexity and latency, one can limit the number of constituent SC codes that are considered in an MD window configuration to be less than L_2 . The major challenge with this decoder is that the degree distribution of the MD window can be very different from the global degree distribution, affecting the decoding threshold of the MD windows [33]. Since the MD window configurations depend on the selected relocations, the score voting algorithm would need to take into account the degree distribution change. This observation requires more analysis which is left as future work.

B. Latency Analysis

In this subsection, we provide a latency analysis of our MD windowed decoder.⁵ For decoding a group of VNs,

⁵The presented latency analysis is inspired by the analysis in [31] performed for the one-dimensional windowed decoding.

we consider the decoding latency as the time the first VN is acquired until the whole VNs in that group are decoded, which is an upper bound for the latency of all VNs in the group. First, we consider the latency of a block decoder. The block decoder requires all the VNs to start decoding. Therefore, its decoding latency T is given by $T = T_{rec} + T_{dec}$, where T_{rec} is the time needed to receive the full codeword and T_{dec} is the time needed to decode the codeword.

We define the window latency as the time needed to decode the targeted VNs for a single MD window. This latency gives the time elapsed between successive decoding of the MD targeted VNs.⁶ We can define window latency as $T^{W_D} = T_{rec}^{W_D} + T_{dec}^{W_D}$, where $T_{rec}^{W_D}$ is the time needed to receive the VNs in the MD window and $T_{dec}^{W_D}$ is the time needed to decode the targeted VNs. We can relate $T_{rec}^{W_D}$ and T_{rec} by

$$T_{rec}^{W_D} \leq \frac{(W_D + m)\kappa L_2 z}{L\kappa L_2 z} T_{rec} = \frac{(W_D + m)}{L} T_{rec}, \quad (10)$$

since all MD windows, except for a few trailing and leading ones, require $(W_D + m)\kappa L_2 z$ values for the VNs in their MD configurations before they can start decoding.

We assume that the number of iterations is the same for both the block decoder and the MD windowed decoder. For iterative decoding, the decoding time grows linearly with the number of VNs in consideration. Each MD configuration has $W_D \kappa L_2 z$ VNs, except for the last $(W_D - 1)$ MD windows that have fewer VNs. Therefore, $T_{dec}^{W_D}$ and T_{dec} are related by:

$$T_{dec}^{W_D} \leq \frac{W_D \kappa L_2 z}{L\kappa L_2 z} T_{dec} = \frac{W_D}{L} T_{dec}. \quad (11)$$

Using (10) and (11), $T^{W_D} \leq \frac{W_D + m}{L} T_{rec} + \frac{W_D}{L} T_{dec} \leq \frac{W_D + m}{L} T$. As expected, the latency reduction is similar to windowed decoding of 1D-SC codes, which shows that our MD-SC construction preserves the latency benefits of the 1D-SC codes.

VI. SIMULATION RESULTS

Our simulation results demonstrate the outstanding performance of our new framework for constructing MD-SC codes, and it is organized as follows: Subsections A and B are dedicated to the analysis of MD-SC codes with girths 6 and 8, respectively. In each subsection, we study the effect of parameters 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 Subsection C, we compare the performance of our well-designed MD-SC codes with random constructions. In Subsection D, we evaluate the performance of the MD windowed decoding. In our simulations, we consider the AWGN channel, and we use quantized min-sum algorithm with 4 bits and 15 iterations for the decoding.

⁶This definition of latency is commonly used in the literature on windowed decoding, e.g., [31], [32].

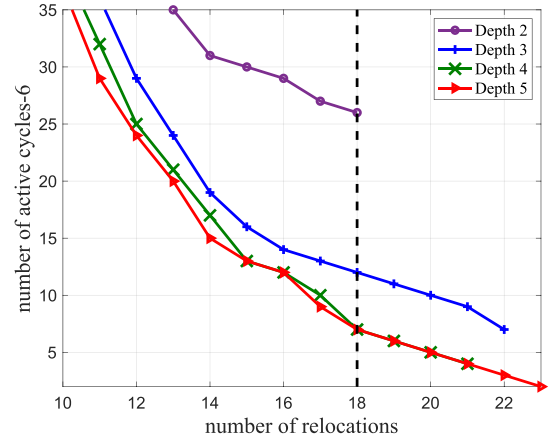


Fig. 7. The number of active cycles-6 for various densities and depths. After a point, increasing T does not decrease the population of active cycles-6, e.g., after 18 relocations for depth 2, which results an early termination.

