

GRADE-AO: Towards Near-Optimal Spatially-Coupled Codes With High Memories

Siyi Yang*, Ahmed Hareedy†, Shyam Venkatasubramanian*, Robert Calderbank†, and Lara Dolecek*

*Electrical and Computer Engineering Department, University of California, Los Angeles, Los Angeles, CA 90095 USA

†Electrical and Computer Engineering Department, Duke University, Durham, NC 27708 USA

siiyang@ucla.edu, ahmed.hareedy@duke.edu, shyam1999@ucla.edu, robert.calderbank@duke.edu, and dolecek@ee.ucla.edu

Abstract—Spatially-coupled (SC) codes, known for their threshold saturation phenomenon and low-latency windowed decoding algorithms, are ideal for streaming applications and data storage systems. SC codes are constructed by partitioning an underlying block code, followed by rearranging and concatenating the partitioned components in a “convolutional” manner. The number of partitioned components determines the “memory” of SC codes. While adopting higher memories results in improved SC code performance, obtaining optimal SC codes with high memory is known to be hard. In this paper, we investigate the relation between the performance of SC codes and the density distribution of partitioning matrices. We propose a probabilistic framework that obtains (locally) optimal density distributions via gradient descent. Starting from random partitioning matrices abiding by the obtained distribution, we perform low complexity optimization algorithms over the cycle properties to construct high memory, high performance quasi-cyclic SC codes. Simulation results show that codes obtained through our proposed method notably outperform state-of-the-art SC codes with the same constraint length and codes with uniform partitioning.

I. INTRODUCTION

Spatially-coupled (SC) codes, also known as low-density parity-check (LDPC) codes with convolutional structures, are an ideal choice for streaming applications and storage devices thanks to their threshold saturation phenomenon [1]–[5] and amenability to low-latency windowed decoding [6]. SC codes are constructed by partitioning the parity-check matrix of an underlying block code, followed by rearranging the component matrices in a “convolutional” manner. In particular, component matrices are concatenated into a “replica”, and then multiple replicas are placed together, resulting in a “coupled” code. The number of component matrices minus one is referred to as the “memory” of the SC codes [7]–[10].

It is known that the performance of an SC code improves as its memory increases. This is a byproduct of improved node expansion and additional degrees of freedom that can be utilized to decrease the number of small cycles and detrimental objects [8], [9], [11], [12]. In this work and many others [8], [9], [13], [14], optimality is with respect to the number of small cycles in the graph of the SC code. Although the optimization problem of designing SC codes with memory less than 4 has been efficiently solved [8], [9], there remains a vacuum in efficient algorithms that construct good enough SC codes with high memories. Esfahanizadeh et al. [8] proposed a combinatorial framework to develop optimal quasi-cyclic (QC) SC codes, comprising so-called optimal-overlap (OO) to search for the optimal partitioning matrices, and circulant power optimization (CPO) to optimize the lifting parameters, which was extended by Hareedy et al. [9]. However, this method is hard to execute in practice for high memory codes due to the increasing computational complexity. Algorithmic methods that

search for good SC codes with high memories are derived in [13]–[15]. However, high memory codes designed by purely algorithmic methods are unable to offer strict guarantees on performance superiority; several of these codes can even be beat by optimally designed QC-SC codes with lower memories under the same constraint length [14]. Therefore, a method that theoretically identifies an avenue to a near-optimal construction of SC codes with high memories is of significant interest.

In a way similar to random coding in spirit, our objective is to obtain some near-optimal solutions starting from a random partitioning matrix, where the density distribution of component matrices (i.e., edge distribution) is analogous to the degree distribution in random coding. While discrete optimization methods [8], [9] have been shown to suffer from exponential growths in complexity, fueled by the increase in degrees of freedom, we adopt a more efficient, probabilistic framework that searches for the optimal edge distribution via gradient descent, referred to as **gradient-descent distributor (GRADE)**, followed by an **algorithmic optimizer (AO)** that obtains a locally optimal partition near a random partition with edge distribution obtained from GRADE. The current goal is still to minimize the number of small cycles, which reduces undesirable dependencies, and thus improves the code performance. The impact of this probabilistic method extends beyond its performance gains and low complexity. Particularly in the error floor region, a more advanced set of detrimental objects (absorbing sets [11]) governs the LDPC code performance. Our probabilistic method also has high potential to be extended to handle detrimental objects specified by the channel.

In this paper, we propose a probabilistic framework that efficiently searches for near-optimal SC codes with high memories. In Section II, we introduce preliminaries of SC codes and the performance-related metrics. In Section III, we develop the theoretical basis of GRADE, which derives a locally optimal edge distribution from an arbitrarily provided initial distribution and conditions. In Section IV, we introduce two examples of GRADE-AOs that result in near-optimal SC codes: the so-called **gradient descent (GD) codes** and **topologically-coupled (TC) codes**. Our proposed framework is validated in Section V by simulation results of four groups of codes, with the best code in each obtained from GRADE-AO. Finally, we make concluding remarks and introduce possible future work in Section VI.

