

A Channel-Aware Combinatorial Approach to Design High Performance Spatially-Coupled Codes

Ahmed Hareedy¹, Member, IEEE, Ruiyi Wu, Student Member, IEEE,
and Lara Dolecek², Senior Member, IEEE

Abstract—Because of their capacity-approaching performance and their complexity/latency advantages, spatially-coupled (SC) codes are among the most attractive error-correcting codes for use in modern dense data storage systems. SC codes are constructed by partitioning an underlying block code and coupling the partitioned components. Here, we focus on circulant-based SC codes. Recently, the optimal overlap (OO), circulant power optimizer (CPO) approach was introduced to construct high performance SC codes for additive white Gaussian noise (AWGN) and Flash channels. The OO stage operates on the protograph of the SC code to derive the optimal partitioning that minimizes the number of graphical objects that undermine the performance of SC codes under iterative decoding. Then, the CPO optimizes the circulant powers to further reduce this number. Since the nature of detrimental objects in the graph of a code critically depends on the characteristics of the channel of interest, extending the OO-CPO approach to construct SC codes for channels with intrinsic memory is not a straightforward task. In this paper, we tackle one relevant extension; we construct high performance SC codes for practical 1-D magnetic recording channels, i.e., partial-response (PR) channels. Via combinatorial techniques, we carefully build and solve the optimization problem of the OO partitioning, focusing on the objects of interest in the case of PR channels. Then, we customize the CPO to further reduce the number of these objects in the graph of the code. SC codes designed using the proposed OO-CPO approach for PR channels outperform prior state-of-the-art SC codes by up to around 3 orders of magnitude in frame error rate (FER) and 1.1 dB in signal-to-noise ratio (SNR). More intriguingly, our SC codes outperform structured block codes of the same length and rate by up to around 1.8 orders of magnitude in FER and 0.4 dB in SNR. The performance advantage of SC codes designed using the devised OO-CPO approach over block codes of the same parameters is not only pronounced in the error floor region, but also in the waterfall region.

Index Terms—LDPC codes, spatially-coupled codes, absorbing sets, optimal partitioning, optimized lifting, data storage, magnetic recording.

I. INTRODUCTION

AS other data storage systems, magnetic recording (MR) systems operate at very low frame error rate (FER) levels that are typically below 10^{-12} [1]–[5]. Consequently, to ensure high error-correction capability in such systems, binary [3], [4], [6] and non-binary (NB) [5], [7]–[10] graph-based codes are used. Under iterative quantized decoding, the objects that dominate the error floor region of low-density parity-check (LDPC) codes simulated in partial-response (PR) and additive white Gaussian noise (AWGN) systems are different in their combinatorial nature because of the detector-decoder looping and the intrinsic memory in PR systems [5]. In particular, the authors in [5] introduced balanced absorbing sets (BASs) to characterize the detrimental objects in the case of PR (1-D MR) channels. Moreover, the weight consistency matrix (WCM) framework was introduced to systematically remove any type of absorbing sets (ASs) from the graph of an NB-LDPC code [11], [12].

Spatially-coupled (SC) codes [13]–[15] are graph-based (LDPC) codes constructed by partitioning an underlying block code into components of the same size, then rewiring these components multiple times [16]. Literature works studying the asymptotic performance of SC codes include [15], [17], and [18]. In this work, the underlying block codes, and consequently our constructed finite-length SC codes, are circulant-based (CB) codes. SC codes offer not only complexity/latency gains (if windowed decoding [19] is used), but also an additional degree of freedom in the code design; this added flexibility is achieved via the partitioning of the parity-check matrix of the underlying block code. This observation makes SC codes attractive across a range of applications. Contiguous [16] and non-contiguous [20]–[22] partitioning schemes were introduced in the literature for various applications. Recently, the optimal overlap (OO), circulant power optimizer (CPO) approach was introduced to design SC codes with superior performance for AWGN [23] and practical asymmetric Flash [24] channels. The OO partitioning operates on the protograph to compute the optimal set of overlap parameters that characterizes the partitioning. The CPO operates on the unlabeled graph (all edge weights are set to 1's) to adjust the circulant powers. The objective is to minimize the number of instances of a common substructure that exists in several detrimental objects, i.e., a parent of multiple detrimental

Manuscript received December 14, 2018; revised November 28, 2019; accepted February 24, 2020. Date of publication March 11, 2020; date of current version July 14, 2020. This work was supported in part by the UCLA Dissertation Year Fellowship, in part by the NSF under Grant CCF-BSF 1718389, and in part by a grant from the ASTC-IDEMA. This article was presented in part at the 2018 IEEE Global Communications Conference (GLOBECOM). (Corresponding author: Ahmed Hareedy.)

Ahmed Hareedy was with the Department of Electrical and Computer Engineering, University of California, Los Angeles (UCLA), Los Angeles, CA 90095 USA. He is now with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: ahareedy@ucla.edu; ahmed.hareedy@duke.edu).

Ruiyi Wu and Lara Dolecek are with the Department of Electrical and Computer Engineering, University of California, Los Angeles (UCLA), Los Angeles, CA 90095 USA (e-mail: ruiyiwu@ucla.edu; dolecek@ee.ucla.edu).

Communicated by M. Lentmaier, Associate Editor for Coding Theory.

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/TIT.2020.2979981

0018-9448 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

absorbing-set children. If the SC code is binary, the unlabeled graph is the final graph. If the SC code is non-binary, the WCM framework [11], [12] is used to optimize the edge weights after applying the OO-CPO approach.

In this paper, we propose an approach based on tools from combinatorics, optimization, and graph theory, to construct high performance time-invariant SC codes for PR channels. Unlike the case of AWGN and Flash channels (see [23] and [24]), the common substructure, whose number of instances we seek to minimize, in the case of PR channels can appear in different ways in the protograph of the SC code, making the optimization problem considerably more challenging. For that reason, we introduce the concept of the *pattern*, which is a configuration in the protograph that can result in instances of the common substructure in the unlabeled graph of the SC code after lifting. We derive an optimization problem, in which we express the weighted sum of the counts (numbers of instances) of all patterns in terms of the overlap parameters. Then, we compute the optimal set of overlap parameters (OO) that minimizes this sum. Moreover, we propose the necessary modifications to the CPO algorithm presented in [23] and [24] to make it suitable for the common substructure in the case of PR channels.

We demonstrate the gains achieved by our OO-CPO (-WCM for NB-SC codes) approach through tables and performance plots that compare our codes not only with SC codes, but also with CB block codes of the same length and rate. The reduction achieved by the OO-CPO approach in the number of detrimental objects reaches 92% compared with the uncoupled setting and 72% compared with a prior state-of-the-art SC code design technique. Furthermore, the performance gain achieved by the OO-CPO approach reaches 3 orders of magnitude and 1.1 dB compared with the prior state-of-the-art. Most interestingly, the proposed SC codes outperform block codes of the same parameters, and the gain reaches 1.8 orders of magnitude and 0.4 dB. A code threshold gain of up to 0.25 dB is also achieved for our SC codes compared with block codes of the same parameters, highlighting that the performance advantage is there even in the early waterfall region.

The rest of the paper is organized as follows. Section II introduces the necessary preliminaries. Different patterns of the common substructure are discussed in Section III. The analysis of the optimization problem is presented in Section IV. The needed modifications over the baseline CPO are detailed in Section V. We present our experimental results in Section VI. Finally, the work is concluded in Section VII.

II. PRELIMINARIES AND MOTIVATION

In this section, we review the construction of SC codes, provide a motivating example for SC codes simulated in a PR system, and present the definitions of the objects of interest. Here, each row (resp., column) in a parity-check matrix corresponds to a check node (CN) (resp., variable node (VN)) in the equivalent graph of the matrix (the graph of the code). Additionally, each non-zero entry in a parity-check matrix corresponds to an edge in the equivalent graph of the matrix. We focus on circulant-based (protograph-based) codes.

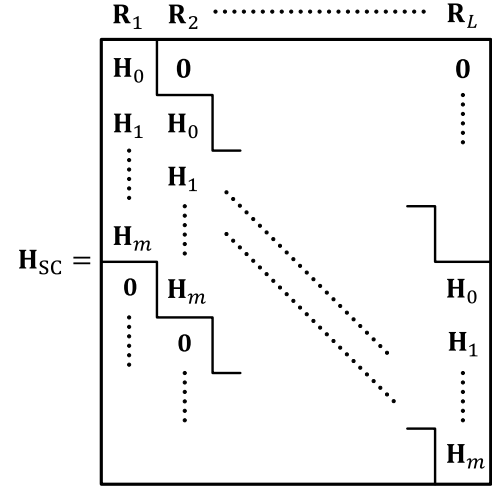


Fig. 1. The parity-check matrix of an SC code with parameters m and L . Replicas are also illustrated.

Since the contribution of this work (the OO-CPO) is to optimize the topology of the underlying graph, we will focus on the unlabeled graphs and binary matrices. Labeled graphs and non-binary matrices will be discussed as needed. Let $\mathbf{M}$ be a binary matrix, and let the protograph matrix (PM) of $\mathbf{M}$ be $\mathbf{M}^p$. Given $\mathbf{M}^p$, the matrix $\mathbf{M}$ is constructed through a process named *lifting*; that is, each 1 in $\mathbf{M}^p$ is replaced by a $z \times z$ non-zero circulant, and each 0 is replaced by a $z \times z$ zero circulant. Let σ be the $z \times z$ identity matrix after cyclically shifting its columns one unit to the left. Each non-zero circulant in $\mathbf{M}$ is σ raised to some power in $\{0, 1, \dots, z-1\}$ representing the number of units with which its columns are cyclically shifted to the left. The circulant size, z , and powers are defined as the lifting parameters.

Let $\mathbf{H}$ be the binary parity-check matrix of the underlying regular CB code. The matrix $\mathbf{H}$ is constructed from its PM $\mathbf{H}^p$ through lifting. The matrix $\mathbf{H}$ (and also $\mathbf{H}^p$) has column weight (VN degree) γ and row weight (CN degree) κ , and it consists of $\gamma\kappa$ non-zero circulants in our design, i.e., $\mathbf{H}^p$ is all 1's. The circulant size is z , and in our design we use $z \geq \kappa$. Each circulant in $\mathbf{H}$ is of the form $\sigma^{f_{i,j}}$, where $0 \leq i \leq \gamma-1$, $0 \leq j \leq \kappa-1$. Circulant powers are $f_{i,j}$, $\forall i, j$. Separable CB (SCB) codes have $f_{i,j} = f(i)f(j)$.

The binary SC code is constructed as follows. First, $\mathbf{H}$ is partitioned into $(m+1)$ disjoint component matrices (they all have the same size as $\mathbf{H}$): $\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_m$, where m is defined as the memory of the SC code. Each component matrix $\mathbf{H}_y$, $0 \leq y \leq m$, contains some of the $\gamma\kappa$ circulants of $\mathbf{H}$ and zero circulants elsewhere such that $\mathbf{H} = \sum_{y=0}^m \mathbf{H}_y$. Then, $\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_m$ are coupled L times, as shown in Fig. 1, to construct the binary parity-check matrix of the SC code, $\mathbf{H}_{SC}$, which is of size $\gamma z(L+m) \times \kappa z L$. In the OO-CPO approach, partitioning is determined on the protograph level. Our approach is general; it works for any m and any $\gamma \geq 3$. A *replica* is any $\gamma z(L+m) \times \kappa z$ submatrix of $\mathbf{H}_{SC}$ that contains $[\mathbf{H}_0^T \mathbf{H}_1^T \dots \mathbf{H}_m^T]^T$ and zero circulants elsewhere. Replicas are denoted by $\mathbf{R}_\rho$, $1 \leq \rho \leq L$ (see Fig. 1).

The PM of a component matrix $\mathbf{H}_y$, $0 \leq y \leq m$, is $\mathbf{H}_y^p$, and we call it a protograph component matrix. The PMs of $\mathbf{H}$ and its component matrices are all of size $\gamma \times \kappa$. The PM of $\mathbf{H}_{SC}$

is $\mathbf{H}_{\text{SC}}^{\text{P}}$, and it is of size $\gamma(L+m) \times \kappa L$. This $\mathbf{H}_{\text{SC}}^{\text{P}}$ also has L replicas, $\mathbf{R}_{\rho}$, $1 \leq \rho \leq L$, but with 1×1 circulants. Non-binary SC (NB-SC) codes can be constructed from binary SC codes as described in [24] and guided by [12]. Finite-length NB-SC codes are also discussed in [25]. The NB codes we use have parity-check matrices with their elements in $\text{GF}(q)$, where GF refers to Galois field, $q = 2^{\lambda}$ is the GF size (order), and $\lambda \in \{2, 3, \dots\}$ (in the binary case, $q = 2$). The SC code has block length $= \kappa z L \log_2(q)$ bits and rate $\approx [1 - \gamma(L+m)/(\kappa L)]$.

A partitioning is contiguous if the non-zero circulants in any component matrix $\mathbf{H}_y$, $0 \leq y \leq m$, are contiguous; otherwise, the partitioning is non-contiguous. A technique for contiguously partitioning $\mathbf{H}$ to construct $\mathbf{H}_{\text{SC}}$, namely cutting vector (CV) partitioning, was investigated aiming to generate SC codes for PR channels [16]. Several non-contiguous partitioning techniques were recently introduced in the literature, e.g., minimum overlap (MO) partitioning [20], [21], general edge spreading [22], in addition to OO partitioning [23], [24]. These non-contiguous partitioning techniques significantly outperform contiguous ones [20], [23], [24]. However, as far as we know, no prior work has proposed non-contiguous techniques in the context of PR channels. The goal of this work is to derive the effective OO-CPO approach for partitioning and lifting to construct high performance SC codes optimized for PR channels.

Consider the graph of an LDPC code. An (a, b) AS in this graph is defined as a set of a VNs with b unsatisfied neighboring CNs such that each VN is connected to strictly more satisfied than unsatisfied neighboring CNs, for some set of VN values (these a VNs have non-zero values, while the remaining VNs are set to zero) [26]. If the configuration is unlabeled, d_1 , which is the number of degree-1 neighboring CNs of the set, is used instead of b . We focus on connected configurations. For canonical channels, e.g., the AWGN channel, elementary ASs (EASs) are the objects that dominate the error floor region of LDPC codes. EASs have the property that all satisfied CNs are of degree 2, and all unsatisfied CNs are of degree 1 [11], [27]. Unique characteristics of storage channels (compared with the AWGN channel) result in changing the combinatorial properties of detrimental objects in graph-based codes simulated over such channels [11].

We now present a motivating example demonstrating the combinatorial properties of detrimental objects in codes simulated over PR channels. We use the PR system described in [5]. The MR channel incorporates inter-symbol interference (intrinsic memory), jitter, and electronic noise. The normalized channel density [28], [29], which is the ratio of the read-head impulse response duration at half the amplitude to the bit duration, is 1.4. The PR equalization target is [8 14 2]. The receiver consists of filtering units followed by a Bahl-Cocke-Jelinek-Raviv (BCJR) detector [30], which is based on pattern-dependent noise prediction (PDNP) [31], in addition to a fast Fourier transform based q -ary sum-product algorithm (FFT-QSPA) LDPC decoder [32], with q being set to 2 in the case of binary codes. The number of global (detector-decoder) iterations is 10, and the number of local (decoder only) iterations is 20. Unless a codeword is reached, the decoder performs its prescribed number of local iterations for each global iteration. More details about this PR system can be found in [5]. We note

TABLE I

ERROR PROFILE OF CODE 1 AT SNR = 15.5 dB OVER THE PR CHANNEL

Absorbing set	(6, 0)	(6, 1)	(8, 0)	(10, 0)	Others
Percentage of errors	76%	7%	7%	4%	6%

TABLE II

ERROR PROFILE OF CODE 2 AT SNR = 15.5 dB OVER THE PR CHANNEL

Absorbing set	(6, 1)	(7, 1)	(8, 0)	(8, 1)	(10, 0)	Others
Percentage of errors	14%	6%	53%	6%	12%	9%

that this PR system is recommended by industry. In fact, the MR channel was directly provided to us by industry.

Example 1: Consider Codes 1 and 2 that are defined over $\text{GF}(4)$. The two codes have $\gamma = 3$, $\kappa = 19$, $z = 46$, $m = 1$, and $L = 5$. Thus, these codes have block length = 8,740 bits. Code 1 is uncoupled. Code 2 is an SC code designed using the CV technique for PR channels as described in [16], and it has rate ≈ 0.81 . The optimal cutting vector used for Code 2 is [4 9 15]. Codes 1 and 2 have SCB circulant powers of the form $f_{i,j} = f(i)f(j) = (i^2)(2j)$, and no specific optimization is performed on the edge weights.¹ The simulation plots of these codes are provided in Section VI.