TABLE I
POPULATION OF CYCLES-6 FOR MD-SC CODES WITH SC-CODE 1 AS THE CONSTITUENT SC CODE, $L_2 = 5$, AND DENSITY 26.47%

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

A. Analysis for MD-SC Codes With Girth 6

We first describe the code parameters of SC-Code 1 with girth 6 that is used as the constituent SC code in the rest of this subsection. SC-Code 1 has parameters $\kappa = z = 17$, $\gamma = 4$, $m = 1$, $L = 10$, rate 0.74, and length 2,890 bits, and it is constructed by the OO-CPO technique [7]. The partitioning and CPM powers of SC-Code 1 are given in Appendix. 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. Fig. 7 shows the effect of increasing the MD coupling density T on the population of cycles-6 for various MD coupling depths. The horizontal axis shows 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 CPMs of the middle replica of the constituent SC code and IRC (i.e., (4)) holds for it. As we see, increasing 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 I shows the number of cycles-6 for MD-SC codes with $L_2 = 5$, density 18 (26.47% of CPMs), and for various MD coupling depths. As we see, increasing the depth improves the cycle properties of the MD-SC codes. According to Table I, 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. Fig. 8 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 BER performance compared to MD-SC code with depth 2.

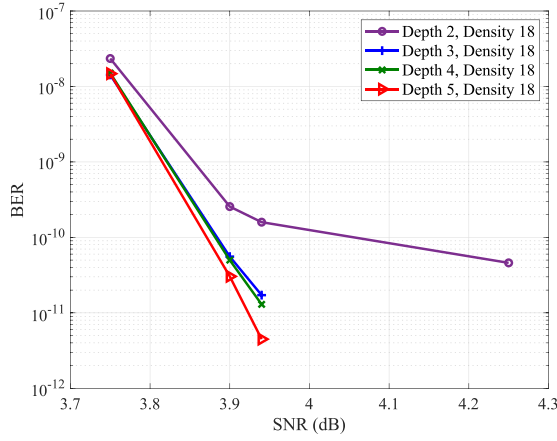


Fig. 8. The BER curves over AWGN channel at density 26.47% and for various depths.

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 1 has $L_2 = 1$, and it is, in fact, one instance of SC-Code 1. MD-SC-Code 2 has $L_2 = 3$, $d = 3$ (maximum depth), and $T = 23$ (maximum density). After reaching the maximum density, relocation does not decrease the population of the cycles of interest. SC-Code 2 is an SC code similar to SC-Code 1 but with $L = 30$ (three times the coupling length of SC-Code 1); thus it has comparable length and rate to MD-SC-Code 2. MD-SC-Code 3 has $L_2 = 5$, $d = 5$ (maximum depth), and $T = 23$ (maximum density). SC-Code 3 is an SC code similar to SC-Code 1 but with $L = 50$; thus it has comparable length and rate to MD-SC-Code 3. The MD mapping matrices, i.e., $\mathbf{M} = [M(\mathcal{C}_{i,j})]$, for MD-SC-Codes 2 and 3 are shown below:

$$\mathbf{M}^2 = \begin{bmatrix} 0 & 1 & 2 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

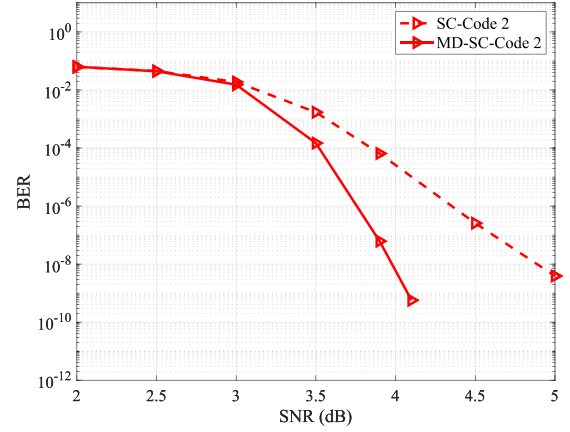
$$\mathbf{M}^3 = \begin{bmatrix} 0 & 3 & 3 & 0 & 2 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 3 & 0 & 0 & 0 & 0 & 3 & 0 & 3 & 4 & 0 & 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 3 & 4 & 3 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Table II shows the number of cycles-6 for SC-Codes 2 and 3 and MD-SC-Codes 1-3. MD-SC-Code 2 has nearly 90% fewer cycles-6 compared to SC-Code 2, and MD-SC-Code 3 has nearly 99% fewer cycles-6 compared to SC-Code 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.