II. PRELIMINARIES

In this section, we recall the typical construction of SC codes with QC structure. Any QC code with a parity-check matrix \mathbf{H} is obtained by replacing each nonzero (zero) entry of some binary matrix \mathbf{H}^P with a circulant (zero) matrix of size z , $z \in \mathbb{N}$. The matrix \mathbf{H}^P and z are referred to as the protograph and

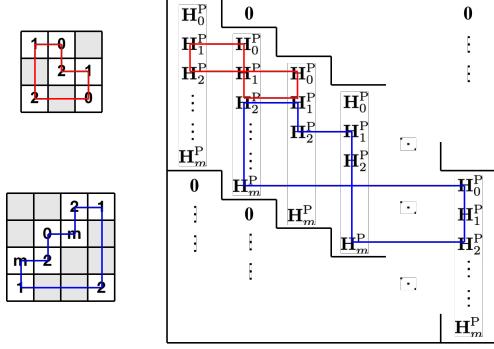


Fig. 1. Cycles in the protograph (right panel) and their corresponding structures in the partitioning matrices (left panel).

the circulant size of the code, respectively. In particular, the protograph $\mathbf{H}_{\text{SC}}^{\text{P}}$ of an SC code has a convolutional structure composed of L replicas, as presented in Fig. 1. Each replica is obtained by stacking the disjoint component matrices $\{\mathbf{H}_i^{\text{P}}\}_{i=0}^m$, where m is the memory and $\mathbf{\Pi} = \mathbf{H}_0^{\text{P}} + \mathbf{H}_1^{\text{P}} + \dots + \mathbf{H}_m^{\text{P}}$ is the protograph of the underlying block code.

In this paper, we constrain $\mathbf{\Pi}$ to be an all-one matrix of size $\gamma \times \kappa$, $\gamma, \kappa \in \mathbb{N}$. An SC code is then uniquely represented by its partitioning matrix \mathbf{P} and lifting matrix \mathbf{L} , where \mathbf{P} and \mathbf{L} are all $\gamma \times \kappa$ matrices. The matrix \mathbf{P} has $(\mathbf{P})_{i,j} = a$ if $(\mathbf{H}_a^{\text{P}})_{i,j} = 1$. The matrix \mathbf{L} is determined by replacing each circulant matrix by its associated exponent. Here, this exponent represents the power to which the matrix σ defined by $(\sigma)_{i,i+1} = 1$ is raised, where $(\sigma)_{z,z+1} = (\sigma)_{z,1}$.

The performance of finite-length LDPC codes is strongly affected by the number of detrimental objects that are subgraphs with certain structures in the Tanner graphs of those codes. Two major classes of detrimental objects are trapping sets and absorbing sets. Since enumerating and minimizing the number of detrimental objects is complicated, existing work typically focuses on common substructures of these objects: the small cycles [8], [9], [13]. A cycle- $2g$ candidate in $\mathbf{H}_{\text{SC}}^{\text{P}}$ ($\mathbf{\Pi}$) is a path of traversing a structure to generate cycles of length $2g$ after lifting (partitioning) [9]. In an SC code, each cycle in the Tanner graph corresponds to a cycle candidate in the protograph $\mathbf{H}_{\text{SC}}^{\text{P}}$, and each cycle candidate in $\mathbf{H}_{\text{SC}}^{\text{P}}$ corresponds to a cycle candidate C in the base matrix $\mathbf{\Pi}$. Lemma 1 specifies a necessary and sufficient condition for a cycle candidate in $\mathbf{\Pi}$ to become a cycle candidate in the protograph and then a cycle in the final Tanner graph.

Lemma 1. *Let C be a cycle- $2g$ candidate in the base matrix, where $g \in \mathbb{N}$, $g \geq 2$. Denote C by $(j_1, i_1, j_2, i_2, \dots, j_g, i_g)$, where (i_k, j_k) , (i_k, j_{k+1}) , $1 \leq k \leq g$, $j_{g+1} = j_1$, are nodes of C in $\mathbf{\Pi}$, \mathbf{P} , and \mathbf{L} . Then C becomes a cycle candidate in the protograph if and only if the following condition follows [13]:*

$$\sum_{k=1}^g \mathbf{P}(i_k, j_k) = \sum_{k=1}^g \mathbf{P}(i_k, j_{k+1}). \quad (1)$$

This cycle candidate becomes a cycle in the Tanner graph if and only if [16]:

$$\sum_{k=1}^g \mathbf{L}(i_k, j_k) \equiv \sum_{k=1}^g \mathbf{L}(i_k, j_{k+1}) \pmod{z}. \quad (2)$$

As shown in Fig. 1, a cycle-6 candidate and a cycle-8 candidate in the partitioning matrix with assignments satisfying

condition (1), and their corresponding cycle candidates in the protograph are marked by red and blue, respectively. An optimization of a QC-SC code is typically divided into two major steps: optimizing \mathbf{P} to minimize the number of cycle candidates in the protograph, and optimizing \mathbf{L} to further reduce that number in the Tanner graph given the optimized \mathbf{P} [8], [9]. The latter goal has been achieved in [8] and [9], using an algorithmic method called circulant power optimization (CPO), while the former goal is yet to be achieved for large m . We note that the step separation highlighted above notably reduces the overall optimization complexity.

In the remainder of this paper, we focus on SC codes for the additive white Gaussian noise (AWGN) channel, where the most detrimental objects are the low weight absorbing sets [8]. Consequently, a simplified optimization focuses on cycle candidates of lengths 4, 6, and 8 [8], [9]. Existing literature shows that the optimal \mathbf{P} for an SC code with $m \leq 2$ typically has a balanced (uniform) edge distribution among component matrices [8]. However, in the remaining sections, we show that the edge distribution for optimal SC codes with large m is not uniform, and we propose the GRADE-AO framework that explores a locally optimal solution.