We present the error profile of both codes at signal-to-noise ratio (SNR) = 15.5 dB, which is in the error floor region of both codes when simulated over the PR channel. The error profile classifies the collected errors according to the graphical configurations resulting in them. All the errors collected for both codes are AS errors. According to Table I of Code 1 and Table II of Code 2, the dominant errors are AS errors with few, if any, unsatisfied CNs.² Observe that the unlabeled configurations of the non-elementary (6, 1) and (8, 1) ASs with $\gamma = 3$ are (6, 0) and (8, 0) configurations, respectively. All ASs in Tables I and II have the (4, 4) configuration with $\gamma = 3$ as a common substructure; see Fig. 2, upper panel, for example. Moreover, in the case of $\gamma = 4$, our simulations indicate that all dominant ASs have the (4, 8) configuration as a common substructure; see Fig. 2, lower panel, for example.

The intrinsic memory in PR channels [2], [5] can result in VN errors having high magnitudes, which is typically not the case for canonical channels. These VN errors with high magnitudes make it very difficult for unsatisfied CNs with degree > 1 to participate in correcting an AS error. Consequently, it becomes more likely to have AS errors with unsatisfied CNs having degree ≥ 2 , which are non-elementary AS errors, e.g., the (6, 1) and (8, 1) AS errors in Example 1. Moreover, the detector-decoder looping (global iterations) helps the decoder correct AS errors with bigger numbers of unsatisfied CNs. In particular, when the number of global iterations is sufficient, the detector becomes capable of providing enough innovation at the decoder input in order to correct AS errors that are “unbalanced”. Thus, the objects that dominate the error floor region of LDPC codes simulated over PR channels can be

¹The multiplication $f_{i,j} = f(i)f(j)$ performed to compute circulant powers in SCB codes is (mod z)-multiplication.

²Here, few (a small number of) and many (a big number of) unsatisfied CNs are always relative to the number of VNs in the object.

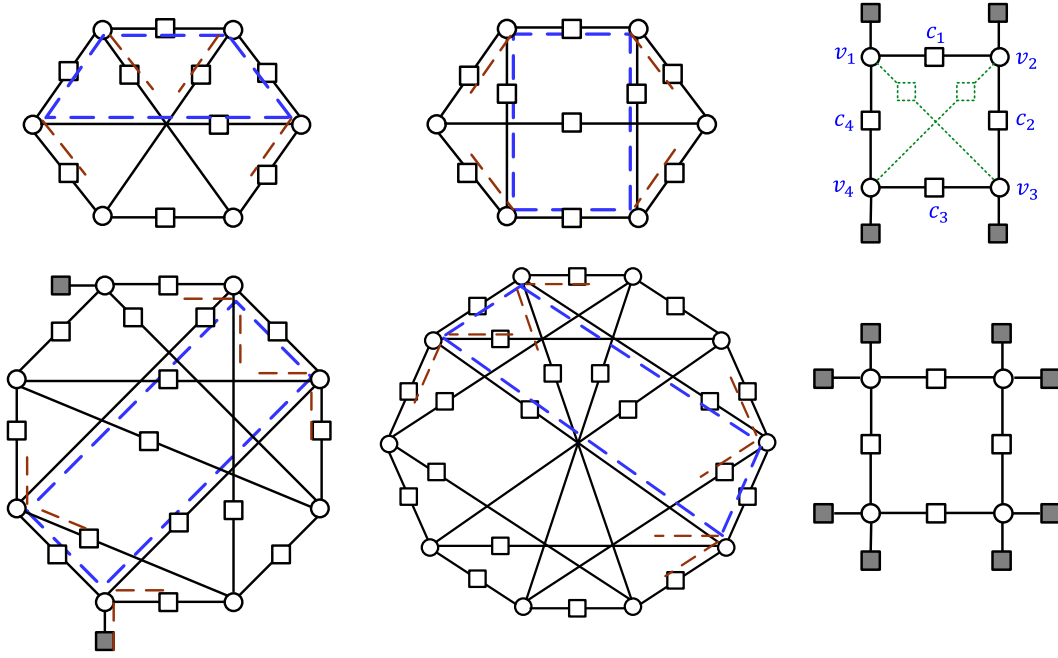


Fig. 2. The UBSs of multiple detrimental BASTs and the associated common substructures. Upper panel ($\gamma = 3$): two non-isomorphic $(6, 0, 9, 0)$ UBSs, and the common substructure is the $(4, 4)$ UAS. Lower panel ($\gamma = 4$): an $(8, 2, 15, 0)$ UBS and a $(10, 0, 20, 0)$ UBS, and the common substructure is the $(4, 8)$ UTS. Common substructures are marked with dashed blue and dashed brown lines. Internal connections in a cycle of length 8 are shown in dotted green lines in the $(4, 4)$ UAS. BASTs have the same configurations with appropriate edge labeling.

non-elementary, and they have a smaller number of unsatisfied (particularly degree-1) CNs, which is the reason why they are called “balanced”, i.e., harder to correct. Our extensive simulations confirm these combinatorial properties of the detrimental objects in the case of PR channels, which is consistent with the motivating example we introduced. These findings were introduced in [5] and then in [16] in the context of SC codes. BASs and BASs of type two (BASTs) were introduced in [5] and [11] to capture the detrimental objects in PR systems.

According to our prior results, the nature of detrimental objects in PR systems remain the same (BASTs) regardless from the PR equalization target being used. While we are using the PR equalization target $[8 \ 14 \ 2]$ in our simulations in this paper, other targets like the PR2 target, $[1 \ 2 \ 1]$, and extended PR4 (EPR4) target, $[1 \ 1 \ -1 \ -1]$, were extensively examined in [5] and the conclusions were the same. The variations in channel density values between 1 and 1.5 also did not affect the conclusions [5]. For various SC codes and with $\gamma \in \{3, 4\}$, the common substructure in different detrimental BASTs was shown in [16] to be consistent with the results in Example 1.

Remark 1: Typically, altering the quantization settings of the decoder changes the classes of detrimental absorbing sets in AWGN systems. In particular, more precise quantization results in the dominance of absorbing sets with fewer, if any, unsatisfied CNs. However, and as described above, a sufficient number of global iterations in PR systems already has the same result. As long as the quantization is reasonable (at least 2 bits for the integer part), altering it does not have a considerable effect on the classes of detrimental absorbing sets in PR systems.

We now present the definitions of different objects of interest. Examples of these objects of interest are in Fig. 2. Circles represent VNs. Grey squares (resp., white squares) represent degree-1 CNs in the unlabeled and unsatisfied CNs in the labeled (resp., degree-2 CNs in the unlabeled and satisfied CNs in the labeled) configurations. Let $g = \lfloor \frac{\gamma-1}{2} \rfloor$, which is the maximum number of unsatisfied CNs a VN can have in an AS.

Definition 1: Consider a subgraph induced by a subset $\mathcal{V}$ of VNs in the (Tanner) graph of a code. Set all the VNs in $\mathcal{V}$ to values $\in \text{GF}(q) \setminus \{0\}$ and set all other VNs to 0. The set $\mathcal{V}$ is said to be an (a, b, d_1, d_2, d_3) **balanced absorbing set of type two (BAST)** over $\text{GF}(q)$ if the size of $\mathcal{V}$ is a , the number of unsatisfied neighboring CNs of $\mathcal{V}$ is b , $0 \leq b \leq \lfloor \frac{aq}{2} \rfloor$, the number of degree-1 (resp., 2 and > 2) neighboring CNs of $\mathcal{V}$ is d_1 (resp., d_2 and d_3), $d_2 > d_3$, all the unsatisfied neighboring CNs of $\mathcal{V}$ (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 some set of VN values.

While the above definition was introduced in the context of non-binary codes [5], [11], it is valid in the binary case as well (set $q = 2$, and b becomes the number of odd-degree CNs). An (a, d_1, d_2, d_3) unlabeled BAST (UBS) is a BAST with the weights of all edges of its graph replaced by 1's. All our abbreviations are short-handed for simplicity.

Definition 2: Let $\mathcal{V}$ be a subset of VNs in the unlabeled graph (all edge weights are 1's) of a code. Let $\mathcal{O}$ (resp., $\mathcal{T}$ and $\mathcal{H}$) be the set of degree-1 (resp., 2 and > 2) neighboring CNs of $\mathcal{V}$. This graphical configuration is an (a, d_1) **unlabeled elementary trapping set (UTS)** if $|\mathcal{V}| = a$, $|\mathcal{O}| = d_1$, and

$|\mathcal{H}| = 0$. A UTS is an **unlabeled elementary absorbing set (UAS)** if each VN in $\mathcal{V}$ is connected to strictly more neighbors in $\mathcal{T}$ than in $\mathcal{O}$.

A binary protograph configuration is also defined by (a, d_1) for simplicity. The WCM framework removes a BAST from the graph of an NB code by careful processing of its edge weights (see [5], [11], and [12] for details).

III. THE COMMON SUBSTRUCTURE AND ITS PATTERNS

The idea of focusing on a common substructure in the design of the unlabeled graph of an SC code simplifies the optimization procedure. Additionally, minimizing the number of instances of the common substructure significantly reduces the multiplicity of several different types of detrimental objects simultaneously [16], [23], which is a lot more feasible compared with operating on all these detrimental objects separately (especially for partitioning). Example 1 demonstrates that in the case of $\gamma = 3$, the $(4, 4)$ UAS is the common substructure of interest for PR channels. More generally, it was shown in [16] that for $\gamma \geq 3$, the $(4, 4(\gamma - 2))$ UAS/UTS is the common substructure of interest for PR channels (unlike the case for AWGN [22], [23] and Flash channels [24], where the substructure of interest is the $(3, 3(\gamma - 2))$). Fig. 2 shows UBSs of multiple detrimental BASTs for codes with $\gamma \in \{3, 4\}$ simulated over PR channels, demonstrating that the common substructure of interest is the $(4, 4(\gamma - 2))$ UAS/UTS.

Remark 2: There are two reasons why we focus on the case of $\gamma \geq 3$ in our analysis:

- 1) Codes with $\gamma = 2$ have poor error floor performance since their graphs have large multiplicities of detrimental unlabeled low weight codewords. In fact, each cycle in a code with $\gamma = 2$ is an unlabeled codeword, i.e., an $(a, 0)$ UAS with a being half the cycle length, where $a \geq 4$ if the code has girth = 8. In order that these codes can have better error floor performance, bigger GF sizes should be used in the code design, which significantly increases the complexity of decoding and thus, is not advisable for data storage [9].
- 2) For codes having $\gamma = 2$, the concept of the common substructure of interest becomes inapplicable. This is because most of the unlabeled low weight codewords of interest, which are $(a, 0)$ UASs, with different values of a do not share any graphical structure (they are cycles having different lengths) in these codes.

Having said that the OO-CPO approach can still be useful to some extent for SC codes with $\gamma = 2$. For example, after applying some minor changes, the modified OO-CPO approach detailed here can be used to minimize the number of $(4, 0)$ UASs.

A configuration that is a cycle of length 2θ **with no internal connections** has exactly θ pairs of adjacent VNs, i.e., exactly θ pairs of directly connected VNs. Any additional pair of adjacent VNs is considered an internal connection. We note that the $(4, 4(\gamma - 2))$ UAS/UTS is a cycle of length 8 with no internal connections (ignore degree-1 CNs). From [33] (see also [24]), it is known that each cycle in the unlabeled

graph (the graph of $\mathbf{H}_{SC}$) is derived from a configuration in the protograph (the graph of the PM $\mathbf{H}_{SC}^p$) under specific conditions on the powers of the circulants involved in that cycle. Thus, in the OO stage, we operate on the protograph. Then, in the CPO stage, we operate on the circulant powers.

Remark 3: Let x^- be a positive integer s.t. $x^- < x$. The $(4, 4(\gamma - 2))$ UAS/UTS has 4 VNs, 4 degree-1 CNs, and also 4 degree-2 CNs. Note that a configuration with 4^- VNs and/or 4^- degree-2 CNs in the protograph of the code can result in $(4, 4(\gamma - 2))$ UASs/UTSs in the unlabeled graph depending on the circulant power arrangement. This is also true for a configuration with 4^- VNs and/or 4^- degree-1 CNs in the protograph of the code. Thus, in the OO stage, we operate on all protograph configurations that can result in $(4, 4(\gamma - 2))$ UASs/UTSs (cycles of length 8 with no internal connections) in the unlabeled graph, including the protograph configurations that do have internal connections. Then in the CPO stage, we treat the $(4, 4(\gamma - 2))$ UASs/UTSs and the $(4, 4(\gamma - 2) - 2\delta)$ UASs/UTSs differently, where $\delta \in \{1, 2\}$ is the number of existing internal connections in the configuration after lifting.

The major difference between the $(4, 4(\gamma - 2))$ UAS/UTS and the $(3, 3(\gamma - 2))$ UAS/UTS is that there are multiple distinct configurations in the protograph, ignoring degree-1 CNs and internal connections, that can generate the former object in the unlabeled graph. We call these different configurations **patterns**. A pattern is defined by the dimensions of the matrix of its subgraph. The following lemma investigates the number and nature of these patterns.

Lemma 1: The number of distinct patterns (with different dimensions) in the protograph of a code that can result in $(4, 4(\gamma - 2))$ UASs/UTSs in the unlabeled graph of the code after lifting is 9 in the case of $\gamma \geq 4$. The numbers of CNs and VNs in these 9 patterns are both in $\{2, 3, 4\}$. This number of distinct patterns reduces to 7 in the case of $\gamma = 3$.

Proof: Since the objects of interest in the unlabeled graph are cycles of length 8 with 4 CNs and 4 VNs, a protograph pattern that can generate some of them must have at most 4 CNs and 4 VNs. Moreover, to result in cycles of length 8 after lifting, the pattern must have at least 2 CNs and 2 VNs. Combining these two statements yields that the numbers of CNs and VNs of a protograph pattern that can result in $(4, 4(\gamma - 2))$ UASs/UTSs in the unlabeled graph must be in $\{2, 3, 4\}$.

Consequently, in order to have 9 distinct patterns for the case of $\gamma \geq 4$, we show that selecting any number of CNs in $\{2, 3, 4\}$ and any number of VNs in $\{2, 3, 4\}$ can result in a distinct pattern (one or more instances) that is capable of generating cycles of length 8 in the unlabeled graph. Fig. 3 illustrates this statement, focusing on the matrix representation of patterns and cycles. In the case of $\gamma = 3$, a pattern cannot have 4 ones in a column, which reduces the number of distinct patterns to 7. ■

We define the 9 patterns according to the dimensions of their submatrices in $\mathbf{H}_{SC}^p$ as follows. Pattern P_1 is 2×2 , Pattern P_2 is 2×3 , Pattern P_3 is 3×2 , Pattern P_4 is 2×4 , Pattern P_5 is 4×2 , Pattern P_6 is 3×3 , Pattern P_7 is 3×4 , Pattern P_8 is 4×3 , and Pattern P_9 is 4×4 (all illustrated in Fig. 3).

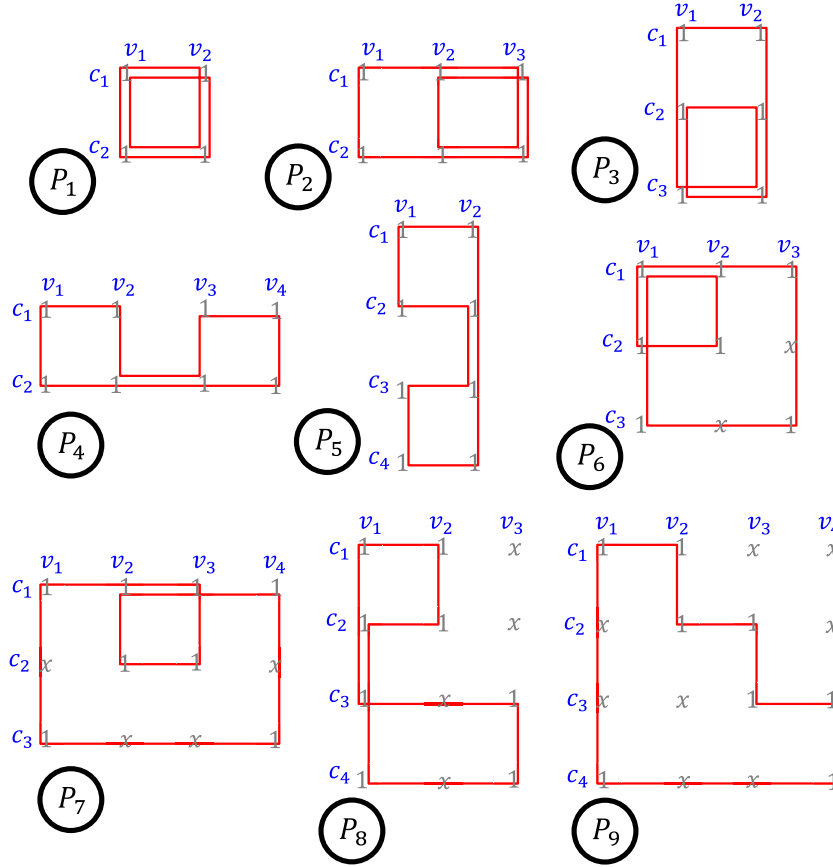


Fig. 3. The 9 protograph patterns that can result in cycles of length 8 in the unlabeled graph after lifting. One way of traversing each pattern to generate cycles of length 8, which is a cycle-8 candidate, is depicted in solid red. Note that only Pattern P_9 represents a cycle of length 8 in the protograph.