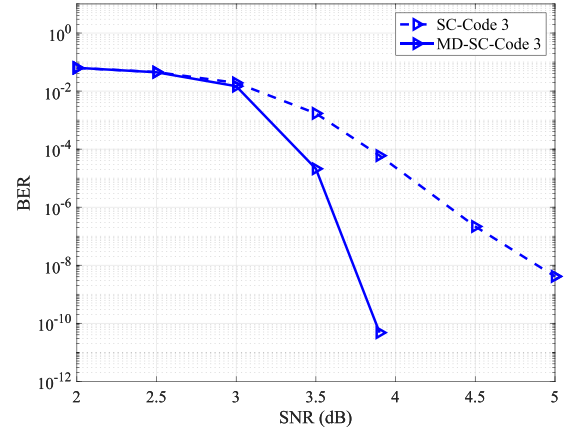
Fig. 9 compares the BER performance for our MD-SC codes and their 1D-SC counterparts. MD-SC-Code 2 shows about 4 orders of magnitude performance improvement compared to SC-Code 2 at SNR= 4.10 dB. This improvement is very pronounced for MD-SC-Code 3 compared to SC-Code 3 (about 6 orders of magnitude at SNR= 3.85 dB). These results demonstrate that the freedom offered by MD-SC codes is

TABLE II
POPULATION OF CYCLES-6 FOR MD-SC CODES AND THEIR 1D COUNTERPARTS

code name	L_2	length	rate	cycles-6
MD-SC-Code 1 (SC-Code 1)	1	2,890	0.74	29,274
SC-Code 2	1	8,670	0.76	91,494
MD-SC-Code 2	3	8,670	0.74	9,078
SC-Code 3	1	14,450	0.76	153,714
MD-SC-Code 3	5	14,450	0.74	1,700



(a)



(b)

Fig. 9. The BER curves over AWGN channel for MD-SC codes compared to their 1D counterparts: (a) $L_2 = 3$, (b) $L_2 = 5$.

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.

B. Analysis for MD-SC Codes With Girth 8

We first describe the code parameters of SC-Code 4 with girth 8 that is used as constituent SC code in the rest of this subsection. SC-Code 4 has parameters $\kappa = 19$, $z = 23$, $\gamma = 3$, $m = 2$, $L = 10$, rate 0.81, and length 4,370 bits, and it is constructed by the OO-CPO technique [7]. The partitioning

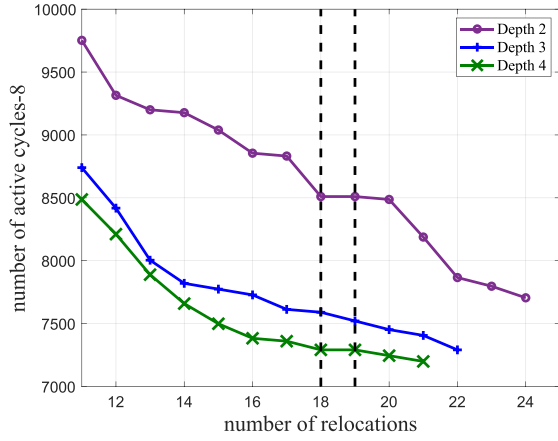


Fig. 10. The number of active cycles-8 for various densities and depths.

TABLE III
POPULATION OF CYCLES-8 FOR MD-SC CODES AND THEIR
1D COUNTERPARTS

code name	num. active cycles-8	total num. cycles-8
SC-Code 5	—	1,397,319
MD-SC-Code 4	8,510	292,560
MD-SC-Code 5	7,521	258,060
MD-SC-Code 6	7,291	249,320

and CPM powers of SC-Code 4 are given in Appendix. The cycles of interest here have length 8, i.e., $k = 8$.

We consider MD-SC codes with $L_2 = 4$ constructed by Algorithm 2. Fig. 10 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 18th and 19th in Fig. 10). 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 $2k = 16$.