III. A PROBABILISTIC OPTIMIZATION FRAMEWORK

In this section, we present a probabilistic framework that searches for a locally optimal edge distribution for the partitioning matrices of SC codes with given memories through the gradient descent algorithm. The complete proofs are in [17].

Definition 1. *Let $\gamma, \kappa, m, m_t \in \mathbb{N}$ and $\mathbf{a} = (a_0, a_1, \dots, a_{m_t})$, where $0 = a_0 < a_1 < \dots < a_{m_t} = m$. A (γ, κ) SC code with memory m is said to have **coupling pattern** \mathbf{a} if $\mathbf{H}_i^{\text{P}} \neq \mathbf{0}^{\gamma \times \kappa}$, for all $i \in \{a_0, a_1, \dots, a_{m_t}\}$, and $\mathbf{H}_i^{\text{P}} = \mathbf{0}^{\gamma \times \kappa}$, otherwise. The value m_t is called the **pseudo-memory** of the SC code.*

A. Probabilistic Metric

In this subsection, we define metrics linking the edge distribution with the expected number of cycle candidates in the protograph in Theorem 1 and Theorem 2. While Schmalen et al. have shown in [18] that nonuniform coupling (nonuniform edge distribution in our paper) yields an improved threshold, our work differs in two areas: 1) Explicit optimal coupling graphs were exhaustively searched and were restricted to small memories in [18], whereas our method produces near-optimal SC protographs for arbitrary memories. 2) Work [18] focused on the asymptotic analysis for the threshold region, while our framework is dedicated to the finite-length construction and has additional demonstrable gains in the error floor region.

Definition 2. *Let $m, m_t \in \mathbb{N}$ and $\mathbf{a} = (a_0, a_1, \dots, a_{m_t})$, where $0 = a_0 < a_1 < \dots < a_{m_t} = m$. Let $\mathbf{p} = (p_0, p_1, \dots, p_{m_t})$, where $0 < p_i \leq 1$, $p_0 + p_1 + \dots + p_{m_t} = 1$: each p_i specifies the probability of a ‘1’ in $\mathbf{\Pi}$ going to the component matrix $\mathbf{H}_{a_i}^{\text{P}}$, thus \mathbf{p} is referred to as **edge distribution** under random partition later on. Then, the following $f(X; \mathbf{a}, \mathbf{p})$, which is abbreviated to $f(X)$ when the context is clear, is called the **coupling polynomial** of an SC code with coupling pattern \mathbf{a} , associated with probability distribution \mathbf{p} :*

$$f(X; \mathbf{a}, \mathbf{p}) \triangleq \sum_{0 \leq i \leq m_t} p_i X^{a_i}. \quad (3)$$

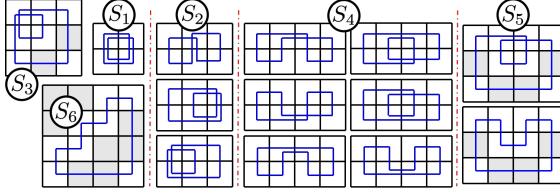


Fig. 2. Structures and cycle candidates for cycle 8.

Theorem 1. Let $[\cdot]_i$ denote the coefficient of X^i of a polynomial. Denote by $P_6(\mathbf{a}, \mathbf{p})$ the probability of a cycle-6 candidate in the base matrix becoming a cycle-6 candidate in the protograph under random partitioning with edge distribution \mathbf{p} . Then,

$$P_6(\mathbf{a}, \mathbf{p}) = [f^3(X)f^3(X^{-1})]_0. \quad (4)$$

Proof sketch. The constant term of $f^3(X)f^3(X^{-1})$ is the sum of all $p_{i_1}p_{i_2}p_{i_3}p_{j_1}p_{j_2}p_{j_3}$ such that $i_1 + i_2 + i_3 = j_1 + j_2 + j_3$, i.e., the sum of the probabilities of all assignments on a cycle-6 candidate that satisfy the cycle condition (1) in Lemma 1. ■

Example 1. Consider SC codes with full memories and uniform partition, i.e., $\mathbf{a} = (0, 1, \dots, m)$ and $\mathbf{p} = \frac{1}{m+1}\mathbf{1}_{m+1}$. When $m = 2$, $P_6(\mathbf{a}, \mathbf{p}) = 0.1934$; when $m = 4$, $P_6(\mathbf{a}, \mathbf{p}) = 0.1121$.

Example 2. First, consider SC codes with $m = m_t = 2$, $\mathbf{a}_1 = (0, 1, 2)$ and $\mathbf{p}_1 = (2/5, 1/5, 2/5)$. According to Theorem 1, $P_6(\mathbf{a}_1, \mathbf{p}_1) = 0.1818$. Second, consider SC codes with $m = m_t = 4$. Let $\mathbf{a}_2 = (0, 1, 2, 3, 4)$ and $\mathbf{p}_2 = (0.31, 0.13, 0.12, 0.13, 0.31)$. According to Theorem 1, $P_6(\mathbf{a}_2, \mathbf{p}_2) = 0.0986$.