Remark 4: Following the same logic we used in Lemma 1 and its proof for the $(3, 3(\gamma - 2))$ UAS/UTS leads to a possibility to also have patterns for this case, with the number of CNs and VNs in $\{2, 3\}$. However, a careful analysis guides to the fact that only one protograph pattern can result in $(3, 3(\gamma - 2))$ UASs/UTSs (cycles of length 6) after lifting, which is the 3×3 pattern, and it is itself a cycle of length 6 [23], [24].

The following lemma discusses the relation between different protograph patterns and the resulting cycles after lifting. Define a **cycle-8 candidate** of Pattern P_ℓ as a way to traverse P_ℓ in order to reach cycles of length 8 in the unlabeled graph of the code after lifting. Some candidates are shown in Fig. 3.

Lemma 2: Let ζ_{P_ℓ} be the number of distinct cycle-8 candidates of Pattern P_ℓ . Then,

$$\zeta_{P_\ell} = \begin{cases} 1, & \ell \in \{1, 6, 9\}, \\ 2, & \ell \in \{7, 8\}, \\ 3, & \ell \in \{2, 3\}, \\ 6, & \ell \in \{4, 5\}. \end{cases} \quad (1)$$

Proof: We define a cycle-8 candidate according to the connectivity as follows: $c_1 - v_1 - c_2 - v_2 - c_3 - v_3 - c_4 - v_4$ (each CN connects the next two VNs in a circular fashion, see Fig. 2).³ From Fig. 3, there is only one cycle-8 candidate for

³In this paper, the notation v (resp., c) refers to the ontology or the index of a VN (resp., CN) depending on the context.

Pattern P_1 , which is $c_1 - v_1 - c_2 - v_2 - c_1 - v_1 - c_2 - v_2$, and this is the case for all square patterns. Thus, $\zeta_{P_\ell} = 1$ for $\ell \in \{1, 6, 9\}$. It can be understood from Fig. 3 that $\zeta_{P_\ell} \neq 1$ for all the remaining patterns. In particular, we have two cycle-8 candidates for Pattern P_7 , that are: $c_1 - v_1 - c_2 - v_2 - c_1 - v_3 - c_3 - v_4$ and $c_1 - v_1 - c_2 - v_3 - c_1 - v_2 - c_3 - v_4$ (which is the red cycle on P_7 in Fig. 3). The situation is the same for Pattern P_8 because it is the transpose of P_7 . Thus, $\zeta_{P_\ell} = 2$ for $\ell \in \{7, 8\}$. The rest of the cases can be derived similarly. ■

Pattern P_1 has $\zeta_{P_\ell} = 1$ (see (1)), and it results in $z/2$ or 0 cycles of length 8 after lifting (since P_1 is only 2×2), while all the remaining patterns result in z or 0 cycles of length 8 after lifting [24], [33]. Thus, we define the **pattern weight**, β_{P_ℓ} , which plays an important role in the discrete optimization problem of the OO, as follows:

$$\beta_{P_\ell} = \begin{cases} 1/2, & \ell = 1, \\ \zeta_{P_\ell}, & \ell \in \{2, 3, 4, 5, 6, 7, 8, 9\}. \end{cases} \quad (2)$$

IV. OO: BUILDING AND SOLVING THE OPTIMIZATION PROBLEM

Now, we are ready to build the optimization problem. Consider the protograph of an SC code. The **weighted sum** of the total number of instances of all patterns is given by:

$$F_{\text{sum}} = \sum_{\ell=1}^9 \beta_{P_\ell} F_{P_\ell}, \quad (3)$$

where F_{P_ℓ} is the total number of instances of Pattern P_ℓ . The goal is to express F_{sum} , through F_{P_ℓ} , $\forall \ell$, as a function of the overlap parameters, then find the optimal set of overlap parameters that minimizes F_{sum} for OO partitioning. We first recall the definition and the properties of overlap parameters. More details can be found in [23].

Definition 3: For any m , let $\Pi_1^1 = [\mathbf{H}_0^T \mathbf{H}_1^T \dots \mathbf{H}_m^T]^T$, and let $\Pi_1^{1,p}$ be its PM (of size $(m+1)\gamma \times \kappa$). Consequently, $\Pi_1^{1,p} = [(\mathbf{H}_0^p)^T (\mathbf{H}_1^p)^T \dots (\mathbf{H}_m^p)^T]^T$. A **degree- μ overlap** among μ rows (or CNs) of $\Pi_1^{1,p}$ indexed by $\{i_1, \dots, i_\mu\}$, $1 \leq \mu \leq \gamma$, $0 \leq i_1, \dots, i_\mu \leq (m+1)\gamma - 1$, is defined as a position (column) in which all these rows have 1's simultaneously. A **degree- μ overlap parameter**, $t_{\{i_1, \dots, i_\mu\}}$, is defined as the number of degree- μ overlaps among the rows indexed by $\{i_1, \dots, i_\mu\}$ in $\Pi_1^{1,p}$. A degree-1 overlap parameter t_{i_1} , $0 \leq i_1 \leq (m+1)\gamma - 1$, is defined as the number of 1's in row i_1 of $\Pi_1^{1,p}$.

Note that a degree- μ overlap parameter, if $\mu > 1$, is always zero if in the set $\{i_1, \dots, i_\mu\}$ there exists at least one pair of distinct row indices, say (i_{τ_1}, i_{τ_2}) , with the property that $i_{\tau_1} \equiv i_{\tau_2} \pmod{\gamma}$ [23]. In words, the same rows at different protograph component matrices do not overlap at any position. This follows from the definition of component matrices, particularly that they are disjoint. Denote the set of all non-zero overlap parameters by $\mathcal{O}$. In the context of overlap parameters, non-zero means not guaranteed to be zero. The parameters in $\mathcal{O}$ are not entirely independent. The set of all independent non-zero overlap parameters, $\mathcal{O}_{\text{ind}}$, is:

$$\mathcal{O}_{\text{ind}} = \{t_{\{i_1, \dots, i_\mu\}} \mid 1 \leq \mu \leq \gamma, 0 \leq i_1, \dots, i_\mu \leq m\gamma - 1, \forall \{i_{\tau_1}, i_{\tau_2}\} \subseteq \{i_1, \dots, i_\mu\} i_{\tau_1} \not\equiv i_{\tau_2} \pmod{\gamma}\}. \quad (4)$$

The other non-zero overlap parameters in $\mathcal{O} \setminus \mathcal{O}_{\text{ind}}$ are obtained from the parameters in $\mathcal{O}_{\text{ind}}$ according to [23, Lemma 3]. In other words, any overlap parameter with at least one of its rows in the last protograph component matrix $\mathbf{H}_m^p$ can be derived via a linear combination involving overlap parameters with no rows in the last protograph component matrix $\mathbf{H}_m^p$. This follows from that $\mathbf{H}^p = \sum_{y=0}^m \mathbf{H}_y^p$. Example 2 further illustrates the concept of independent non-zero overlap parameters.

The cardinality of the set $\mathcal{O}_{\text{ind}}$, which determines the complexity of the discrete optimization problem of the OO stage, is given by (see also [23, Lemma 4] for more details):

$$|\mathcal{O}_{\text{ind}}| = (m+1)^\gamma - 1. \quad (5)$$

As demonstrated in Fig. 3, for all the patterns of interest, the highest overlap degree is $\mu = 4$ (a pattern has at most 4 CNs). Note that while the overlap parameters themselves must be restricted to $\Pi_1^{1,p}$, the concept of the degree- μ overlap can be generalized from $\Pi_1^{1,p}$ to the PM of the SC code, $\mathbf{H}_{\text{SC}}^p$. We will use this generalization in the analysis of patterns.

Example 2: Consider the case of $\gamma = 4$ and $m = 1$. From (5), we have $|\mathcal{O}_{\text{ind}}| = 2^4 - 1 = 15$ independent non-zero overlap parameters. In particular, $\mathcal{O}_{\text{ind}} = \{t_0, t_1, t_2, t_3, t_{\{0,1\}}, t_{\{0,2\}}, t_{\{0,3\}}, t_{\{1,2\}}, t_{\{1,3\}}, t_{\{2,3\}}, t_{\{0,1,2\}}, t_{\{0,1,3\}}, t_{\{0,2,3\}}, t_{\{1,2,3\}}, t_{\{0,1,2,3\}}\}$. Any overlap parameter that is not

in $\mathcal{O}_{\text{ind}}$ involves at least one row with its index in $\{4, 5, 6, 7\}$ (a row in $\mathbf{H}_m^p$). Such overlap parameter can be obtained from the ones in $\mathcal{O}_{\text{ind}}$. For example, the overlap parameter $t_{\{0,6\}}$ involves the row indexed by 6, which is the third row in $\mathbf{H}_m^p = \mathbf{H}_1^p$. This overlap parameter is not independent, and it is computed as follows: $t_{\{0,6\}} = t_0 - t_{\{0,2\}}$. Similarly, the overlap parameter $t_{\{6,7\}}$ involves the rows indexed by 6 and 7, which are the third and fourth rows in $\mathbf{H}_m^p = \mathbf{H}_1^p$, respectively. This overlap parameter is not independent, and it is computed as follows: $t_{\{6,7\}} = \kappa - t_2 - t_3 + t_{\{2,3\}}$. These calculations can be verified using Fig. 12.

We aim at expressing F_{P_ℓ} , $\forall \ell$, in terms of the parameters in $\mathcal{O}_{\text{ind}}$. Let $\mathbf{R}_r$ be a replica in which at least one VN of the pattern being studied exists. We call $\mathbf{R}_r$ the reference replica. Moreover, let the CNs (or rows) of the pattern be of the form $c_x = (r-1)\gamma + i_x$, $1 \leq x \leq 4$. Here, c_x is the index of the row in $\mathbf{H}_{\text{SC}}^p$ corresponding to the CN. In the following, we consider the protograph of an SC code with parameters $\gamma \geq 3$, κ , m , L , and $\mathcal{O}$. We define $(x)^+ = \max\{x, 0\}$, and $F_{P_\ell, 1}^k$ as the number of instances of Pattern P_ℓ that start at replica $\mathbf{R}_1$ and span k consecutive replicas. Here, “start” and “span” are both with respect to the VNs of these instances. Note that each VN in a pattern corresponds to an overlap (see the patterns in Fig. 3).

As we shall see later, a Pattern P_ℓ spans at most χ consecutive replicas, where χ either $= m+1$ or $= 2m+1$, depending on the value of ℓ . In the math, we always consider the case of $L \geq \chi$.

We say here that i_x is the **start of replica** $\mathbf{R}_\rho$ if i_x is the index of the first non-zero row in $\mathbf{R}_\rho$ relative to $\mathbf{R}_r$. We also say that i_y is the **end of replica** $\mathbf{R}_\rho$ if i_y is the index of the last non-zero row in $\mathbf{R}_\rho$ relative to $\mathbf{R}_r$. In particular, the start and end of replica $\mathbf{R}_{r+\nu}$ are $\nu\gamma$ and $(m+\nu+1)\gamma-1$, respectively. For example, the start and end of $\mathbf{R}_r$ are 0 and $(m+1)\gamma-1$, respectively, regardless from the value of r since $\mathbf{R}_r$ is the reference replica. Moreover, the start and end of $\mathbf{R}_{r+2}$ (resp., $\mathbf{R}_{r-1}$) are 2γ and $(m+3)\gamma-1$ (resp., $-\gamma$ and $m\gamma-1$). Furthermore, the indices 1, h , w , and k of replicas are always s.t. $1 < k$ for two replicas, $1 < h < k$ for three replicas, and $1 < h < w < k$ for four replicas.

The steps we take to compute F_{P_ℓ} , for any Pattern P_ℓ , in terms of the overlap parameters are as follows:

- 1) We find the maximum number of consecutive replicas P_ℓ can span, which is χ . If all pairs of VNs in the pattern are adjacent (directly connected), $\chi = m+1$. Otherwise, $\chi = 2m+1$.
- 2) We specify the different existence possibilities of P_ℓ (the cases) according to all feasible distributions of its VNs across replicas. For example, P_6 has three VNs, two correspond to degree-2 overlaps and one corresponds to a degree-3 overlap. Thus, the existence possibilities are: a) All overlaps are in one replica. b) The two degree-2 overlaps are in one replica, and the degree-3 is in another replica. c) One degree-2 overlap along with the degree-3 overlap are in one replica, and the other degree-2 overlap is in another replica. d) The three overlaps are in three distinct replicas.

- 3) For each of the cases specified above, we find an expression for the number of instances of P_ℓ .
- 4) We compute $F_{P_\ell,1}^k$, for all $k \in \{1, 2, \dots, \chi\}$, through the case expressions from the previous step. In particular, $F_{P_\ell,1}^1$ (P_ℓ spans one replica) is a summation, over all possible overlap parameter values, of the case expression when all overlaps are in one replica. Then, $F_{P_\ell,1}^2$ (P_ℓ spans two replicas) is a summation of the case expression(s) when overlaps are in two replicas. Then, $F_{P_\ell,1}^3$ (P_ℓ spans three replicas) is a summation of the case expression(s) when overlaps are in two replicas (the first and the last of the three) and the case expression(s) when overlaps are in three replicas (if possible). This continues until until $F_{P_\ell,1}^\chi$.
- 5) We use [23, Theorem 1] to find F_{P_ℓ} from $F_{P_\ell,1}^k$, for all $k \in \{1, 2, \dots, \chi\}$, as follows:

$$F_{P_\ell} = \sum_{k=1}^{\chi} (L - k + 1) F_{P_\ell,1}^k. \quad (6)$$

The idea of (6) is to exploit the repetitive nature of the SC parity-check matrix in order to significantly reduce the computations in the OO stage (and later on, simplify enumerating objects in the CPO stage). In particular, an instance of Pattern P_ℓ that starts at Replica $\mathbf{R}_1$ and spans k consecutive replicas appears again starting from Replica $\mathbf{R}_2$, Replica $\mathbf{R}_3$, ..., Replica $\mathbf{R}_{L-k+1}$. Thus, it suffices to compute $F_{P_\ell,1}^k$, for any k , and then multiply by $(L - k + 1)$.

In the forthcoming subsections, we provide the analysis for Patterns P_1 , P_2 , and P_6 . For each pattern, there is a lemma performing Steps 2 and 3, followed by a theorem performing Steps 4 and 5. The proofs of all lemmas and theorems in this section are in Appendices A, B, and C. The analysis of the remaining six patterns is in [34], and it is left out of the paper for better readability.

A. Analysis of Pattern P_1 (size 2×2)

This pattern has two VNs, which are *adjacent* (connected via at least one path with only one CN). Thus, Pattern P_1 has its VNs located in at most two replicas, and the pattern spans (i.e., its VNs span) at most $m + 1$ consecutive replicas (see [23, Lemma 1]). Suppose P_1 has the CNs c_1 and c_2 . The two overlaps forming the pattern are of degree 2, and they are both $c_1 - c_2$ overlaps (among c_1 and c_2).

Lemma 3: Case 1.1: The number of instances of P_1 with CNs c_1 and c_2 , and all overlaps in one replica, $\mathbf{R}_r$, is:

$$\mathcal{A}_{P_1}(t_{\{i_1, i_2\}}) = \binom{t_{\{i_1, i_2\}}}{2}. \quad (7)$$

Case 1.2: The number of instances of P_1 with CNs c_1 and c_2 , and overlaps in two replicas, $\mathbf{R}_r$ and $\mathbf{R}_e$, $r < e$, is:

$$\begin{aligned} \mathcal{B}_{P_1}(t_{\{i_1, i_2\}}, t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}) \\ = t_{\{i_1, i_2\}} t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}. \end{aligned} \quad (8)$$

The two cases are illustrated in Fig. 4.