Next, we study the BER performance of MD-SC codes with various depths and their 1D-SC counterpart. We first describe the codes: MD-SC-Codes 4-6 have $L_2 = 4$, $\mathcal{T} = 19$, SC-Code 4 as their constituent SC codes, length 17,480, and rate 0.81. MD-SC-Code 4, resp. 5 and 6, have depth 2, resp., 3 and 4. SC-Code 5 is an SC code similar to SC-Code 4 but with $L = 40$ (four times the coupling length of SC-Code 4); thus it has comparable length and rate to MD-SC-Codes 4-6 (length 17,480 and rate 0.83).⁷

According to Fig. 11, MD-SC-Codes 4-6 show about 2 orders of magnitude performance improvement compared to SC-Code 5 at SNR = 4.50 dB. Table III shows the number of cycles-8 for SC-Code 5 and MD-SC-Codes 4-6. MD-SC-Code 6 has nearly 82% fewer cycles-8 compared

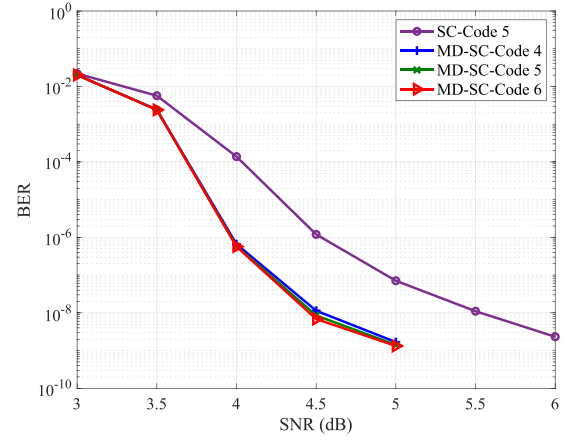


Fig. 11. The BER curves over AWGN channel at density 25% and for various depths along with the BER performance for the 1D-SC counterpart (SC-Code 5).

to SC-Code 5 and nearly 15% fewer cycles-8 compared to MD-SC-Code 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 error floor performance.

C. 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 [10], [11], [14], [15], 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 section have the same constituent SC code, i.e., SC-Code 6. SC-Code 6 has parameters $\kappa = 17$, $z = 17$, $\gamma = 3$, $m = 1$, $L = 15$, rate 0.81, and length 4,335 bits, and it is constructed by the OO-CPO technique [7]. The partitioning and CPM powers of SC-Code 6 are given in Appendix. The cycles of interest here have length 6, i.e., $k = 6$.

MD-SC-Codes 7-10 have $L_2 = 3$, $\mathcal{T} = 9$, SC-Code 6 as their constituent SC codes, length 13,005 bits, and rate 0.81. MD-SC-Codes 7 and 8, have depths 2 and 3, respectively, and they are constructed by Algorithm 2 introduced in this paper.⁸ MD-SC-Codes 9 and 10 are constructed by random relocations, and they both have depth 2. For MD-SC-Code 9, the relocated CPMs 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 10 is constructed in a similar way to MD-SC-Code 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.

⁷The MD mapping matrices for MD-SC-Codes 4-6 can be found in Appendix.

⁸The MD mapping matrices for MD-SC-Codes 7 and 8 can be found in Appendix.

TABLE IV
POPULATION OF SHORT CYCLES FOR MD-SC CODES CONSTRUCTED BY
VARIOUS POLICIES

code name	num. cycles-4	num. cycles-6	num. cycles-8
MD-SC-Code 7	0	2,856	685,032
MD-SC-Code 8	0	0	643,110
MD-SC-Code 9	255	9,010	585,820
MD-SC-Code 10	0	8,211	606,543

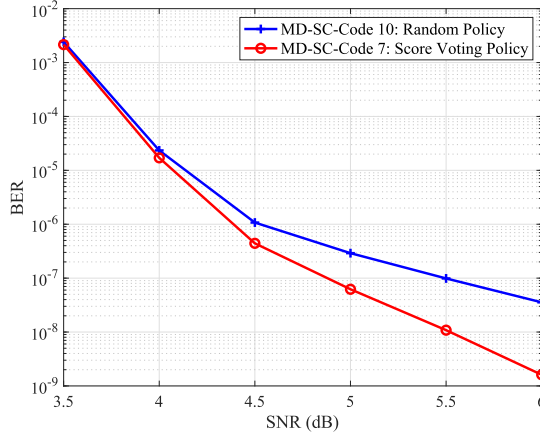


Fig. 12. The BER curves over AWGN channel for MD-SC codes with SC-Code 6 as the constituent SC code, $L_2 = 3$, density 18%, and constructed based on a random policy and our new score-voting policy.