After we have derived the metric for cycle-6 candidates in the protograph, we now turn to the case of cycle-8 candidates. As shown in Fig. 2, cycle candidates in the base matrix that result in cycle-8 candidates in the protograph can be categorized into 6 different structures, labeled S_1, \dots, S_6 . Different cases are differentiated by the number of rows and columns (without order) the structures span in the partitioning matrix [9]. Specifically, S_1, \dots, S_5 denote the structures that span submatrices of size 2×2 , 2×3 or 3×2 , 3×3 , 2×4 or 4×2 , 3×4 or 4×3 , and 4×4 , respectively. Any structure that belongs to S_2, S_4, S_5 has multiple cycle-8 candidates, and these distinct candidates are marked by blue in Fig. 2.

Lemma 2. Denote $P_{8;i}(\mathbf{a}, \mathbf{p})$, $1 \leq i \leq 6$, as the average probability of a cycle-8 candidate of structure S_i in the base matrix becoming a cycle-8 candidate in the protograph, under random partition with edge distribution \mathbf{p} . Then,

$$\begin{aligned} P_{8;1}(\mathbf{a}, \mathbf{p}) &= [f^2(X)f^2(X^{-1})]_0, \\ P_{8;2}(\mathbf{a}, \mathbf{p}) &= [f(X^2)f(X^{-2})f^2(X)f^2(X^{-1})]_0, \\ P_{8;3}(\mathbf{a}, \mathbf{p}) &= [f(X^2)f^2(X)f^4(X^{-1})]_0, \text{ and} \\ P_{8;4}(\mathbf{a}, \mathbf{p}) &= P_{8;5}(\mathbf{a}, \mathbf{p}) = P_{8;6}(\mathbf{a}, \mathbf{p}) = [f^4(X)f^4(X^{-1})]_0. \end{aligned}$$

Theorem 2. Denote $N_8(\mathbf{a}, \mathbf{p})$ as the expectation of the number of cycle-8 candidates in the protograph. Then,

$$\begin{aligned} N_8(\mathbf{a}, \mathbf{p}) &= w_1 [f^2(X)f^2(X^{-1})]_0 \\ &+ w_2 [f(X^2)f(X^{-2})f^2(X)f^2(X^{-1})]_0 \\ &+ w_3 [f(X^2)f^2(X)f^4(X^{-1})]_0 + w_4 [f^4(X)f^4(X^{-1})]_0, \end{aligned}$$

where $w_1 = \binom{\gamma}{2}\binom{\kappa}{2}$, $w_2 = 3\binom{\gamma}{2}\binom{\kappa}{3} + 3\binom{\gamma}{3}\binom{\kappa}{2}$, $w_3 = 18\binom{\gamma}{3}\binom{\kappa}{3}$, $w_4 = 6\binom{\gamma}{2}\binom{\kappa}{4} + 6\binom{\gamma}{4}\binom{\kappa}{2} + 36\binom{\gamma}{3}\binom{\kappa}{4} + 36\binom{\gamma}{4}\binom{\kappa}{3} + 24\binom{\gamma}{4}\binom{\kappa}{4}$.

We ignore the number of replicas a cycle candidate spans in \mathbf{H}_{SC}^p . We take care of that number for \mathbf{H}_{SC} at the CPO stage.

B. Gradient Descent Distributor

By contrasting Examples 1 with 2, it is clear that for a given coupling pattern, an optimal edge distribution is not necessarily reached by a uniform partition. In this subsection, we develop an algorithm that obtains a locally optimal distribution by gradient descent.

Lemma 3. Given $m_t \in \mathbb{N}$ and $\mathbf{a} = (a_0, a_1, \dots, a_{m_t})$, a necessary condition for $P_6(\mathbf{a}, \mathbf{p})$ to reach its minimum value is that the following equation holds for some $c_0 \in \mathbb{R}$:

$$[f^3(X)f^2(X^{-1})]_{a_i} = c_0, \quad \forall i, 0 \leq i \leq m_t. \quad (5)$$

Proof sketch. This can be proved by the KKT conditions of the Lagrangian $L_6(\mathbf{a}, \mathbf{p}) = P_6(\mathbf{a}, \mathbf{p}) + c(1 - p_0 - p_1 - \dots - p_{m_t})$. ■

Lemma 4. Given $\gamma, \kappa, m_t \in \mathbb{N}$ and $\mathbf{a} = (a_0, a_1, \dots, a_{m_t})$, a necessary condition for $N_8(\mathbf{a}, \mathbf{p})$ to reach its minimum value is that the following equation holds for some $c_0 \in \mathbb{R}$:

$$\begin{aligned} &[4f^2(X)f(X^{-1})]_{a_i} + \bar{w}_2 [2f(X^2)f^2(X)f^2(X^{-1})]_{2a_i} \\ &+ \bar{w}_2 [4f(X^2)f(X^{-2})f^2(X)f(X^{-1})]_{a_i} \\ &+ \bar{w}_3 [f^2(X)f^4(X^{-1})]_{-2a_i} + \bar{w}_3 [2f(X^2)f(X)f^4(X^{-1})]_{-a_i} \\ &+ \bar{w}_3 [4f(X^2)f^2(X)f^3(X^{-1})]_{a_i} \\ &+ \bar{w}_4 [8f^4(X)f^3(X^{-1})]_{a_i} = c_0, \quad \forall i, 0 \leq i \leq m_t, \end{aligned}$$

where $\bar{w}_2 = \gamma + \kappa - 4$, $\bar{w}_3 = 2(\gamma - 2)(\kappa - 2)$, and $\bar{w}_4 = \frac{1}{2}[(\gamma - 2)(\gamma - 3) + (\kappa - 2)(\kappa - 3)] + (\gamma - 2)(\kappa - 2)(\gamma + \kappa - 6) + \frac{1}{6}(\gamma - 2)(\gamma - 3)(\kappa - 2)(\kappa - 3)$.