Theorem 1: The total number of instances of Pattern P_1 in the binary protograph of an SC code that has parameters

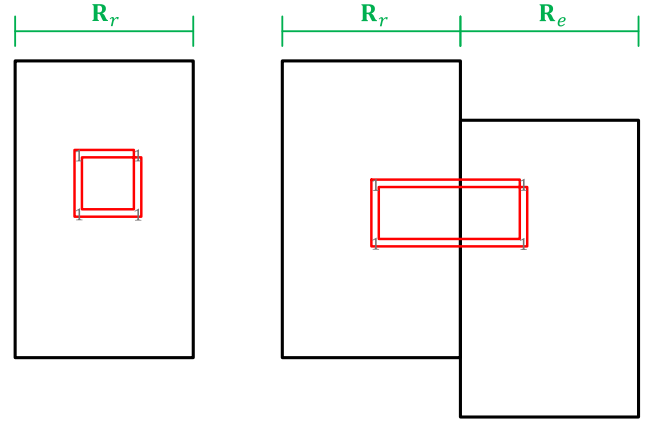


Fig. 4. An instance of Pattern P_1 in Case 1.1 (overlaps are in one replica) and in Case 1.2 (overlaps are in two replicas), from left to right. For simplicity, we have $e = r + 1$.

$\gamma \geq 3$, κ , m , $L \geq m + 1$, and $\mathcal{O}$, is:

$$F_{P_1} = \sum_{k=1}^{m+1} (L - k + 1) F_{P_1,1}^k, \quad (9)$$

where $F_{P_1,1}^k$, $k \in \{1, 2, \dots, m + 1\}$, are given by:

$$\begin{aligned} F_{P_1,1}^1 &= \sum_{\{i_1, i_2\} \subset \{0, \dots, (m+1)\gamma-1\}} \mathcal{A}_{P_1}(t_{\{i_1, i_2\}}), \\ F_{P_1,1}^{k \geq 2} &= \sum_{\{i_1, i_2\} \subset \{(k-1)\gamma, \dots, (m+1)\gamma-1\}} \mathcal{B}_{P_1}(t_{\{i_1, i_2\}}, t_{\{i_1+(1-k)\gamma, i_2+(1-k)\gamma\}}), \end{aligned} \quad (10)$$

with $\overline{i_1} \neq \overline{i_2}$, and $\overline{i_x}$ is defined by: $\overline{i_x} = (i_x \bmod \gamma)$.

B. Analysis of Pattern P_2 (size 2×3)

This pattern has three VNs, with each two of them being adjacent. Thus, P_2 spans at most $m + 1$ consecutive replicas. Suppose P_2 has the CNs c_1 and c_2 . The three overlaps forming P_2 are of degree 2, and they are all $c_1 - c_2$ overlaps.

Lemma 4: Case 2.1: The number of instances of P_2 with CNs c_1 and c_2 , and all overlaps in one replica, $\mathbf{R}_r$, is:

$$\mathcal{A}_{P_2}(t_{\{i_1, i_2\}}) = \binom{t_{\{i_1, i_2\}}}{3}. \quad (11)$$

Case 2.2: The number of instances of P_2 with CNs c_1 and c_2 , and all overlaps in two replicas s.t. two overlaps are in $\mathbf{R}_r$, and one overlap is in $\mathbf{R}_e$, is:

$$\begin{aligned} \mathcal{B}_{P_2}(t_{\{i_1, i_2\}}, t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}) \\ = \binom{t_{\{i_1, i_2\}}}{2} t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}. \end{aligned} \quad (12)$$

Case 2.3: The number of instances of P_2 with CNs c_1 and c_2 , and overlaps in three replicas (one in each), $\mathbf{R}_r$, $\mathbf{R}_e$, and $\mathbf{R}_s$, $r < e < s$, is:

$$\begin{aligned} \mathcal{C}_{P_2}(t_{\{i_1, i_2\}}, t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}, t_{\{i_1+(r-s)\gamma, i_2+(r-s)\gamma\}}) \\ = t_{\{i_1, i_2\}} t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}} t_{\{i_1+(r-s)\gamma, i_2+(r-s)\gamma\}}. \end{aligned} \quad (13)$$

The three cases are illustrated in Fig 5.

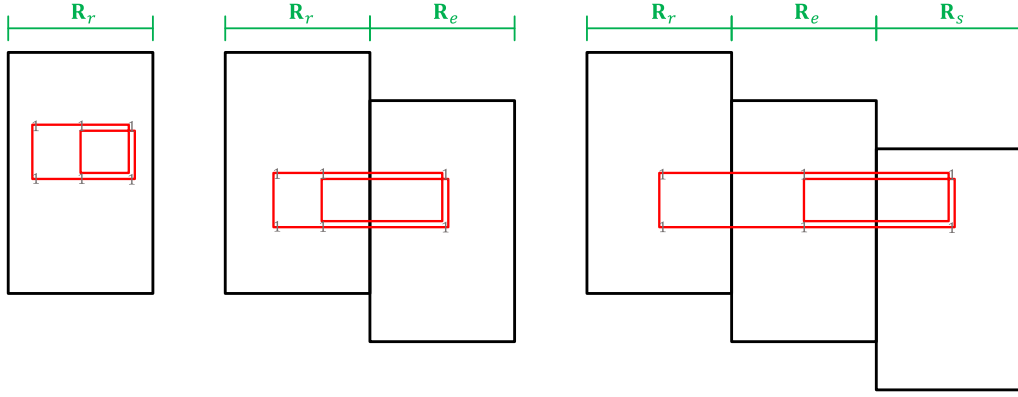


Fig. 5. An instance of Pattern P_2 in Case 2.1 (overlaps are in one replica), in Case 2.2 (overlaps are in two replicas), and in Case 2.3 (overlaps are in three replicas), from left to right. For simplicity, we have $e = r + 1$ and $s = e + 1$.

Theorem 2: The total number of instances of Pattern P_2 in the binary protograph of an SC code that has parameters $\gamma \geq 3$, κ , m , $L \geq m + 1$, and $\mathcal{O}$, is:

$$F_{P_2} = \sum_{k=1}^{m+1} (L - k + 1) F_{P_2,1}^k, \quad (14)$$

where $F_{P_2,1}^k$, $k \in \{1, 2, \dots, m + 1\}$, are given by:

$$\begin{aligned} F_{P_2,1}^1 &= \sum_{\{i_1, i_2\} \subset \{0, \dots, (m+1)\gamma-1\}} \mathcal{A}_{P_2}(t_{\{i_1, i_2\}}), \\ F_{P_2,1}^2 &= \sum_{\{i_1, i_2\} \subset \{\gamma, \dots, (m+1)\gamma-1\}} \mathcal{B}_{P_2}(t_{\{i_1, i_2\}}, t_{\{i_1-\gamma, i_2-\gamma\}}) \\ &\quad + \sum_{\{i_1, i_2\} \subset \{0, \dots, m\gamma-1\}} \mathcal{B}_{P_2}(t_{\{i_1, i_2\}}, t_{\{i_1+\gamma, i_2+\gamma\}}), \\ F_{P_2,1}^{k \geq 3} &= \sum_{\{i_1, i_2\} \subset \{(k-1)\gamma, \dots, (m+1)\gamma-1\}} \mathcal{B}_{P_2}(t_{\{i_1, i_2\}}, t_{\{i_1+(1-k)\gamma, i_2+(1-k)\gamma\}}) \\ &\quad + \sum_{\{i_1, i_2\} \subset \{0, \dots, (m-k+2)\gamma-1\}} \mathcal{B}_{P_2}(t_{\{i_1, i_2\}}, t_{\{i_1+(k-1)\gamma, i_2+(k-1)\gamma\}}) \\ &\quad + \sum_{h=2}^{k-1} \sum_{\{i_1, i_2\} \subset \{(k-1)\gamma, \dots, (m+1)\gamma-1\}} \mathcal{C}_{P_2}(t_{\{i_1, i_2\}}, t_{\{i_1+(1-h)\gamma, i_2+(1-h)\gamma\}}) \\ &\quad \cdot t_{\{i_1+(1-k)\gamma, i_2+(1-k)\gamma\}}, \end{aligned} \quad (15)$$

with $\bar{i}_1 \neq \bar{i}_2$.

C. Analysis of Pattern P_6 (size 3×3)

This pattern has three VNs, with each two of them being adjacent. Thus, P_6 spans at most $m + 1$ consecutive replicas. Suppose P_6 has the CNs c_1 , c_2 , and c_3 . Define **distinct overlaps** to be overlaps from different families, i.e., overlaps among different sets of CNs. Pattern P_6 is formed of three overlaps; two (distinct) of degree-2 and one of degree-3. Define c_1 as the CN connecting the three VNs. Thus, the overlaps are $c_1 - c_2$, $c_1 - c_3$, and $c_1 - c_2 - c_3$ (see P_6 in Fig. 3). Again, each VN corresponds to an overlap.

Lemma 5: Case 6.1: The number of instances of P_6 with CNs c_1 , c_2 , and c_3 as defined in the previous paragraph, and

all overlaps in one replica, $\mathbf{R}_r$, is:

$$\begin{aligned} \mathcal{A}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_3\}}, t_{\{i_1, i_2, i_3\}}) \\ = t_{\{i_1, i_2, i_3\}} (t_{\{i_1, i_2, i_3\}} - 1)^+ (t_{\{i_1, i_3\}} - 2)^+ \\ + t_{\{i_1, i_2, i_3\}} (t_{\{i_1, i_2\}} - t_{\{i_1, i_2, i_3\}}) (t_{\{i_1, i_3\}} - 1)^+. \end{aligned} \quad (16)$$

Case 6.2: The number of instances of P_6 with CNs c_1 , c_2 , and c_3 as defined in the previous paragraph, and all overlaps in two replicas s.t. the two degree-2 overlaps are in $\mathbf{R}_r$, and the degree-3 overlap is in $\mathbf{R}_e$, is:

$$\begin{aligned} \mathcal{B}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_3\}}, t_{\{i_1, i_2, i_3\}} \\ , t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma, i_3+(r-e)\gamma\}}) \\ = [t_{\{i_1, i_2, i_3\}} (t_{\{i_1, i_3\}} - 1)^+ \\ + (t_{\{i_1, i_2\}} - t_{\{i_1, i_2, i_3\}}) t_{\{i_1, i_3\}}] \\ \cdot t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma, i_3+(r-e)\gamma\}}. \end{aligned} \quad (17)$$

Case 6.3: The number of instances of P_6 with CNs c_1 , c_2 , and c_3 as defined in the previous paragraph, and all overlaps in two replicas s.t. the degree-3 overlap and the $c_1 - c_2$ overlap are in $\mathbf{R}_r$, and the $c_1 - c_3$ overlap is in $\mathbf{R}_e$, is:

$$\begin{aligned} \mathcal{C}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1+(r-e)\gamma, i_3+(r-e)\gamma\}}) \\ = t_{\{i_1, i_2, i_3\}} (t_{\{i_1, i_2\}} - 1)^+ t_{\{i_1+(r-e)\gamma, i_3+(r-e)\gamma\}}. \end{aligned} \quad (18)$$

Case 6.4: The number of instances of P_6 with CNs c_1 , c_2 , and c_3 as defined in the previous paragraph, and overlaps in three replicas s.t. the $c_1 - c_2$ overlap is in $\mathbf{R}_r$, the $c_1 - c_3$ overlap is in $\mathbf{R}_e$, and the degree-3 overlap is in $\mathbf{R}_s$, $r < e$, is:

$$\begin{aligned} \mathcal{D}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1+(r-e)\gamma, i_3+(r-e)\gamma\}} \\ , t_{\{i_1+(r-s)\gamma, i_2+(r-s)\gamma, i_3+(r-s)\gamma\}}) \\ = t_{\{i_1, i_2\}} t_{\{i_1+(r-e)\gamma, i_3+(r-e)\gamma\}} \\ \cdot t_{\{i_1+(r-s)\gamma, i_2+(r-s)\gamma, i_3+(r-s)\gamma\}}. \end{aligned} \quad (19)$$

Three of the four cases are illustrated in Fig. 6.

Theorem 3: The total number of instances of Pattern P_6 in the binary protograph of an SC code that has parameters $\gamma \geq 3$, κ , m , $L \geq m + 1$, and $\mathcal{O}$, is:

$$F_{P_6} = \sum_{k=1}^{m+1} (L - k + 1) F_{P_6,1}^k, \quad (20)$$

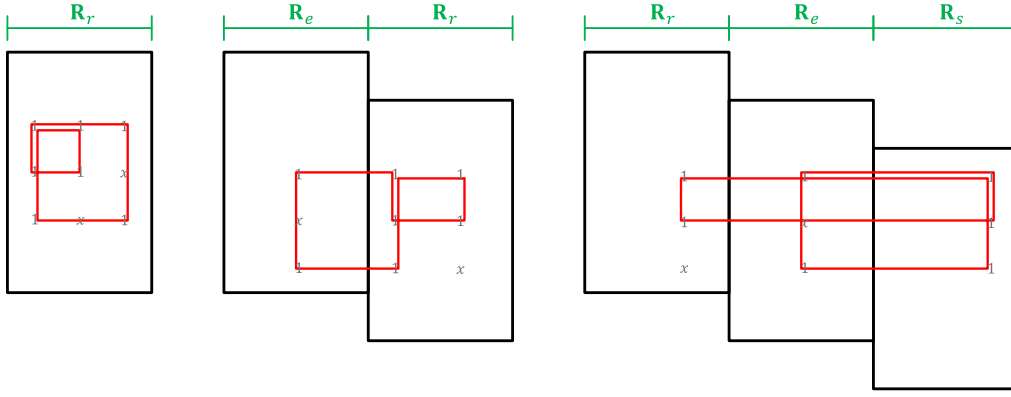


Fig. 6. An instance of Pattern P_6 in Case 6.1 (overlaps are in one replica), in Case 6.3 (overlaps are in two replicas), and in Case 6.4 (overlaps are in three replicas), from left to right. For simplicity, we have $e = r + y$, where $y \in \{-1, 1\}$, and $s = e + 1$.

where $F_{P_6,1}^k$, $k \in \{1, 2, \dots, m+1\}$, are given by (21), shown at the bottom of the page, with $\overline{i_1} \neq \overline{i_2}$, $\overline{i_1} \neq \overline{i_3}$, and $\overline{i_2} \neq \overline{i_3}$.

As mentioned before, the analysis of the remaining six patterns is in [34].

After deriving the expressions of F_{P_ℓ} , $\forall \ell$, as functions of the overlap parameters in $\mathcal{O}$, we use (3), (4), and [23, Lemma 3] to express F_{sum} as a function of the parameters in $\mathcal{O}_{\text{ind}}$ (which is the set of all independent non-zero overlap parameters). Thus,

our **discrete optimization problem** is:

$$F_{\text{sum}}^* = \min_{\mathcal{O}_{\text{ind}}} F_{\text{sum}}. \quad (22)$$