Table IV shows the population of short cycles for MD-SC-Codes 7-10. As we see, MD-SC-Code 7 has 65% fewer cycles-6 compared to MD-SC-Code 10, and they both have zero cycles-4. These two codes have the same structure, but the relocated CPMs are chosen randomly for MD-SC-Code 10, while they are chosen to specifically reduce the number of cycles-6 for MD-SC-Code 7. MD-SC-Code 8, which is similar to MD-SC-Code 7 but with depth 3, has zero cycles-6 and 6.1% fewer cycles-8 compared to MD-SC-Code 7. MD-SC-Code 9 is similar to MD-SC-Code 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. Fig. 12 shows the BER performance comparison for MD-SC-Code 7 and MD-SC-Code 10. These two codes both have depth 2 and have the MD structure described in (3). At SNR = 6.0 dB, MD-SC-Code 7 shows nearly 1.3 orders of magnitude BER improvement compared to MD-SC-Code 10.

D. Evaluation of MD Windowed Decoding

In this subsection, we use SC-Code 1 as the constituent code and construct MD-SC-Code 11 with $L_2 = 5$, $d = 2$, $T = 18$, length 14,450 bits, and rate 0.74. We evaluate the BER performance of MD-SC-Code 11 using the MD windowed decoder with MD window sizes 3 and 4. As a reference, we also show the BER performance using a block decoder. Both the MD windowed decoder and the block decoder use min-sum algorithm with 15 iterations. The results are illustrated in Fig. 13. As expected, there is a slight degradation in the BER performance for windowed decoder

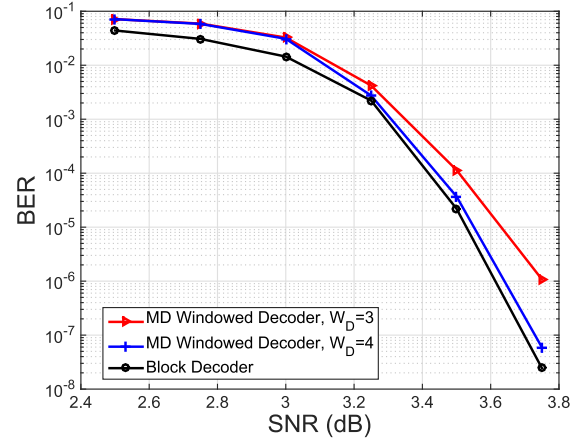


Fig. 13. The BER performance comparison for MD windowed decoder, with MD window sizes 3 and 4, and block decoder for decoding an MD-SC code.

compared to the block decoder. In addition, the degradation decreases as the MD window size increases, and it is already small for MD window size 4.

APPENDIX

The partitioning matrix $\mathbf{PM} = [h_{i,j}]$ and CPM power matrix $\mathbf{CM} = [f_{i,j}]$, with dimensions $\gamma \times \kappa$, describe partitioning and CPM powers, respectively. A CPM with row group index i and column group index j in the block code $\mathbf{H}$ is assigned to the component matrix $\mathbf{H}_{h_{i,j}}$, and it has power $f_{i,j}$. For SC-Codes 1-3, these two matrices are given below:

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

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

For SC-Codes 4-5, these two matrices are given below:

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

$$\mathbf{CM}^2 = \begin{bmatrix} 21 & 0 & 16 & 3 & 19 & 1 & 0 & 0 & 21 & 5 & 0 & 0 & 1 & 0 & 9 & 0 & 16 & 1 & 0 \\ 0 & 11 & 7 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \\ 0 & 17 & 0 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 & 22 & 1 & 3 & 5 & 19 & 9 & 11 & 13 \end{bmatrix}.$$