Based on Lemma 3 and Lemma 4, we adopt the gradient descent algorithm to obtain a locally optimal edge distribution for SC codes with coupling pattern \mathbf{a} , starting from the uniform distribution inside \mathbf{P} as presented in Algorithm 1. Note that $\text{conv}(\cdot)$ and $\text{inverse}(\cdot)$ refer to convolution and reverse of vectors, respectively.

IV. CONSTRUCTION

In this section, we present two algorithmic optimization methods based on GRADE to obtain locally optimal SC codes with a fixed coupling pattern.

A. Gradient Descent Codes

In this subsection, we consider SC codes with full memories, i.e., $m = m_t$ and $\mathbf{a} = (0, 1, \dots, m)$. In this case, our proposed GRADE algorithm obtains a highly skewed edge distribution. Starting from a random partitioning matrix \mathbf{P} with the derived distribution, one can perform a semi-greedy algorithm that searches for the locally optimal partitioning matrix near the initial \mathbf{P} . Constraining the search space to contain \mathbf{P} 's that have distributions within small L_1 and L_∞ distances from that of the original \mathbf{P} , and adopting the CPO next, significantly reduces the computational complexity to find a strong high memory code. This procedure is given in Algorithm 2.

We refer to codes obtained from Algorithm 2 as **gradient descent (GD) codes**. By replacing the input distribution \mathbf{p} in Algorithm 2 with the uniform distribution, we obtain the so-called **uniform (UNF) codes**. We show in Section V by simulation that the distribution obtained by GRADE results

Algorithm 1 Gradient Descent Distributor (GRADE)

Inputs and Parameters:

$\gamma, \kappa, m_t, m, \mathbf{a}$: parameters of the SC code;
 w : weight of each cycle-6 candidate;
 ϵ, α : accuracy and step size of gradient descent;

Outputs and Intermediate Variables:

\mathbf{p} : a locally optimal edge distribution over $\text{supp}(\mathbf{a})$;

- 1: $\bar{w}_1 \leftarrow \frac{2w}{3}(\gamma - 2)(\kappa - 2)$, obtain $\{\bar{w}_i\}_{i=2}^4$ in Lemma 4;
- 2: $v_{\text{prev}} = 1; v_{\text{cur}} = 1$;
- 3: $\mathbf{p}, \mathbf{g} \leftarrow \mathbf{0}_{m_t+1}, \mathbf{f}, \bar{\mathbf{f}} \leftarrow \mathbf{0}_{m+1}, \mathbf{f}_2, \bar{\mathbf{f}}_2 \leftarrow \mathbf{0}_{2m+1}$;
- 4: $\mathbf{p} \leftarrow \frac{1}{m_t+1} \mathbf{1}_{m_t+1}$;
- 5: $\mathbf{f} [a_0, \dots, a_{m_t}] \leftarrow \mathbf{p}, \bar{\mathbf{f}} \leftarrow \text{inverse}(\mathbf{f})$;
- 6: $\mathbf{f}_2 [1, 3, \dots, 2m+1] \leftarrow \mathbf{f}, \bar{\mathbf{f}}_2 \leftarrow \text{inverse}(\mathbf{f}_2)$;
- 7: $\mathbf{q}_1 \leftarrow \bar{w}_1 \text{conv}(\mathbf{f}, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \mathbf{f}, \bar{\mathbf{f}}), \mathbf{q}_2 \leftarrow \text{conv}(\mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}})$;
- 8: $\mathbf{q}_3 \leftarrow \bar{w}_2 \text{conv}(\mathbf{f}_2, \bar{\mathbf{f}}_2, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}}) + \bar{w}_3 \text{conv}(\mathbf{f}_2, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}}, \bar{\mathbf{f}}) + \bar{w}_4 \text{conv}(\mathbf{f}, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}}, \bar{\mathbf{f}})$;
- 9: $v_{\text{prev}} = v_{\text{cur}}, v_{\text{cur}} = \mathbf{q}_1 [3m] + \mathbf{q}_2 [2m] + \mathbf{q}_3 [4m]$;
- 10: $\mathbf{g}_1 \leftarrow 6\bar{w}_1 \text{conv}(\mathbf{f}, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \mathbf{f}), \mathbf{g}_2 \leftarrow 4 \text{conv}(\mathbf{f}, \mathbf{f}, \bar{\mathbf{f}})$;
- 11: $\mathbf{g}_3 \leftarrow 4\bar{w}_2 \text{conv}(\mathbf{f}_2, \bar{\mathbf{f}}_2, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}) + 2\bar{w}_3 \text{conv}(\bar{\mathbf{f}}_2, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}}) + 4\bar{w}_4 \text{conv}(\mathbf{f}_2, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}})$;
- 12: $\mathbf{g}_4 \leftarrow 2\bar{w}_2 \text{conv}(\mathbf{f}_2, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}}) + \bar{w}_3 \text{conv}(\mathbf{f}, \mathbf{f}, \mathbf{f}, \bar{\mathbf{f}}, \bar{\mathbf{f}})$;
- 13: $\mathbf{g} \leftarrow \mathbf{g}_1 [2m + \mathbf{a}] + \mathbf{g}_2 [m + \mathbf{a}] + \mathbf{g}_3 [3m + \mathbf{a}] + \mathbf{g}_4 [2m + 2\mathbf{a}], \mathbf{g} \leftarrow \mathbf{g} - \text{mean}(\mathbf{g})$;
- 14: **if** $|v_{\text{prev}} - v_{\text{cur}}| > \epsilon$ **then** $\mathbf{p} \leftarrow \mathbf{p} - \alpha \frac{\mathbf{g}}{\|\mathbf{g}\|}$, **goto** step 5;
- 15: **return** \mathbf{p} ;