The constraints of the optimization problem in (22) are linear constraints capturing the interval constraints under which the resultant partitioning is valid. These constraints can be directly obtained from the combinatorial formula used to compute the number of partitioning choices given a specific set of values for $\mathcal{O}_{\text{ind}}$ (there is an example below). We also add the

$$\begin{aligned}
 F_{P_6,1}^1 &= \sum_{i_1 \in \{0, \dots, (m+1)\gamma-1\}, \{i_2, i_3\} \subset \{0, \dots, (m+1)\gamma-1\}} \mathcal{A}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_3\}}, t_{\{i_1, i_2, i_3\}}), \\
 F_{P_6,1}^2 &= \sum_{i_1 \in \{\gamma, \dots, (m+1)\gamma-1\}, \{i_2, i_3\} \subset \{\gamma, \dots, (m+1)\gamma-1\}} \mathcal{B}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_3\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1-\gamma, i_2-\gamma, i_3-\gamma\}}) \\
 &\quad + \sum_{i_1 \in \{0, \dots, m\gamma-1\}, \{i_2, i_3\} \subset \{0, \dots, m\gamma-1\}} \mathcal{B}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_3\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1+\gamma, i_2+\gamma, i_3+\gamma\}}) \\
 &\quad + \sum_{i_1 \in \{\gamma, \dots, (m+1)\gamma-1\}, i_2 \in \{0, \dots, (m+1)\gamma-1\}, i_3 \in \{\gamma, \dots, (m+1)\gamma-1\}} \mathcal{C}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1-\gamma, i_3-\gamma\}}) \\
 &\quad + \sum_{i_1 \in \{0, \dots, m\gamma-1\}, i_2 \in \{0, \dots, (m+1)\gamma-1\}, i_3 \in \{0, \dots, m\gamma-1\}} \mathcal{C}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1+\gamma, i_3+\gamma\}}), \\
 F_{P_6,1}^{k \geq 3} &= \sum_{i_1 \in \{(k-1)\gamma, \dots, (m+1)\gamma-1\}, \{i_2, i_3\} \subset \{(k-1)\gamma, \dots, (m+1)\gamma-1\}} \mathcal{B}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_3\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1+(1-k)\gamma, i_2+(1-k)\gamma, i_3+(1-k)\gamma\}}) \\
 &\quad + \sum_{i_1 \in \{0, \dots, (m-k+2)\gamma-1\}, \{i_2, i_3\} \subset \{0, \dots, (m-k+2)\gamma-1\}} \mathcal{B}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_3\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1+(k-1)\gamma, i_2+(k-1)\gamma, i_3+(k-1)\gamma\}}) \\
 &\quad + \sum_{i_1 \in \{(k-1)\gamma, \dots, (m+1)\gamma-1\}, i_2 \in \{0, \dots, (m+1)\gamma-1\}, i_3 \in \{(k-1)\gamma, \dots, (m+1)\gamma-1\}} \mathcal{C}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1+(1-k)\gamma, i_3+(1-k)\gamma\}}) \\
 &\quad + \sum_{i_1 \in \{0, \dots, (m-k+2)\gamma-1\}, i_2 \in \{0, \dots, (m+1)\gamma-1\}, i_3 \in \{0, \dots, (m-k+2)\gamma-1\}} \mathcal{C}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1, i_2, i_3\}}, t_{\{i_1+(k-1)\gamma, i_3+(k-1)\gamma\}}) \\
 &\quad + \sum_{h=2}^{k-1} \sum_{i_1 \in \{(k-1)\gamma, \dots, (m+1)\gamma-1\}, i_2 \in \{(k-1)\gamma, \dots, (m+1)\gamma-1\}, i_3 \in \{(k-1)\gamma, \dots, (m+h)\gamma-1\}} \mathcal{D}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1+(1-h)\gamma, i_3+(1-h)\gamma\}}, t_{\{i_1+(1-k)\gamma, i_2+(1-k)\gamma, i_3+(1-k)\gamma\}}) \\
 &\quad + \sum_{h=2}^{k-1} \sum_{i_1 \in \{(k-1)\gamma, \dots, (m+1)\gamma-1\}, i_2 \in \{(h-1)\gamma, \dots, (m+1)\gamma-1\}, i_3 \in \{(k-1)\gamma, \dots, (m+h)\gamma-1\}} \mathcal{D}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1+(1-k)\gamma, i_3+(1-k)\gamma\}}, t_{\{i_1+(1-h)\gamma, i_2+(1-h)\gamma, i_3+(1-h)\gamma\}}) \\
 &\quad + \sum_{h=2}^{k-1} \sum_{i_1 \in \{(k-h)\gamma, \dots, (m-h+2)\gamma-1\}, i_2 \in \{0, \dots, (m-h+2)\gamma-1\}, i_3 \in \{(k-h)\gamma, \dots, (m-h+2)\gamma-1\}} \mathcal{D}_{P_6}(t_{\{i_1, i_2\}}, t_{\{i_1+(h-k)\gamma, i_3+(h-k)\gamma\}}, t_{\{i_1+(h-1)\gamma, i_2+(h-1)\gamma, i_3+(h-1)\gamma\}}). \quad (21)
 \end{aligned}$$

balanced partitioning constraint, which guarantees a balanced distribution of the non-zero circulants among the $(m + 1)$ component matrices (see also [23] and [24]). A balanced partitioning is preferred in order to prevent the situation where a group of non-zero elements in a particular component matrix are involved in significantly more cycles than the remaining non-zero elements. This constraint, although it might result in a slightly suboptimal solution in the protograph (in few cases), is observed to be very beneficial when we apply the CPO to construct the final code.

As with the set $\mathcal{O}_{\text{ind}}$, the optimization constraints depend only on code parameters, and not on the common substructure of interest (which depends on the channel). For example, in the case of $\gamma = 3$, $m = 1$, and any κ , $\mathcal{O}_{\text{ind}} = \{t_0, t_1, t_2, t_{\{0,1\}}, t_{\{0,2\}}, t_{\{1,2\}}, t_{\{0,1,2\}}\}$. Let, $\mathcal{Z}$ be the number of partitioning choices given a specific set of values for $\mathcal{O}_{\text{ind}}$. Then,

$$\mathcal{Z} = \binom{k}{t_0} \binom{t_0}{t_{\{0,1\}}} \binom{\kappa - t_0}{t_1 - t_{\{0,1\}}} \cdot \binom{t_{\{0,1\}}}{t_{\{0,1,2\}}} \binom{t_0 - t_{\{0,1\}}}{t_{\{0,2\}} - t_{\{0,1,2\}}} \binom{t_1 - t_{\{0,1\}}}{t_{\{1,2\}} - t_{\{0,1,2\}}} \cdot \binom{\kappa - t_0 - t_1 + t_{\{0,1\}}}{t_2 - t_{\{0,2\}} - t_{\{1,2\}} + t_{\{0,1,2\}}}. \quad (23)$$

The optimization constraints can be derived from the terms in (23), and thus they are (see also [23] and [24]):

$$\begin{aligned} 0 &\leq t_0 \leq \kappa, \quad 0 \leq t_{\{0,1\}} \leq t_0, \\ t_{\{0,1\}} &\leq t_1 \leq \kappa - t_0 + t_{\{0,1\}}, \quad 0 \leq t_{\{0,1,2\}} \leq t_{\{0,1\}}, \\ t_{\{0,1,2\}} &\leq t_{\{0,2\}} \leq t_0 - t_{\{0,1\}} + t_{\{0,1,2\}}, \\ t_{\{0,1,2\}} &\leq t_{\{1,2\}} \leq t_1 - t_{\{0,1\}} + t_{\{0,1,2\}}, \\ t_{\{0,2\}} + t_{\{1,2\}} - t_{\{0,1,2\}} &\leq t_2 \\ &\leq \kappa - t_0 - t_1 + t_{\{0,1\}} + t_{\{0,2\}} + t_{\{1,2\}} - t_{\{0,1,2\}}, \\ \text{and } \lfloor 3\kappa/2 \rfloor &\leq t_0 + t_1 + t_2 \leq \lceil 3\kappa/2 \rceil. \end{aligned} \quad (24)$$

The solution of this optimization problem is not unique. However, since all the solutions have the same performance (e.g., they all achieve F_{sum}^* , see also [24]), we work with one of these solutions, and call it an optimal vector, $\mathbf{t}^*$.

V. CPO: CUSTOMIZATION FOR PR SYSTEMS

Using an optimal vector $\mathbf{t}^*$, computed as described in the previous section, $\mathbf{H}^p$ is partitioned and the protograph matrix of the SC code, $\mathbf{H}_{\text{SC}}^p$, is constructed. The next step is preventing as many objects in the protograph as possible from being reflected in the unlabeled graph of the SC code, via optimizing the circulant powers using the CPO. Here, the CPO is customized for the $(4, 4(\gamma - 2))$ object, which is the common substructure for detrimental configurations in the case of PR systems (see also Fig. 2).

From the previous analysis, a Pattern P_ℓ spans at most either $m + 1$ or $2m + 1$ consecutive replicas, depending on the value of ℓ . Let $\xi = 2m + 1$. Thus, in the CPO, it suffices to operate on the PM $\Pi_1^{\xi,p}$, which is the non-zero part of the first ξ replicas in $\mathbf{H}_{\text{SC}}^p$ and has the size $(\xi + m)\gamma \times \xi\kappa$. The circulant powers associated with the 1's in $\mathbf{H}^p$ are denoted by $f_{i,j}$, where $0 \leq i \leq \gamma - 1$ and $0 \leq j \leq \kappa - 1$. Let the circulant powers

associated with the 1's in $\Pi_1^{\xi,p}$ be $f'_{i',j'}$, where $0 \leq i' \leq (\xi + m)\gamma - 1$ and $0 \leq j' \leq \xi\kappa - 1$. From the repetitive nature of the PM $\Pi_1^{\xi,p}$, $f'_{i',j'} = f'_{\bar{i}',\bar{j}'}$, where $\bar{i}' = (i' \bmod \gamma)$ and $\bar{j}' = (j' \bmod \kappa)$. Define our cycle-8 candidate in the graph of $\Pi_1^{\xi,p}$ as $c_1 - v_1 - c_2 - v_2 - c_3 - v_3 - c_4 - v_4$, which is again a particular way of traversing a pattern and not necessarily a protograph cycle (see also Figures 2 and 3). This candidate results in z (or $z/2$ in the case of P_1 only) cycles of length 8 after lifting if and only if [33]:

$$\begin{aligned} f'_{c_1,v_1} + f'_{c_2,v_2} + f'_{c_3,v_3} + f'_{c_4,v_4} \\ \equiv f'_{c_1,v_2} + f'_{c_2,v_3} + f'_{c_3,v_4} + f'_{c_4,v_1} \pmod{z}. \end{aligned} \quad (25)$$

The goal is to prevent as many cycle-8 candidates in the graph of $\mathbf{H}_{\text{SC}}^p$ as possible from being converted into z (or $z/2$ in the case of P_1) $(4, 4(\gamma - 2))$ UASs/UTSs in the graph of $\mathbf{H}_{\text{SC}}$, which is the unlabeled graph of the SC code. A cycle-8 candidate in the graph of $\mathbf{H}_{\text{SC}}^p$ is allowed to be converted into multiple $(4, 4(\gamma - 2) - 2\delta)$ UASs/UTSs, with $\delta \in \{1, 2\}$, as long as they are not $(4, 0)$ UASs, in the unlabeled graph since these are not instances of the common substructure of interest. These $(4, 4(\gamma - 2) - 2\delta)$ UASs/UTSs, $\delta \in \{1, 2\}$, are cycles of length 8 with **internal connections**, which means v_1 and v_3 are adjacent or/and v_2 and v_4 are adjacent (see Fig. 2). For the cycle-8 candidate in the graph of $\Pi_1^{\xi,p}$ that is described in the previous paragraph and has a CN, say c_5 , connecting v_1 and v_3 , in order to have this internal connection in the lifted cycles, the following condition for a cycle of length 6 must be satisfied in addition to (25):

$$f'_{c_1,v_1} + f'_{c_2,v_2} + f'_{c_5,v_3} \equiv f'_{c_1,v_2} + f'_{c_2,v_3} + f'_{c_5,v_1} \pmod{z}. \quad (26)$$

Similarly, for that cycle-8 candidate in the graph of $\Pi_1^{\xi,p}$ that has a CN, say c_6 , connecting v_2 and v_4 , in order to have this internal connection in the lifted cycles, the following condition for a cycle of length 6 must be satisfied in addition to (25):

$$f'_{c_1,v_1} + f'_{c_6,v_2} + f'_{c_4,v_4} \equiv f'_{c_1,v_2} + f'_{c_6,v_4} + f'_{c_4,v_1} \pmod{z}. \quad (27)$$

Note that the two CNs, c_5 and c_6 , have to be different from the CNs the pattern encompasses in order that we consider them in the CPO algorithm as possible internal connections. The reason is that the final unlabeled graphs of our codes must have no cycles of length 4 (which is also why (25) is applied for P_1 since $f'_{c_1,v_1} + f'_{c_2,v_2} \equiv f'_{c_1,v_2} + f'_{c_2,v_1} \pmod{z}$ is not allowed for any protograph cycle of length 4, $c_1 - v_1 - c_2 - v_2$).

The following lemma discusses the internal connections for different patterns in the protograph.

Lemma 6: Let η_{P_ℓ} be the maximum number of internal connections Pattern P_ℓ can have (multiple internal connections between the same two VNs are only counted once). Then,

$$\eta_{P_\ell} = \begin{cases} 0, & \ell \in \{1, 3, 5\}, \\ 1, & \ell \in \{2, 6, 8\}, \\ 2, & \ell \in \{4, 7, 9\}. \end{cases} \quad (28)$$

Proof: A protograph pattern, P_ℓ , with only two VNs ($\ell \in \{1, 3, 5\}$) cannot have any internal connections. A protograph pattern with three VNs ($\ell \in \{2, 6, 8\}$) can have at most one internal connection. A protograph pattern with four VNs

Algorithm 1 Optimizing Circulant Powers of SC Codes for PR Systems

- 1: **Inputs:** SC code parameters $\gamma \geq 3$, κ , m , and $L \geq 2m+1$, in addition to an optimal vector $\mathbf{t}^*$.
- 2: Assign initial circulant powers to all the $\gamma\kappa$ 1's in $\mathbf{H}^P$. In this work, our initial powers are as in SCB codes. For example, $f_{i,j} = (i^2)(2j)$, $0 \leq i \leq \gamma-1$ and $0 \leq j \leq \kappa-1$. (initially, no cycles of length 4 are in $\mathbf{H}_{SC}$)
- 3: Construct $\Pi_1^{\xi,p}$ via $\mathbf{H}^P$ and $\mathbf{t}^*$. Circulant powers of the 1's in $\Pi_1^{\xi,p}$, $f'_{i',j'}$, are obtained from the 1's in $\mathbf{H}^P$.
- 4: Define a counting variable $\psi_{i,j}$, $0 \leq i \leq \gamma-1$ and $0 \leq j \leq \kappa-1$, for each of the 1's in $\mathbf{H}^P$.
- 5: Define another counting variable $\psi'_{i',j'}$, $0 \leq i' \leq (\xi+m)\gamma-1$ and $0 \leq j' \leq \xi\kappa-1$, for each of the elements in $\Pi_1^{\xi,p}$.
- 6: Initialize all the variables in Steps 4 and 5 with zeros. Only $\xi\gamma\kappa$ counting variables of the form $\psi'_{i',j'}$ are associated with 1's in $\Pi_1^{\xi,p}$. The other variables remain zeros all the time.
- 7: Locate all instances of the nine patterns in $\Pi_1^{\xi,p}$. Note that locating P_1 means also locating all cycles of length 4 in $\Pi_1^{\xi,p}$, which is needed.
- 8: Determine the ζ_{P_ℓ} ways to traverse each instance of P_ℓ , $\forall \ell$, to reach $(4, 4(\gamma-2))$ UASs/UTSs in the unlabeled graph, which are the ζ_{P_ℓ} cycle-8 candidates.
- 9: Specify all internal connections (CNs) in each candidate determined in Step 8 if they can exist.
- 10: **for** each cycle-8 candidate in $\Pi_1^{\xi,p}$ **do**
- 11: Check whether (25) is satisfied for its circulant powers or not.
- 12: **if** ((25) is satisfied and the candidate has no internal connections) **||** ((25) is satisfied and the candidate has internal connection(s) but neither (26) nor (27) is satisfied for any internal connection) **then**
- 13: Mark this cycle-8 candidate as an *active candidate*.
- 14: **end if**
- 15: **end for**
- 16: Define $F_{P_\ell,1}^{k,a}$, $k \in \{1, 2, \dots, \xi\}$, as the number of active candidates of P_ℓ starting at the first replica and spanning k consecutive replicas in $\Pi_1^{\xi,p}$.
- 17: The number of active candidates of P_ℓ spanning k consecutive replicas in $\Pi_1^{\xi,p}$ is $(\xi-k+1)F_{P_\ell,1}^{k,a}$. (for example, for $k=1$, $\xi F_{P_\ell,1}^{1,a}$ is the number of active candidates of P_ℓ , for any value of ℓ , spanning one replica in $\Pi_1^{\xi,p}$)
- 18: Compute the number of $(4, 4(\gamma-2))$ UASs/UTSs in $\mathbf{H}_{SC}$ using the following formula (see also [23]):

$$F_{SC} = \sum_{\ell=1}^9 \sum_{k=1}^{\xi} \left((L-k+1) F_{P_\ell,1}^{k,a} \right) z_{P_\ell}, \quad (29)$$

where $z_{P_\ell} = z/2$ if $\ell = 1$, and $z_{P_\ell} = z$ otherwise. Recall that $\xi = 2m+1$.

- 19: Initialize the flag stop_iterating with 0.
- 20: **while** stop_iterating = 0 **do**
- 21: Determine all the active candidates each 1 in $\Pi_1^{\xi,p}$ is involved in.