For SC-Code 6, these two matrices are given below:

$$\mathbf{PM}^3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix},$$

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

For MD-SC-Codes 4-8, the MD mapping matrices are given below:

$$\mathbf{M}^4 = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$

$$\mathbf{M}^5 = \begin{bmatrix} 0 & 2 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 2 & 0 & 0 \end{bmatrix},$$

$$\begin{aligned}
\mathbf{M}^6 &= \begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 3 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2 & 0 & 3 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 & 0 & 1 & 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 \end{bmatrix}, \\
\mathbf{M}^7 &= \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \\
\mathbf{M}^8 &= \begin{bmatrix} 2 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

VII. 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. The MD-SC codes can be classified as SC LDPC codes, but they indeed have specific structures beyond the traditional SC codes that are well-exploited in our design to improve the cycle properties. We presented a new code construction realized by jointly using traditional spatial coupling (in the constituent codes) and our new MD coupling (for connecting constituent codes). For MD coupling, we rewire connections (relocate CPMs) that are most problematic within each SC code. Our framework encompasses a systematic way to sequentially identify and relocate problematic CPMs, thus utilizing them to connect the constituent SC codes. Our MD-SC codes show a notable reduction in the population of small cycles and a significant improvement in the BER performance compared to the 1D setting. We also presented a windowed decoder for the MD-SC codes that exploits the locality of the constituent SC codes to attain a low decoding latency.

Two promising research directions are to investigate MD-SC codes on non-uniform channels, such as multilevel Flash and multi-dimensional magnetic recording channels, in addition to improve the presented windowed decoder by incorporating the MD coupling depth to further reduce the decoding latency and complexity. Furthermore, the presented methodology for constructing MD-SC codes can be extended to use circulant-based underlying block codes that have circulants of weight 0, 1, or larger than 1. This can possibly result in constructing irregular MD-SC codes, and it is an interesting research direction for future studies.

ACKNOWLEDGMENT

The authors would like to thank the reviewers whose suggestions helped improve and clarify this manuscript.

REFERENCES

- [1] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Multi-dimensional spatially-coupled code design through informed relocation of circulants," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2018, pp. 695–701.
- [2] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Multi-dimensional spatially-coupled code design with improved cycle properties," in *Proc. Annu. Non-Volatile Memories Workshop*, Mar. 2019, pp. 1–55.
- [3] A. J. Felstrom and K. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.
- [4] R. Tanner, D. Sridharan, A. Sridharan, T. Fuja, and D. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

- [5] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.
- [6] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.
- [7] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Finite-length construction of high performance spatially-coupled codes via optimized partitioning and lifting," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 3–16, Jan. 2019.
- [8] D. G. M. Mitchell and E. Rosnes, "Edge spreading design of high rate array-based SC-LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2940–2944.
- [9] A. Beemer and C. A. Kelley, "Avoiding trapping sets in SC-LDPC codes under windowed decoding," in *Proc. IEEE Int. Symp. Inf. Theory Appl.*, Oct. 2016, pp. 206–210.
- [10] D. Truhachev, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "New codes on graphs constructed by connecting spatially coupled chains," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2012, pp. 392–397.
- [11] R. Ohashi, K. Kasai, and K. Takeuchi, "Multi-dimensional spatially-coupled codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 2448–2452.
- [12] D. Truhachev, D. Mitchell, M. Lentmaier, and D. J. Costello, "Connecting spatially coupled LDPC code chains," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 2176–2180.
- [13] 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 Inf. Theory Workshop (ITW)*, Sep. 2013, pp. 1–5.
- [14] L. Schmalen and K. Mahdavian, "Laterally connected spatially coupled code chains for transmission over unstable parallel channels," in *Proc. 8th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Aug. 2014, pp. 77–81.
- [15] Y. Liu, Y. Li, and Y. Chi, "Spatially coupled LDPC codes constructed by parallelly connecting multiple chains," *IEEE Commun. Lett.*, vol. 19, no. 9, pp. 1472–1475, Sep. 2015.
- [16] R. Tanaka and K. Ishibashi, "Robust coded cooperation based on multi-dimensional spatially-coupled repeat-accumulate codes," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.
- [17] I. Ali, H. Lee, A. Hussain, and S.-H. Kim, "Protograph-based folded spatially coupled LDPC codes for burst erasure channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 516–519, Apr. 2019.
- [18] Y. Wang, S. C. Draper, and J. S. Yedidia, "Hierarchical and high-girth QC LDPC codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4553–4583, Jul. 2013.
- [19] 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 Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.
- [20] T. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2003, pp. 1426–1435.
- [21] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.
- [22] M. Karimi and A. H. Banihashemi, "On characterization of elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5188–5203, Sep. 2014.
- [23] Y. Hashemi and A. H. Banihashemi, "New characterization and efficient exhaustive search algorithm for leafless elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6713–6736, Dec. 2016.
- [24] R. Smarandache and P. O. Vontobel, "Quasi-cyclic LDPC codes: Influence of proto- and tanner-graph structure on minimum Hamming distance upper bounds," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 585–607, Feb. 2012.
- [25] M. Battaglioli, M. Baldi, and G. Cancellieri, "Connections between low-weight codewords and cycles in spatially coupled LDPC convolutional codes," *IEEE Trans. Commun.*, vol. 66, no. 8, pp. 3268–3280, Aug. 2018.
- [26] D. Mackay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
- [27] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Spatially coupled codes optimized for magnetic recording applications," *IEEE Trans. Magn.*, vol. 53, no. 2, pp. 1–11, Feb. 2017.