in constructions that are better than those adopting uniform distribution and in existing literature.

B. Topologically-Coupled Codes

In this subsection, we explore SC codes with pseudo-memory m_t such that $m_t < m$ and $\mathbf{a} \neq (0, 1, \dots, m)$. The motivation behind this task is to construct an SC code with memory m with the same computational complexity needed to construct a memory m_t code, where $m_t < m$. Given m_t and m , we first find the optimal \mathbf{a} with length $m_t + 1$ in a brute-force manner. Taking $m = 4$ and $m_t = 2$ as an example, the optimal coupling pattern is reached by $\mathbf{a} = (0, 1, 4)$ and the corresponding optimal distribution is almost uniform.

Given the optimal coupling pattern \mathbf{a} , we then obtain an optimal partitioning matrix by the OO method proposed in [8] and [9]. We extend the OO method for memory m_0 SC codes to any SC code with pseudo-memory $m_t = m_0$, which does not increase the complexity of the approach. We refer to the codes obtained from the extended OO method followed by the CPO as **topologically-coupled (TC) codes**.

Optimal TC codes with pseudo-memory m_t have strictly fewer cycle candidates in their photographs than optimal SC codes with full memory $m = m_t$. Take $m = 4$ and $m_t = 2$ as an example. Suppose the optimal SC code has the partition $\mathbf{\Pi} = \mathbf{H}_0^P + \mathbf{H}_1^P + \mathbf{H}_2^P$. Consider the TC code with partition $\mathbf{\Pi} = \mathbf{H}_0^P + \mathbf{H}_1^P + \mathbf{H}_4^P$ such that $\mathbf{H}_2^P = \mathbf{H}_4^P$. Then, any cycle-6 candidate resulting from a cycle candidate in the base matrix assigned with 0-1-0-1-2-0, 1-2-1-2-2-0, or 0-1-2-1- x - x , $x \in \{0, 1, 2\}$, in \mathbf{P} no longer has a counterpart in the TC code, since by replacing 2's with 4's, assignments 0-1-0-1-4-0, 1-4-1-4-4-0, and 0-1-4-1- x - x , $x \in \{0, 1, 4\}$, no longer satisfy the cycle condition in Lemma 1. Moreover, there exists

Algorithm 2 GRADE-A Optimizer (AO)

Inputs and Parameters:

γ, κ, m : parameters of an SC code with full memory;
 \mathbf{p} : edge distribution obtained from Algorithm 1;
 d_1, d_2 : parameters indicating the size of the searching space;

Outputs and Intermediate Variables:

\mathbf{P} : a locally optimal partitioning matrix;

- 1: Obtain the lists $\mathcal{L}_6(i, j), \mathcal{L}_8(i, j)$ of cycle-6 candidates and cycle-8 candidates in the base matrix that contain node (i, j) , $1 \leq i \leq \gamma, 1 \leq j \leq \kappa$;
- 2: Obtain $\mathbf{u} = \arg \min_{\mathbf{x} \in \mathbb{N}^{m+1}, \|\mathbf{x}\|_1 = \gamma\kappa} \|\frac{1}{\gamma\kappa} \mathbf{x} - \mathbf{p}\|_2$;
- 3: **for** $i \in \{0, 1, \dots, m\}$ **do**
- 4: | Place $\mathbf{u}[i+1]$ i 's into \mathbf{P} randomly;
- 5: $\mathbf{d} \leftarrow \mathbf{0}_{m+1}$;
- 6: **for** $i \in \{1, 2, \dots, \gamma\}, j \in \{1, 2, \dots, \kappa\}$ **do**
- 7: | $n_{\text{optimal}} \leftarrow \text{False}$;
- 8: | $n_6 \leftarrow |\mathcal{L}_6(i, j)|, n_8 \leftarrow |\mathcal{L}_8(i, j)|, n \leftarrow w n_6 + n_8$;
- 9: **for** $v \in \{0, 1, \dots, z-1\}$ **do**
- 10: | $\mathbf{d}' = \mathbf{d}, \mathbf{d}'[v+1] \leftarrow \mathbf{d}'[v+1] + 1, p \leftarrow \mathbf{P}(i, j)$;
- 11: | **if** $\|\mathbf{d}'\|_1 \leq d_1$ and $\|\mathbf{d}'\|_\infty \leq d_2$ **then**
- 12: | | $\mathbf{P}(i, j) \leftarrow v$;
- 13: | | $t_6 \leftarrow |\mathcal{L}_6(i, j)|, t_8 \leftarrow |\mathcal{L}_8(i, j)|, t \leftarrow w t_6 + t_8$;
- 14: | | **if** $t < n$ **then**
- 15: | | | $n_{\text{optimal}} \leftarrow \text{True}, n \leftarrow t, \mathbf{d} \leftarrow \mathbf{d}'$;
- 16: | | **else** $\mathbf{P}(i, j) \leftarrow p$;
- 17: **if** n_{optimal} **then** **goto** step 6;
- 18: **return** \mathbf{P} ;

a bijection between the remaining candidates in the SC code and all candidates in the TC code through the replacement of 2's with 4's. Therefore, TC codes are better (have less cycles) than SC codes with the same circulant size. In Section V, we present simulation results of such codes and show that they can also outperform SC codes with the same constraint length (larger circulant size).