- 22: Assign weight $w_k = (L-k+1)/(\xi-k+1)$ to the number of active candidates from Step 21 that span k consecutive replicas in $\Pi_1^{\xi,p}$ (see also [23]). (for example, for $k=\xi$, the weight of the number of active candidates spanning ξ consecutive replicas is $(L-\xi+1)$)
- 23: Multiply w_k by $1/2$ if the candidate is associated with P_1 . Then, compute the weighted sum.
- 24: Store this weighted count associated with each 1 in $\Pi_1^{\xi,p}$, which is indexed by (i', j') , in $\psi'_{i',j'}$.
- 25: Calculate the counting variables $\psi_{i,j}$, $\forall i, j$, associated with the 1's in $\mathbf{H}^P$ from the counting variables $\psi'_{i',j'}$ associated with the 1's in $\Pi_1^{\xi,p}$ (computed in Steps 21 through 24) using the following formula:

$$\psi_{i,j} = \sum_{i':i'=i} \sum_{\substack{j':j'=j \\ \Pi_1^{\xi,p}[i'][j'] \neq 0}} \psi'_{i',j'}. \quad (30)$$

- 26: Sort the $\gamma\kappa$ 1's of $\mathbf{H}^P$ in a list descendingly according to the counts in $\psi_{i,j}$, $\forall i, j$.
- 27: Initialize the flag update_occurred with 0.
- 28: **while** update_occurred = 0 **do**
- 29: Pick a subset of 1's from the top of this list, and change the circulant powers associated with them. (each time, a different set of circulant powers or/and a different subset of 1's (from the 1's in $\mathbf{H}^P$) is picked)
- 30: Using these interim powers, do Steps 10 through 18.
- 31: **if** F_{SC} is reduced while maintaining no cycles of length 4 and no $(4, 0)$ objects (in the case of $\gamma=3$) in $\mathbf{H}_{SC}$ **then**
- 32: Update F_{SC} and the circulant powers.
- 33: Set the flag update_occurred to 1. (this is to exit the current while loop)
- 34: **end if**
- 35: **end while**
- 36: **if** (the target F_{SC} (set by the code designer) is achieved) **||** (the reduction in F_{SC} approaches zero) **then**
- 37: Set the flag stop_iterating to 1. (this is to exit the current while loop)
- 38: **end if**
- 39: **end while**
- 40: Assign the optimized circulant powers to the 1's in $\mathbf{H}^P$ to construct $\mathbf{H}$. Then, partition $\mathbf{H}$ (using $\mathbf{t}^*$) and couple L times to construct $\mathbf{H}_{SC}$.
- 41: **Output:** Binary parity-check matrix $\mathbf{H}_{SC}$ with optimized circulant powers.

($\ell \in \{4, 7, 9\}$) can have up to two internal connection, which completes the proof. ■

The case of multiple internal connections between the same two VNs is addressed in the CPO algorithm.

Algorithm 1 is the customized CPO algorithm for SC codes designed for PR systems.

Step 29 in the CPO algorithm is performed heuristically. The number of 1's to work with depends on the circulant size, the values of the counts, and how these values are distributed. Moreover, tracking the counts of active candidates and the

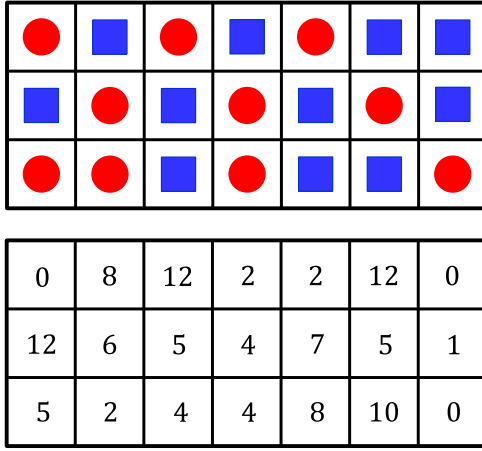


Fig. 7. Upper panel: the OO partitioning of $\mathbf{H}^p$ (or $\mathbf{H}$) of the SC code in Example 3. Entries with circles (resp., squares) are assigned to $\mathbf{H}_0^p$ (resp., $\mathbf{H}_1^p$). Lower panel: the circulant power arrangement for the circulants in $\mathbf{H}$.

distribution of their values over different 1's in $\mathbf{H}^p$ is the main factor to decide which 1's to select in each iteration.

The reduction in the number of detrimental objects achieved by the CPO algorithm depends on the value of the circulant size z relative to the row weight κ of the underlying block code. In particular, as the ratio z/κ increases, more percentage reduction is achievable. Increasing the column weight γ results in less percentage reduction. Increasing the memory m results in more percentage reduction. We note that the CPO algorithm always reaches a solution for the circulant power arrangement that results in less (worst case the same) number of detrimental objects compared with the initial arrangement.

Example 3: Suppose we are designing an SC code with $\gamma = 3$, $\kappa = 7$, $z = 13$, $m = 1$, and $L = 10$ using the OO-CPO approach for PR systems. Solving the optimization problem in (22) gives an optimal vector $\mathbf{t}^* = [t_0^* \ t_1^* \ t_2^* \ t_{\{0,1\}}^* \ t_{\{0,2\}}^* \ t_{\{1,2\}}^* \ t_{\{0,1,2\}}^*]^T = [3 \ 3 \ 4 \ 0 \ 1 \ 2 \ 0]^T$, with $F_{\text{sum}}^* = 5,170$ patterns (rounded weighted sum) in the graph of $\mathbf{H}_{\text{SC}}^p$. Fig. 7, upper panel, shows how the partitioning is applied on $\mathbf{H}^p$ (or $\mathbf{H}$). Next, applying the CPO results in 2,613 (4,4) UASs in the graph of $\mathbf{H}_{\text{SC}}$. Fig. 7, lower panel, shows the final circulant power arrangement for all circulants in $\mathbf{H}$.

Remark 5: After introducing the concept of patterns in this work, the OO-CPO approach can be extended to target other common substructures if needed.

VI. EXPERIMENTAL RESULTS

In this section, we propose experimental results demonstrating the effectiveness of the OO-CPO approach compared with other code design techniques in PR (1-D MR) systems.

Remark 6: In this section, all the codes used have no cycles of length 4. The codes with $\gamma = 3$ (resp., $\gamma = 4$) have a minimum distance of at least 6 (resp., 8). Moreover, we opted to work with circulant sizes $z > \kappa$ in order to give more freedom to the CPO, which results in less detrimental objects.

First, we compare the total number of instances of the common substructure of interest in the unlabeled graphs of SC

TABLE III
NUMBER OF (4, 4) UASs IN SC CODES WITH $\gamma = 3$, $m \in \{1, 2\}$, AND $L = 10$ DESIGNED USING DIFFERENT TECHNIQUES

Design technique	Number of (4, 4) UASs			
	$\kappa = 7$, $z = 13$	$\kappa = 11$, $z = 23$	$\kappa = 13$, $z = 29$	$\kappa = 17$, $z = 37$
Uncoupled with SCB	32,370	254,610	540,850	1,700,890
SC CV with SCB and $m = 1$	9,464	91,333	197,084	652,347
SC OO with SCB and $m = 1$	6,500	53,130	123,395	440,818
SC OO-CPO and $m = 1$	2,613	32,361	70,151	254,005
SC OO with SCB and $m = 2$	3,172	27,508	60,233	194,176
SC OO-CPO and $m = 2$	819	13,110	32,074	117,697

TABLE IV
NUMBER OF (4, 8) UTSS IN SC CODES WITH $\gamma = 4$, $m = 1$, AND $L = 10$ DESIGNED USING DIFFERENT TECHNIQUES

Design technique	Number of (4, 8) UTSS			
	$\kappa = 7$, $z = 13$	$\kappa = 11$, $z = 23$	$\kappa = 13$, $z = 29$	$\kappa = 17$, $z = 37$
Uncoupled with SCB	131,820	1,034,310	2,193,850	7,081,430
SC CV with SCB	48,074	396,474	843,233	2,782,844
SC OO with SCB	27,729	230,230	508,544	1,667,886
SC OO-CPO	17,095	165,071	366,212	1,253,745

codes designed using various techniques. We present results for two groups of codes.

All the codes in the first group have $\gamma = 3$ (i.e., the common substructure of interest is the (4,4) UAS in Fig. 2) and $m \in \{1, 2\}$. We also choose $L = 10$ for this group. In addition to the uncoupled setting ($\mathbf{H}_0 = \mathbf{H}$ and $\mathbf{H}_1 = \mathbf{0}$), we show results for the following five SC code design techniques:

- 1) The CV technique (see [16]) with $m = 1$.
- 2) The OO technique with no CPO applied and with $m = 1$.
- 3) The OO technique with circulant powers optimized via the CPO (the OO-CPO approach) and with $m = 1$.
- 4) The OO technique with no CPO applied and with $m = 2$.
- 5) The OO technique with circulant powers optimized via the CPO (the OO-CPO approach) and with $m = 2$.

In the uncoupled setting in addition to the first, second, and fourth techniques, circulant powers as in SCB codes, $f_{i,j} = (i^2)(2j)$, are used. This choice of circulant powers guarantees no cycles of length 4.

The results of the first group of codes for different choices of κ and z are listed in Table III.⁴ For a particular choice of κ , z , m , and L , SC codes designed using these different techniques all have block length $= \kappa z L \log_2(q)$ bits and rate $\approx [1 - 3(L + m)/(\kappa L)]$. Table III demonstrates the significant gains achieved by the OO-CPO approach compared with other techniques. In particular, for $m = 1$, the proposed OO-CPO approach achieves a reduction in the number of

⁴In Tables III and IV, “with SCB” means “with SCB circulant powers”. Similarly, in Figures 9 and 13, “w SCB” means “with SCB circulant powers”.

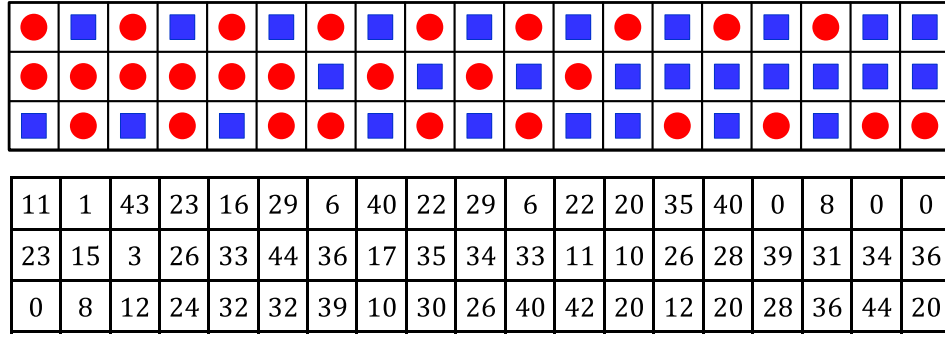


Fig. 8. Upper panel: the OO partitioning of $\mathbf{H}^P$ (or $\mathbf{H}$) of Code 3. Entries with circles (resp., squares) are assigned to $\mathbf{H}_0^P$ (resp., $\mathbf{H}_1^P$). Lower panel: the circulant power arrangement for the circulants in $\mathbf{H}$ of Code 3.

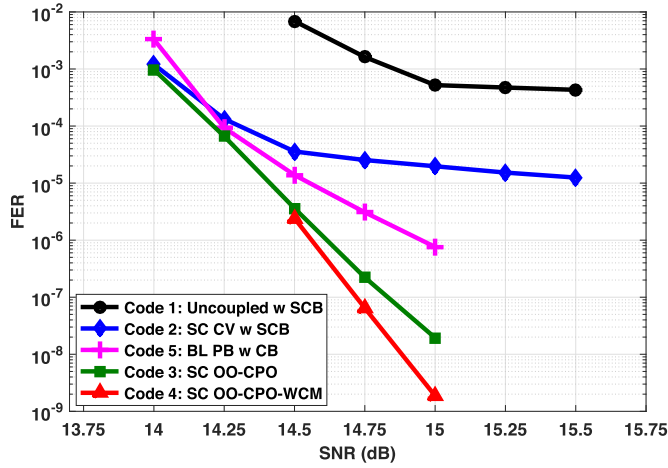


Fig. 9. Simulation results over the PR channel for SC codes having $\gamma = 3$ and $m = 1$ designed using different techniques and a BL code of the same length and rate. Significant gains are achieved by the OO-CPO approach (the plots of Codes 3 and 4).

(4, 4) UASs that ranges between 85% and 92% compared with the uncoupled setting, and between 61% and 72% compared with the CV technique. The table also illustrates the positive effect of increasing the memory of the SC code. In particular, the OO-CPO approach with $m = 2$ achieves a reduction in the number of (4, 4) UASs that ranges between 54% and 69% compared with the OO-CPO approach with $m = 1$. Moreover, the importance of the two stages (the OO and the CPO) is highlighted by the numbers in Table III.

As for the second group, all the codes have $\gamma = 4$ (i.e., the common substructure of interest is the (4, 8) UTS in Fig. 2) and $m = 1$. We also choose $L = 10$ for this group. In addition to the uncoupled setting ($\mathbf{H}_0 = \mathbf{H}$ and $\mathbf{H}_1 = \mathbf{0}$), we show results for the following three SC code design techniques:

- 1) The CV technique (see [16]).
- 2) The OO technique with no CPO applied.
- 3) The OO technique with circulant powers optimized via the CPO (the OO-CPO approach).

In the uncoupled setting in addition to the first and second techniques, circulant powers as in SCB codes, $f_{i,j} = (i^2)(2j)$, are used. This choice of circulant powers guarantees no cycles of length 4.

The results of the second group of codes for different choices of κ and z are listed in Table IV. For a particular choice of κ , z , and L , SC codes designed using these different techniques all have block length $= \kappa z L \log_2(q)$ bits and rate $\approx [1 - 4(L + 1)/(\kappa L)]$. Table IV again demonstrates the significant gains achieved by the OO-CPO approach compared with other techniques. In particular, the proposed OO-CPO approach achieves a reduction in the number of (4, 8) UTSs that ranges between 82% and 87% compared with the uncoupled setting, and between 55% and 64% compared with the CV technique. Moreover, the importance of the two stages (the OO and the CPO) is again highlighted by the numbers in Table IV.

Additionally, we construct SC codes through the OO technique only and pseudo random circulant powers such that the codes have no cycles of length 4. Table V shows the comparison between these codes, whose design approach is called OO-Rand, with the codes designed using the OO-CPO approach. In the case of $\gamma = 3$ and $m = 1$, the reduction in the number of (4, 4) UASs achieved by the OO-CPO approach compared with the OO-Rand approach ranges between 16% and 56%. In the case of $\gamma = 3$ and $m = 2$, this reduction ranges between 29% and 61%. In the case of $\gamma = 4$ and $m = 1$, this reduction ranges between 10% and 27%. These numbers further demonstrate the importance of the CPO stage.

Second, we present simulation results of binary and non-binary SC codes designed using various techniques over the PR channel. We present results for three groups of codes. We use the PR channel described in Section II. In the simulations, we reach FER levels as low as $\approx 10^{-9}$ in order to demonstrate the suitability of the designed codes to MR applications. Levels below 10^{-12} can only be reached via Field Programmable Gate Array (FPGA) simulations.

The first group of simulated codes contains five different codes. All the five codes are defined over $\text{GF}(4)$. Codes 1, 2, 3, and 4 have $\gamma = 3$, $\kappa = 19$, $z = 46$, $m = 1$, and $L = 5$. Thus, these codes have block length $= 8,740$ bits, and the SC codes have rate ≈ 0.81 . Code 1 is uncoupled. Code 2 is an SC code designed using the CV technique for PR channels as described in [16]. The optimal cutting vector used for Code 2 is [4 9 15]. Codes 1 and 2 have SCB circulant powers of the form $f_{i,j} = (i^2)(2j)$. Code 3 is an SC code designed using the OO-CPO approach. The partitioning and

TABLE V

NUMBER OF $(4, 4(\gamma - 2))$ UASs/UTSs IN SC CODES WITH $L = 10$ DESIGNED USING THE OO TECHNIQUE WITH RANDOM CIRCULANT POWERS VERSUS THE OO-CPO APPROACH

Values of γ and m	Number of $(4, 4(\gamma - 2))$ UASs/UTSs							
	$\kappa = 7, z = 13$		$\kappa = 11, z = 23$		$\kappa = 13, z = 29$		$\kappa = 17, z = 37$	
	OO-Rand	OO-CPO	OO-Rand	OO-CPO	OO-Rand	OO-CPO	OO-Rand	OO-CPO
$\gamma = 3, m = 1$	5,928	2,613	42,849	32,361	86,884	70,151	300,662	254,005
$\gamma = 3, m = 2$	2,093	819	24,932	13,110	54,578	32,074	165,390	117,697
$\gamma = 4, m = 1$	23,309	17,095	203,274	165,071	437,146	366,212	1,395,973	1,253,745

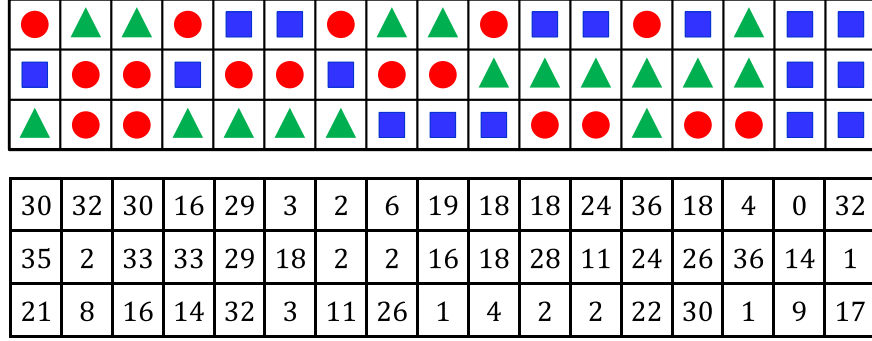


Fig. 10. Upper panel: the OO partitioning of $\mathbf{H}^P$ (or $\mathbf{H}$) of Code 6. Entries with circles (resp., squares and triangles) are assigned to $\mathbf{H}_0^P$ (resp., $\mathbf{H}_1^P$ and $\mathbf{H}_2^P$). Lower panel: the circulant power arrangement for the circulants in $\mathbf{H}$ of Code 6.

the circulant power arrangement of Code 3 are given in Fig. 8. Codes 1, 2, and 3 have unoptimized edge weights.⁵ Code 4 is the result of applying the WCM framework to Code 3 in order to optimize its edge weights. The numbers of $(4, 4)$ UASs in the unlabeled graphs of Codes 1, 2, and 3 are 2,425,120, 845,434, and 184,667, respectively. Code 5 is a block (BL) code, which is also protograph-based (PB), designed as in [11] and [12]. Code 5 has column weight = 3, circulant size = 46, block length = 8,832 bits, rate ≈ 0.81 (same as all SC codes), and unoptimized weights (similar to all codes except Code 4).