- [28] A. Hareedy, H. Esfahanizadeh, and L. Dolecek, "High performance non-binary spatially-coupled codes for flash memories," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2017, pp. 229–233.
- [29] M. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.
- [30] M. Battaglioni, F. Chiaraluce, M. Baldi, and D. Mitchell, "Efficient search and elimination of harmful objects in optimized QC SC-LDPC codes," 2019, *arXiv:1904.07158*. [Online]. Available: <https://arxiv.org/abs/1904.07158>
- [31] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.
- [32] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2277–2292, Apr. 2013.
- [33] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.



Homa Esfahanizadeh (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Tehran in 2012 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of California, Los Angeles (UCLA), Los Angeles, CA, USA. She currently works at the Laboratory for Robust Information Systems (LORIS), and her focus is on coding schemes for modern storage systems. Her research interests include coding and information

theory, signal processing, graph theory, and machine learning. She received the Electrical and Computer Engineering Department fellowship from UCLA in 2015 for her outstanding academic achievements. In 2018, she received the Memorable Paper Award at the Non-Volatile Memories Workshop (NVMW), in the area of devices, coding, and information theory. In 2018, she also received the prestigious 2018–2019 Dissertation Year Fellowship (DYF) at UCLA.



Lev Tautz (Student Member, IEEE) received the B.S. degree (Hons.) in electrical engineering and computer science from the University of California, Berkeley, CA, USA, in 2016. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of California, Los Angeles (UCLA), Los Angeles, CA. He currently works at the Laboratory for Robust Information Systems (LORIS), and he is focused on coding techniques for distributed storage and computation. His research interests include distributed systems, error-correcting codes, machine learning, and graph theory. He was a recipient of the Best Preliminary Exam in Signals and Systems Award in the Electrical and Computer Engineering Department, UCLA, in 2019.



Lara Dolecek (Senior Member, IEEE) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering and computer sciences, and the M.A. degree in statistics from the University of California, Berkeley. She is currently a Full Professor with the Electrical and Computer Engineering Department and the Mathematics Department (courtesy), University of California, Los Angeles (UCLA), Los Angeles, CA, USA. Prior to joining UCLA, she was a Post-Doctoral Researcher with the Laboratory for Information and Decision Systems, Massachusetts

Institute of Technology, Cambridge, MA, USA. She has served as a consultant for a number of companies specializing in data communications and storage. Her research interests span coding and information theory, graphical models, statistical methods, and algorithms, with applications to emerging systems for data storage and computing. She received the 2007 David J. Sakrison Memorial Prize for the most outstanding doctoral research in the Department of Electrical Engineering and Computer Sciences, UC Berkeley, CA. She received the IBM Faculty Award in 2014, the Northrop Grumman Excellence in Teaching Award in 2013, the Intel Early Career Faculty Award in 2013, the University of California Faculty Development Award in 2013, the Okawa Research Grant 2013, the NSF CAREER Award in 2012, and the Hellman Fellowship Award in 2011. With her research group and collaborators, she received numerous best paper awards. She currently serves as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY and as the Secretary of the IEEE Information Theory Society.