V. SIMULATION RESULTS

In this section, we obtain the frame error rate (FER) curves of four groups of SC codes designed by the GRADE-AO methods presented in Section IV. We show that codes constructed by the GRADE-AO methods offer significant performance gains compared with codes with uniform edge distributions and codes constructed through purely algorithmic methods.

Out of these three plots, the left and center ones compare GD codes with UNF codes designed as in Section IV-A. The right plot compares a TC code designed as in Section IV-B with optimal SC codes constructed through the OO-CPO method proposed in [8]. The GD/UNF codes have parameters $(\gamma, \kappa, m, z, L) = (3, 7, 5, 13, 100), (3, 17, 9, 7, 100)$, and $(4, 29, 19, 29, 20)$, respectively. The TC code has parameters $(\gamma, \kappa, m_t, z, L) = (4, 17, 2, 17, 50)$ with coupling pattern $\mathbf{a} = (0, 1, 4)$. For a fair comparison, we have selected two SC codes: one with a similar constraint length $(m+1)z$ and the other with an identical circulant power z . To ensure that the SC codes and the TC code have close rates and codelengths, the two SC codes

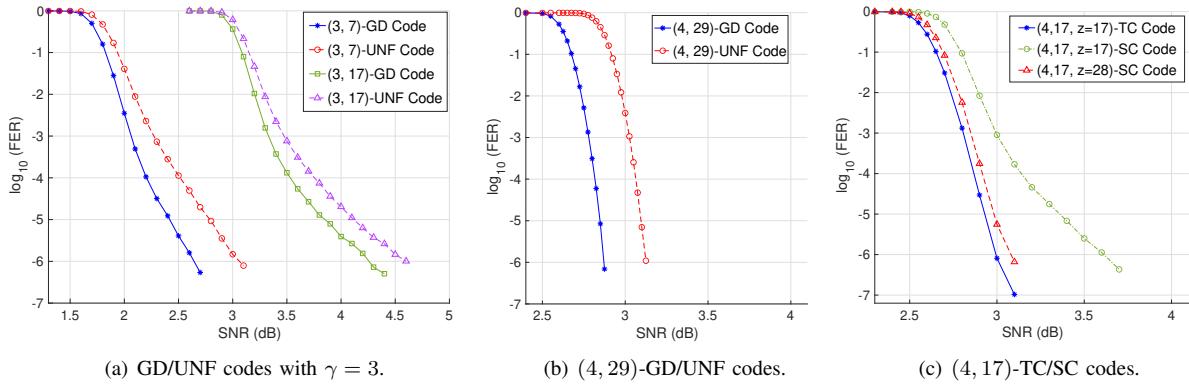


Fig. 3. Simulation results of codes constructed using the methods presented in Section IV.

TABLE I
STATISTICS OF THE NUMBER OF CYCLES

(γ, κ)	Code	Cycles-6	Cycles-8
$(3, 7)$	GD	0	0
	UNF	0	6292
$(3, 17)$	GD	0	397880
	UNF	0	559902
	Battaglioni et al. [13]	0	451337
$(4, 29)$	GD	0	528,090
	UNF	0	1,087,268
$(4, 17)$	TC	15,436	-
	SC (matched constraint length)	19,180	-
	SC (matched circulant size)	74,579	-

have parameters $(\gamma, \kappa, m, z, L) = (4, 17, 2, 28, 30)$ and $(4, 17, 2, 17, 50)$, respectively. The partitioning and the lifting matrices of these codes are specified in [17]. The statistics regarding the number of cycles of each code are presented in Table I. The decoder adopts a finite-precision fast Fourier transform based sum-product algorithm and performs a maximum of 50 iterations: it stops if a codeword is reached sooner.

Fig. 3(a) shows FER curves of our GD/UNF comparisons with $(\gamma, \kappa) = (3, 7)$ and $(3, 17)$. When $\gamma = 3$, cycles-6 are easily removed by the CPO. Therefore, we perform joint optimization on the number of cycle-6 and cycle-8 candidates by assigning different weights to cycle candidates in Algorithm 2. We observe a performance gain for the GD code with respect to the UNF code in both the waterfall region and the error floor region. Moreover, the number of cycles-8 in the $(3, 17)$ GD code is reduced by 29% and 12% compared with the UNF code and the code constructed by Battaglioni et al. in [13], respectively. In addition, the $(3, 17)$ GD code has no weight-6 absorbing sets (ASs) and 133 weight-7 ASs, whereas the UNF code has 6 weight-6 ASs and 361 weight-7 ASs. As for the $(3, 7)$ codes, all cycles-6 and cycles-8 are removed. Thus, the gain of the GD code compared with the UNF code exceeds the gain observed in the $(3, 17)$ codes.