Fig. 9 demonstrates the effectiveness of the proposed OO-CPO approach in designing high performance SC codes for PR channels. In particular, Code 3 (designed using the OO-CPO approach) outperforms Code 2 (designed using the CV technique) by about 3 orders of magnitude at SNR = 15 dB, and by about 1.1 dB at FER = 10^{-5} . More intriguingly, Code 3 outperforms Code 5 (the block code) by about 1.6 orders of magnitude at SNR = 15 dB, and by almost 0.4 dB at FER = 10^{-6} . The performance of Code 3 is better than the performance of Code 5 not only in the error floor region, but also in the waterfall region. An interesting observation is that in the error profile of Code 3, we found no codewords of weights $\in \{6, 8\}$ (which are $(6, 0, 0, 9, 0)$ and $(8, 0, 0, 12, 0)$ BASTs) despite the dominant presence of such low weight codewords in the error profiles of Codes 1, 2, and 5 (see also [5], [12], and [16]). This indicates a significant improvement in the minimum distance properties of the code designed using the OO-CPO approach. From Fig. 9, the WCM framework achieves 1 order of magnitude additional gain.

An important reason behind the improved waterfall performance of Code 3 is the significant reduction in the

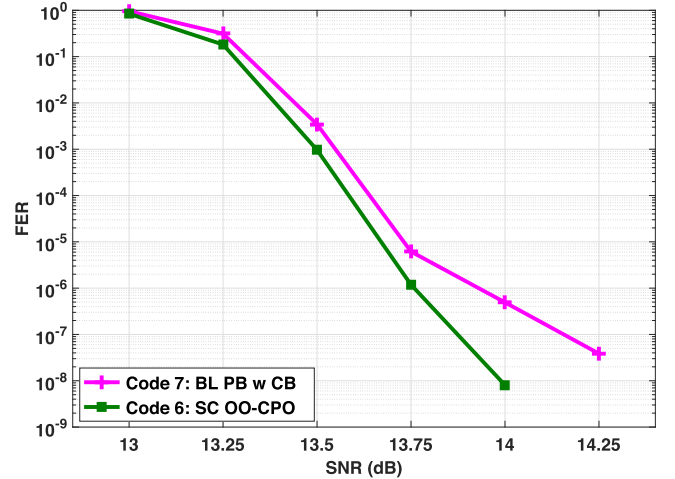


Fig. 11. Simulation results over the PR channel for an SC code having $\gamma = 3$ and $m = 2$ designed using the OO-CPO approach and a BL code of the same length and rate. Significant gains are achieved by the OO-CPO approach (the plot of Code 6).

multiplicity of low weight codewords achieved by the OO-CPO approach. This reduction is a result of the fact that such low weight codewords also have the $(4, 4)$ UAS as a common substructure in their configurations (see Fig. 2). More details about minimum distance analysis of SC codes can be found in [35] and [36].

The second group of simulated codes contains two different codes. The two codes are defined over GF(4). Code 6 has $\gamma = 3, \kappa = 17, z = 37, m = 2$, and $L = 7$. Thus, this code has block length = 8,806 bits and rate ≈ 0.77 . Code 6 is an SC code designed using the OO-CPO approach. The partitioning and the circulant power arrangement of Code 6 are given in Fig. 10. Code 6 has unoptimized edge weights. The number of $(4, 4)$ UASs in the unlabeled graph of Code 6 is reduced to

⁵Codes 1 and 2 are the two codes used in Example 1.

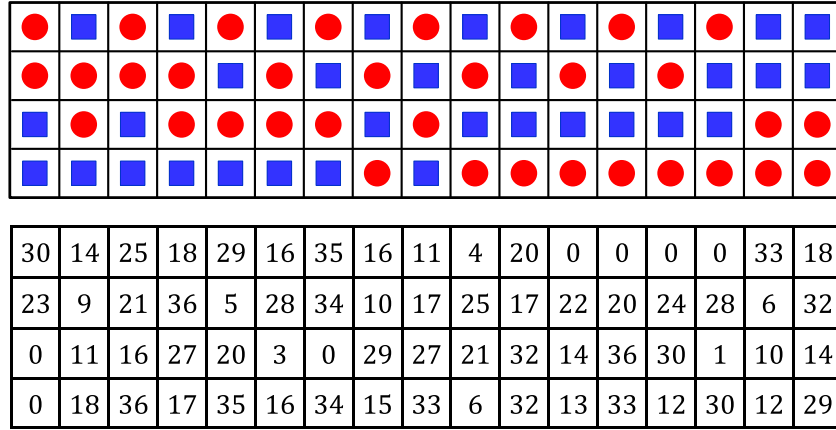


Fig. 12. Upper panel: the OO partitioning of $\mathbf{H}^P$ (or $\mathbf{H}$) of Code 10. Entries with circles (resp., squares) are assigned to $\mathbf{H}_0^P$ (resp., $\mathbf{H}_1^P$). Lower panel: the circulant power arrangement for the circulants in $\mathbf{H}$ of Code 10.

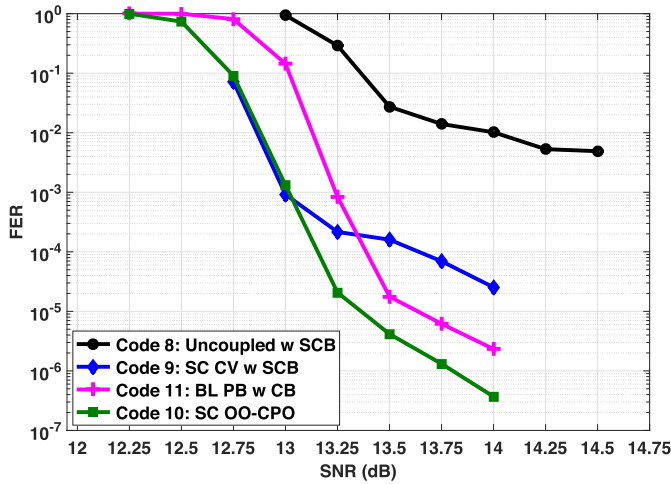


Fig. 13. Simulation results over the PR channel for SC codes having $\gamma = 4$ and $m = 1$ designed using different techniques and a BL code of the same length and rate. Significant gains are achieved by the OO-CPO approach (the plot of Code 10).

75,850 via the OO-CPO approach. Code 7 is a BL PB code designed as in [11] and [12]. Code 7 has column weight = 3, circulant size = 43, block length = 8,944 bits, rate ≈ 0.77 (same as the SC code), and unoptimized weights (similar to the SC code).

The purpose of Fig. 11 is to stress on the intriguing conclusion that SC codes designed using the OO-CPO approach outperform block codes having the same parameters. In particular, Code 6 (designed using the OO-CPO approach) outperforms Code 7 (the block code) by about 1.8 orders of magnitude at SNR = 14 dB, and by about 0.3 dB at FER = 10^{-7} . These gains are projected to be significantly bigger as we go deeper in FER noting that we could not collect a single error after simulating around 10^9 frames of Code 6 at SNR = 14.25 dB. Moreover, the performance of Code 6 is better than the performance of Code 7 not only in the error floor region, but also in the waterfall region.

The third group of simulated codes contains four different codes. All the four codes are defined over GF(2), i.e., binary

codes. Codes 8, 9, and 10 have $\gamma = 4$, $\kappa = 17$, $z = 37$, $m = 1$, and $L = 6$. Thus, these codes have block length = 3,774 bits, and the SC codes have rate ≈ 0.73 . Code 8 is uncoupled. Code 9 is an SC code designed using the CV technique for PR channels as described in [16]. The optimal cutting vector used for Code 9 is [3 7 11 14]. Codes 8 and 9 have SCB circulant powers of the form $f_{i,j} = (i^2)(2j)$. Code 10 is an SC code designed using the OO-CPO approach. The partitioning and the circulant power arrangement of Code 10 are given in Fig. 12. The numbers of (4,8) UTSs in the unlabeled graphs of Codes 8, 9, and 10 are 4,248,858, 1,589,816, and 705,849, respectively. Code 11 is a BL PB code designed as in [11] and [12]. Code 11 has column weight = 4, circulant size = 41, block length = 3,690 bits and rate ≈ 0.73 (same as all SC codes).

Fig. 13 again demonstrates the effectiveness of the OO-CPO approach in designing high performance SC codes with various parameters for PR channels. In particular, Code 10 (designed using the OO-CPO approach) outperforms Code 9 (designed using the CV technique) by more than 1.8 orders of magnitude at SNR = 14 dB, and by nearly 0.75 dB at FER = 3×10^{-5} . More intriguingly, Code 10 outperforms Code 11 (the block code) by about 0.8 of an order of magnitude at SNR = 14 dB, and by about 0.35 dB at FER = 3×10^{-6} . A very interesting observation here is that Code 10 achieves an early waterfall gain of about 0.25 dB compared with Code 11 (see, for example, the performance of the two codes at FER = 10^{-1}). In other words, Code 10 has a 0.25 dB threshold improvement compared with Code 11. Note that in the error profile of Code 10, we found no (6, 2, 2, 11, 0), no (8, 0, 0, 16, 0), no (8, 2, 2, 15, 0), and no (8, 4, 4, 14, 0) BASTs despite the dominant presence of such objects in the error profiles of Codes 8, 9, and 11 (see also [12], and [16]). Note also that the codes here have a relatively low rate, which demonstrates the gains achieved by the OO-CPO approach for a diverse range of rates.

There are two key takeaways from these experimental results. First, SC codes constructed using the proposed OO-CPO approach significantly outperform SC codes

constructed using the techniques currently available in the literature. Second, and most important, appropriately exploiting the additional degree of freedom provided by partitioning in the construction of SC codes enables designing SC codes that outperform block codes of the same total length and rate, which conclusively answers an open question about whether SC codes can outperform block codes under equal total length and rate. This proper exploitation is performed exclusively through taking into account the characteristics of the channel of interest, which is what we do in this work.

Remark 7: Unlike a lot of literature works that compare an SC code to a block code having a length equal to the constraint length of the SC code, which is $\kappa z(m+1)\log_2(q)$ bits, we compare an SC code to a block code having the same length of the SC code in total, which is $\kappa zL\log_2(q)$ bits, approximately. Moreover, while our high performance SC codes designed using the OO-CPO approach do outperform block codes of the same parameters, other SC codes available in the literature do not. An example demonstrating this statement is presented in Fig. 9 and Fig. 13, where block codes outperform SC codes designed using the CV technique and of the same parameters.

Remark 8: Because our main focus in this work is the performance, a relatively small to average values of L ($5 \leq L \leq 7$) along with block decoding are used for all SC Codes.

VII. CONCLUSION

We proposed the OO-CPO approach to optimally design binary and non-binary SC codes for PR channels, via minimizing the number of detrimental objects in the graph of the code. A common substructure was first identified in the graphs of the detrimental configurations in the case of PR systems. We graphically determined the protograph patterns that are capable of generating instances of this common substructure in the final graph of the code. Next, through combinatorial techniques, we built a discrete optimization problem in which the weighted sum of the total number of instances of these patterns is expressed in terms of the partitioning parameters. The partitioning that achieves the minimum weighted sum was obtained. Then, the lifting parameters were optimized in order to achieve more reduction in the number of detrimental objects of interest. SC codes designed using the proposed OO-CPO approach were shown to significantly outperform SC codes designed using techniques from the literature over PR channels. More importantly, our channel-aware combinatorial approach demonstrated that appropriate exploitation of the available degrees of freedom in the SC code design can give SC codes significant performance advantages over structured block codes having the same parameters. We believe this research will open the door for engineers to deploy high performance SC codes in a wide variety of applications in addition to data storage. Possible future directions include theoretical analysis of how the multiplicities of different BASs affect the SC code performance in PR systems in addition to minimum distance analysis of SC codes designed using the OO-CPO approach.

APPENDIX A PROOFS OF PATTERN P_1

A. Proof of Lemma 3

Proof: In Case 1.1, the number of instances is the number of ways to choose 2 overlaps out of $t_{\{i_1, i_2\}}$ overlaps (the pattern has two $c_1 - c_2$ overlaps), which is given by (7). In Case 1.2, the number of instances is the number of ways to choose 1 overlap out of $t_{\{i_1, i_2\}}$ and 1 overlap out of $t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}$, which is given by (8). ■

B. Proof of Theorem 1

Proof: To compute F_{P_1} , we use Formula (6) with χ , which is the maximum number of replicas the pattern can span, equals $m+1$. Since the overlaps of P_1 can exist in up to 2 replicas, we need to find expressions only for $F_{P_1,1}^1$ (overlaps are in 1 replica) and $F_{P_1,1}^{k \geq 2}$ (overlaps are in 2 replicas).

Then, $F_{P_1,1}^1$ is the sum of function $\mathcal{A}_{P_1}$ in (7), with $r=1$, over all possible values of $\{i_1, i_2\}$. Here, $\{i_1, i_2\}$ can take any distinct two values in the range from the start to the end of $\mathbf{R}_1$, i.e., from 0 to $(m+1)\gamma-1$ (see Fig. 4).

Moreover, $F_{P_1,1}^{k \geq 2}$ is the sum of function $\mathcal{B}_{P_1}$ in (8), with $r=1$ and $e=k$, over all possible values of $\{i_1, i_2\}$. Here, $\{i_1, i_2\}$ can take any distinct two values in the range from the start of $\mathbf{R}_k$ to the end of $\mathbf{R}_1$, i.e., from $(k-1)\gamma$ to $(m+1)\gamma-1$ (see also Fig. 4). ■

APPENDIX B PROOFS OF PATTERN P_2

A. Proof of Lemma 4

Proof: In Case 2.1, the number of instances is the number of ways to choose 3 overlaps out of $t_{\{i_1, i_2\}}$ overlaps (the pattern has three $c_1 - c_2$ overlaps), which is given by (11). In Case 2.2, the number of instances is the number of ways to choose 2 overlap out of $t_{\{i_1, i_2\}}$ and 1 overlap out of $t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}$, which is given by (12). In Case 2.3, the number of instances is the number of ways to choose 1 overlap out of $t_{\{i_1, i_2\}}$, 1 overlap out of $t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma\}}$, and 1 overlap out of $t_{\{i_1+(r-s)\gamma, i_2+(r-s)\gamma\}}$, which is given by (13). ■

B. Proof of Theorem 2

Proof: To compute F_{P_2} , we use Formula (6) with $\chi = m+1$. Since the overlaps of P_2 can exist in up to 3 replicas, we need to find expressions only for $F_{P_2,1}^1$, $F_{P_2,1}^2$, and $F_{P_2,1}^{k \geq 3}$.

Then, $F_{P_2,1}^1$ is the sum of function $\mathcal{A}_{P_2}$ in (11), with $r=1$, over all possible values of $\{i_1, i_2\}$. Here, $\{i_1, i_2\}$ can take any distinct two values in the range from the start to the end of $\mathbf{R}_1$, i.e., from 0 to $(m+1)\gamma-1$ (see Fig. 5).

Regarding $F_{P_2,1}^2$, we need to distinguish between two situations; when $r < e$ (i.e., replica $\mathbf{R}_r$, which has two overlaps, comes before replica $\mathbf{R}_e$), and when $r > e$ (i.e., replica $\mathbf{R}_r$ comes after replica $\mathbf{R}_e$). This distinction gives the two summations of function $\mathcal{B}_{P_2}$ in $F_{P_2,1}^2$. For the first summation, $\mathcal{B}_{P_2}$ in (12) has $r=1$ and $e=2$. Thus, $\{i_1, i_2\}$ can take any distinct two values in the range from the start of $\mathbf{R}_2$ to the end of $\mathbf{R}_1$, i.e., from γ to $(m+1)\gamma-1$ (see Fig. 5 for more illustration). For the second summation, $\mathcal{B}_{P_2}$ in (12) has $r=2$ and $e=1$. Thus, $\{i_1, i_2\}$ can take any distinct two values in

the range from the start of $\mathbf{R}_2$ (which is now $\mathbf{R}_r$) to the end of $\mathbf{R}_1$, i.e., from 0 to $m\gamma - 1$.

As for $F_{P_2,1}^{k \geq 3}$, the overlaps can be in 2 replicas (the first two summations in $F_{P_2,1}^{k \geq 3}$) or 3 replicas (the third summation in $F_{P_2,1}^{k \geq 3}$). The first two summations are derived in a way similar to what we did for $F_{P_2,1}^2$, with a change in the summation indices; $\mathbf{R}_2$ is replaced by $\mathbf{R}_k$ here. For the third (double) summation, $\mathcal{C}_{P_2}$ in (13) has $r = 1$, $e = h$, and $s = k$. Thus, $\{i_1, i_2\}$ can take any distinct two values in the range from the start of $\mathbf{R}_k$ to the end of $\mathbf{R}_1$, i.e., from $(k-1)\gamma$ to $(m+1)\gamma - 1$ (see Fig. 5). The outer summation is over all possible values of h , and we have $1 < h < k$. ■

APPENDIX C

PROOFS OF PATTERN P_6

A. Proof of Lemma 5

Proof: In Case 6.1, the number of instances is the number of ways to choose 1 overlap from each family in $\mathbf{R}_r$ (there exist three different families for P_6 ; $c_1 - c_2 - c_3$, $c_1 - c_2$, and $c_1 - c_3$). We choose the $c_1 - c_2 - c_3$ degree-3 overlap first. Then, in order to avoid over-counting, it is required to distinguish between the two situations when the $c_1 - c_2$ degree-2 overlap is part of a $c_1 - c_2 - c_3$ degree-3 overlap, and when this is not the case. Taking this requirement into account yields the two added terms in (16). The same applies for Case 6.2, with the exception that here the degree-3 overlap is chosen from $t_{\{i_1+(r-e)\gamma, i_2+(r-e)\gamma, i_3+(r-e)\gamma\}}$ overlaps, resulting in (17). Following the same logic of Case 6.1 for Case 6.3, with the exception that the $c_1 - c_3$ overlap is chosen from $t_{\{i_1+(r-e)\gamma, i_3+(r-e)\gamma\}}$ overlaps, gives (18). In Case 6.4, the number of instances is the number of ways to choose 1 overlap out of $t_{\{i_1, i_2\}}$, 1 overlap out of $t_{\{i_1+(r-e)\gamma, i_3+(r-e)\gamma\}}$, and 1 overlap out of $t_{\{i_1+(r-s)\gamma, i_2+(r-s)\gamma, i_3+(r-s)\gamma\}}$, which is given by (19). ■

B. Proof of Theorem 3

Proof: To compute F_{P_6} , we use Formula (6) with $\chi = m + 1$. Since the overlaps of P_6 can exist in up to 3 replicas, we need to find expressions only for $F_{P_6,1}^1$, $F_{P_6,1}^2$, and $F_{P_6,1}^{k \geq 3}$.

Then, $F_{P_6,1}^1$ is the sum of function $\mathcal{A}_{P_6}$ in (16), with $r = 1$, over all possible values of i_1 and $\{i_2, i_3\}$. In Pattern P_6 , CN c_1 , which connects all three VNs, is different from the other two CNs. Moreover, in a group of three CNs that can form P_6 , c_1 can be any one of these three CNs, which means we have three possible ways to form P_6 from these three CNs. These facts combined are the reason why i_1 of c_1 has to be separated from $\{i_2, i_3\}$, despite having the same range, in the expression of $F_{P_6,1}^1$ (this applies for other expressions too). Here, i_1 (resp., $\{i_2, i_3\}$) can take any value (resp., distinct two values) in the range from the start to the end of $\mathbf{R}_1$, i.e., from 0 to $(m+1)\gamma - 1$ (see also Fig. 6).

Regarding $F_{P_6,1}^2$, we need to account for Case 6.2 and Case 6.3. For each of the two cases, we need to distinguish between two situations; when $r < e$ and when $r > e$. This distinction gives the two summations of $\mathcal{B}_{P_6}$ and the two summations of $\mathcal{C}_{P_6}$ in $F_{P_6,1}^2$. In Case 6.2, each of the three CNs of P_6 connects overlaps in $\mathbf{R}_r$ and $\mathbf{R}_e$ (because the degree-3

overlap is moved to $\mathbf{R}_e$). For the first summation in $F_{P_6,1}^2$, $\mathcal{B}_{P_6}$ in (17) has $r = 1$ and $e = 2$. Thus, i_1 (resp., $\{i_2, i_3\}$) can take any value (resp., distinct two values) in the range from the start of $\mathbf{R}_2$ to the end of $\mathbf{R}_1$, i.e., from γ to $(m+1)\gamma - 1$. For the second summation, $\mathcal{B}_{P_6}$ in (17) has $r = 2$ and $e = 1$. Thus, i_1 (resp., $\{i_2, i_3\}$) can take any value (resp., distinct two values) in the range from 0 to $m\gamma - 1$. In Case 6.3, and as shown in Fig. 6, c_1 and c_3 each connects overlaps in $\mathbf{R}_r$ and $\mathbf{R}_e$, while c_2 connects overlaps in $\mathbf{R}_r$ only (because the $c_1 - c_3$ overlap is moved to $\mathbf{R}_e$ here). For the third summation in $F_{P_6,1}^2$, $\mathcal{C}_{P_6}$ in (18) has $r = 1$ and $e = 2$. Thus, i_1 (resp., i_2 and i_3) can take any value in the range from the start of $\mathbf{R}_2$ (resp., $\mathbf{R}_1$ and $\mathbf{R}_2$) to the end of $\mathbf{R}_1$, i.e., from γ (resp., 0 and γ) to $(m+1)\gamma - 1$. For the fourth summation, $\mathcal{C}_{P_6}$ in (18) has $r = 2$ and $e = 1$ (see Fig. 6). Thus, i_1 (resp., i_2 and i_3) can take any value in the range from the start of $\mathbf{R}_2$ to the end of $\mathbf{R}_1$ (resp., $\mathbf{R}_2$ and $\mathbf{R}_1$), i.e., from 0 to $m\gamma - 1$ (resp., $(m+1)\gamma - 1$ and $m\gamma - 1$). Note that the ranges of i_2 and i_3 are different in Case 6.3, unlike Case 6.2, which is the reason why i_2 and i_3 are not in a set in the summations of $\mathcal{C}_{P_6}$.

As for $F_{P_6,1}^{k \geq 3}$, the overlaps can be in 2 replicas (the first four summations in $F_{P_6,1}^{k \geq 3}$) or 3 replicas (the following three summations in $F_{P_6,1}^{k \geq 3}$). The first four summations are derived in a way similar to what we did for $F_{P_6,1}^2$, with a change in the summation indices; $\mathbf{R}_2$ is replaced by $\mathbf{R}_k$ here. The following three summations are associated with Case 6.4. In Case 6.4, c_1 connects overlaps in $\mathbf{R}_r$, $\mathbf{R}_e$, and $\mathbf{R}_s$. On the other hand, c_2 (resp., c_3) connects overlaps in $\mathbf{R}_r$ (resp., $\mathbf{R}_e$) and $\mathbf{R}_s$. For the fifth (double) summation, $\mathcal{D}_{P_6}$ in (19) has $r = 1$, $e = h$, and $s = k$ (see Fig. 6). Thus, i_1 (resp., i_2 and i_3) can take any value in the range from the start of $\mathbf{R}_k$ to the end of $\mathbf{R}_1$ (resp., $\mathbf{R}_1$ and $\mathbf{R}_h$), i.e., from $(k-1)\gamma$ to $(m+1)\gamma - 1$ (resp., $(m+1)\gamma - 1$ and $(m+h)\gamma - 1$). For the sixth (double) summation, $\mathcal{D}_{P_6}$ in (19) has $r = 1$, $e = k$, and $s = h$. Thus, i_1 (resp., i_2 and i_3) can take any value in the range from the start of $\mathbf{R}_k$ (resp., $\mathbf{R}_h$ and $\mathbf{R}_k$) to the end of $\mathbf{R}_1$ (resp., $\mathbf{R}_1$ and $\mathbf{R}_h$), i.e., from $(k-1)\gamma$ (resp., $(h-1)\gamma$ and $(k-1)\gamma$) to $(m+1)\gamma - 1$ (resp., $(m+1)\gamma - 1$ and $(m+h)\gamma - 1$). For the seventh (double) summation, $\mathcal{D}_{P_6}$ in (19) has $r = h$, $e = k$, and $s = 1$. Thus, i_1 (resp., i_2 and i_3) can take any value in the range from the start of $\mathbf{R}_k$ (resp., $\mathbf{R}_h$ and $\mathbf{R}_k$) to the end of $\mathbf{R}_1$, i.e., from $(k-h)\gamma$ (resp., 0 and $(k-h)\gamma$) to $(m-h+2)\gamma - 1$. The outer summation is over all possible values of h , and we have $1 < h < k$. ■

ACKNOWLEDGMENT

The authors would like to thank Homa Esfahanizadeh and Andrew Tan for their assistance in carrying out this research.

REFERENCES

- [1] 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)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–7.
- [2] B. Vasic and E. Kurtas, *Coding and Signal Processing for Magnetic Recording Systems*. Boca Raton, FL, USA: CRC Press, 2005.
- [3] G. Colavolpe and G. Germe, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 818–825, May 2005.

- [4] Y. Han and W. Ryan, "Low-floor detection/decoding of LDPC-coded partial response channels," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 252–260, Feb. 2010.
- [5] 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 Trans. Commun.*, vol. 64, no. 8, pp. 3194–3207, Aug. 2016.
- [6] X. Hu, Z. Li, B. V. K. Vijaya Kumar, and R. Barndt, "Error floor estimation of long LDPC codes on magnetic recording channels," *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 1836–1839, Jun. 2010.
- [7] W. Phakphisut, P. Supnithi, and N. Puttarak, "EXIT chart analysis of nonbinary protograph LDPC codes for partial response channels," *IEEE Trans. Magn.*, vol. 50, no. 11, Nov. 2014, Art. no. 3101904.
- [8] Z. Qin, Y. Ng, and K. Cai, "Iterative detection and decoding for non-binary LDPC coded partial-response channels with written-in errors," *IEE Commun.*, vol. 10, no. 4, pp. 399–406, Mar. 2016.
- [9] Y. Toriyama and D. Markovic, "A 2.267-Gb/s, 93.7-pJ/bit non-binary LDPC decoder with logarithmic quantization and dual-decoding algorithm scheme for storage applications," *IEEE J. Solid-State Circuits*, vol. 53, no. 8, pp. 2378–2388, Aug. 2018.
- [10] P. Chen, C. Kui, L. Kong, Z. Chen, and M. Zhang, "Non-binary protograph-based LDPC codes for 2-D-ISI magnetic recording channels," *IEEE Trans. Magn.*, vol. 53, no. 11, Nov. 2017, Art. no. 8108905.
- [11] A. Hareedy, C. Lanka, and L. Dolecek, "A general non-binary LDPC code optimization framework suitable for dense flash memory and magnetic storage," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 9, pp. 2402–2415, Sep. 2016.
- [12] A. Hareedy, C. Lanka, N. Guo, and L. Dolecek, "A combinatorial methodology for optimizing non-binary graph-based codes: Theoretical analysis and applications in data storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2128–2154, Apr. 2019.
- [13] A. Jimenez Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.
- [14] A. E. Pusane, R. Smarandache, P. O. Vontobel, and D. J. Costello, "Deriving good LDPC convolutional codes from LDPC block codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 835–857, Feb. 2011.
- [15] 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.
- [16] 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, Art. no. 3100211.
- [17] 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.
- [18] I. Andriyanova and A. G. I. Amat, "Threshold saturation for nonbinary SC-LDPC codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 62, no. 5, pp. 2622–2638, May 2016.
- [19] 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.
- [20] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "A novel combinatorial framework to construct spatially-coupled codes: Minimum overlap partitioning," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1693–1697.
- [21] H. Esfahanizadeh, A. Hareedy, R. Wu, R. Galbraith, and L. Dolecek, "Spatially-coupled codes for channels with SNR variation," *IEEE Trans. Magn.*, vol. 54, no. 11, Nov. 2018, Art. no. 9401505.
- [22] 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)*, Aachen, Germany, Jun. 2017, pp. 2940–2944.
- [23] 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.
- [24] A. Hareedy, H. Esfahanizadeh, and L. Dolecek, "High performance non-binary spatially-coupled codes for flash memories," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Kaohsiung, Taiwan, Nov. 2017, pp. 229–233.
- [25] I. E. Bocharova, B. D. Kudryashov, and R. Johannesson, "Searching for binary and nonbinary block and convolutional LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 163–183, Jan. 2016.
- [26] 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.
- [27] S. Landner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput.*, Maui, HI, USA, Jun. 2005, pp. 630–635.
- [28] S. G. Srinivasa, Y. Chen, and S. Dahandeh, "A communication-theoretic framework for 2-DMR channel modeling: Performance evaluation of coding and signal processing methods," *IEEE Trans. Magn.*, vol. 50, no. 3, Mar. 2014, Art. no. 3000307.
- [29] T. V. Souvignier, M. Oberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo decoding for partial response channels," *IEEE Trans. Commun.*, vol. 48, no. 8, pp. 1297–1308, Aug. 2000.
- [30] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.
- [31] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 730–743, Apr. 2001.
- [32] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over $GF(q)$," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.
- [33] M. P. C. 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.
- [34] A. Hareedy, R. Wu, and L. Dolecek, "A channel-aware combinatorial approach to design high performance spatially-coupled codes for magnetic recording systems," 2018, *arXiv:1804.05504*. [Online]. Available: <http://arxiv.org/abs/1804.05504>
- [35] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.
- [36] E. Rosnes, "On the minimum distance of array-based spatially-coupled low-density parity-check codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 884–888.

Ahmed Hareedy (Member, IEEE) is a Postdoctoral Associate with the Electrical and Computer Engineering Department at Duke University. He is interested in research questions in coding/information theory that are fundamental to opportunities created by the current, unparalleled access to data and computing. He received his Bachelor and M.S. degrees in Electronics and Communications Engineering from Cairo University in 2006 and 2011, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from the University of California, Los Angeles (UCLA) in 2018. He worked with Mentor Graphics Corporation between 2006 and 2014. He worked with Intel Corporation in the summers of 2015 and 2017.

Dr. Hareedy won the 2018–2019 Distinguished Ph.D. Dissertation Award in Signals and Systems from the Electrical and Computer Engineering Department at UCLA. He is a recipient of the Best Paper Award from the IEEE Global Communications Conference (GLOBECOM), 2015 (Selected Areas in Communications, Data Storage Track). He won the 2017–2018 Dissertation Year Fellowship (DYF) at UCLA. He won the 2016–2017 Electrical Engineering Henry Samueli Excellence in Teaching Award for teaching Probability and Statistics at UCLA. He is a recipient of the Memorable Paper Award from the Non-Volatile Memories Workshop (NVMW), 2018, in the area of devices, coding, and information theory.

Ruiyi Wu (Student Member, IEEE) is a Ph.D. student in the Electrical and Computer Engineering Department at the University of California, Los Angeles (UCLA). He received his B.S. degree from the Electrical and Computer Engineering Department at UCLA in 2017. His research interests include designing error-correcting codes for modern storage devices.

Lara Dolecek (Senior Member, IEEE) is a Full Professor with the Electrical and Computer Engineering Department and Mathematics Department (courtesy) at the University of California, Los Angeles (UCLA). She holds a B.S. (with honors), M.S., and Ph.D. degrees in Electrical Engineering and Computer Sciences, as well as an M.A. degree in Statistics, all from the University of California, Berkeley. She received the 2007 David J. Sakris Memorial Prize for the most outstanding doctoral research in the Department of Electrical Engineering and Computer Sciences at UC Berkeley. Prior to joining UCLA, she was a Postdoctoral Researcher with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. She received IBM Faculty Award (2014), Northrop Grumman Excellence in Teaching Award (2013), Intel Early Career Faculty Award (2013), University of California Faculty Development Award (2013), Okawa Research Grant (2013), NSF CAREER Award (2012), and Hellman Fellowship Award (2011). With her research group and collaborators, she received numerous best paper awards. 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 currently serves as an Associate Editor for IEEE Transactions on Information Theory and as the Secretary of the IEEE Information Theory Society. Prof. Dolecek has served as a consultant for a number of companies specializing in data communications and storage.