Fig. 3(b) shows FER curves of the GD/UNF comparison with $(\gamma, \kappa) = (4, 29)$. Cycles-6 in the GD code and the UNF code are both removed, and the number of cycles-8 in the GD code demonstrates a 51.4% reduction from the count observed in the UNF code. It is worth mentioning that both codes have no ASs of weights up to 8, which is reflected in their FER curves via the sharp waterfall regions and the non-existing error floor regions. The FER of the GD/UNF codes decreases with a rate exceeding 12 orders of magnitude per 0.5 dB. Moreover, the GD code has

a significant gain of about 0.25 dB over the UNF code. These results substantiate the significant potential of the GRADE-AO method in constructing SC codes with superior performance for storage devices, with further applications including wireless communication systems and quantum communication systems.

Fig. 3(c) shows the FER curves of the TC/SC codes with $(\gamma, \kappa) = (4, 17)$. The number of cycles-6 in the $(4, 17)$ TC code demonstrates a 79% and a 20% reduction from the counts observed in the SC codes with a matched constraint length and a matched circulant size, respectively. Moreover, the TC code has no weight-6 nor weight-8 ASs. It is shown that the TC code outperforms the optimal SC code with a matched constraint length, and that the gain is of greater magnitude when compared with the SC code of identical circulant size. Note that although TC codes have higher memories and thus larger constraint lengths than SC codes of matched circulant sizes, they possess the same number of nonzero component matrices, and thus the same degrees of freedom in construction. This fact makes TC codes even more promising if we can devise for them windowed decoding algorithms with window sizes comparable to the corresponding SC codes of matched circulant sizes.

VI. CONCLUSION

Discrete optimization of the constructions of spatially-coupled (SC) codes with high memories is known to be computationally expensive. Algorithmic optimization is efficient, but can hardly guarantee the performance because of the lack of theoretical guidance. In this paper, we proposed a so-called GRADE-AO method, a probabilistic framework that efficiently searches for locally optimal QC-SC codes with arbitrary memories. We obtain a locally optimal edge distribution that minimizes the expected number of cycle candidates by gradient descent. Starting from a random partitioning matrix with the derived edge distribution, we use algorithmic optimization to find a locally optimal partitioning matrix near it. Simulation results show that our proposed constructions have a significant performance gain over state-of-the-art codes. Future work includes extending the framework on cycle optimization into a one that focuses on detrimental objects.

ACKNOWLEDGMENT

This work was supported in part by UCLA Dissertation Year Fellowship, NSF under the Grants CCF-BSF 1718389, CCF 1717602, CCF 2008728, and CCF 1908730, and in part by AFOSR under the Grant 8750-20-2-0504.

REFERENCES

[1] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Information Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.

[2] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Trans. Information Theory*, vol. 60, no. 12, pp. 7389–7415, Dec. 2014.

[3] P. M. Olmos and R. L. Urbanke, "A scaling law to predict the finite-length performance of spatially-coupled LDPC codes," *IEEE Trans. Information Theory*, vol. 61, no. 6, pp. 3164–3184, 2015.

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

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

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

[7] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from photographs," *IEEE Trans. Information Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

[8] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Finite-length construction of high performance spatially-coupled codes via optimized partitioning and lifting," *IEEE Trans. Communications*, vol. 67, no. 1, pp. 3–16, Jan. 2018.

[9] A. Hareedy, R. Wu, and L. Dolecek, "A channel-aware combinatorial approach to design high performance spatially-coupled codes," *IEEE Trans. Information Theory*, vol. 66, no. 8, pp. 4834–4852, Aug. 2020.

[10] A. E. Pusane, R. Smarandache, P. O. Vontobel, and D. J. Costello, "Deriving good LDPC convolutional codes from LDPC block codes," *IEEE Trans. Information Theory*, vol. 57, no. 2, pp. 835–857, Feb. 2011.

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

[12] S. Naseri and A. H. Banihashemi, "Spatially coupled LDPC codes with small constraint length and low error floor," *IEEE Communications Letters*, vol. 24, no. 2, pp. 254–258, Feb. 2020.

[13] M. Battaglioni, A. Tasdighi, G. Cancellieri, F. Chiaraluce, and M. Baldi, "Design and analysis of time-invariant SC-LDPC convolutional codes with small constraint length," *IEEE Trans. Communications*, vol. 66, no. 3, pp. 918–931, Mar. 2018.

[14] S. Mo, L. Chen, D. J. Costello, D. G. M. Mitchell, R. Smarandache, and J. Qiu, "Designing photograph-based quasi-cyclic spatially coupled LDPC codes with large girth," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5326–5337, 2020.

[15] A. Beemer, S. Habib, C. A. Kelley, and J. Kliewer, "A generalized algebraic approach to optimizing SC-LDPC codes," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017, pp. 672–679.

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

[17] S. Yang, A. Hareedy, S. Venkatasubramanian, R. Calderbank, and L. Dolecek, "GRADE-AO: Towards near optimal spatially-coupled codes with high memories," 2020. [Online]. Available: <https://arxiv.org/abs/2101.09962>

[18] L. Schmalen, V. Aref, and F. Jardel, "Non-uniformly coupled LDPC codes: Better thresholds, smaller rate-loss, and less complexity," in *2017 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2017, pp. 376–380.