Rate-Loss Mitigation of SC-LDPC Codes Without Performance Degradation

Hee-Youl Kwak, Dae-Young Yun, and Jong-Seon No, *Fellow, IEEE*

***Abstract*— In research on spatially-coupled low-density parity-check (SC-LDPC) codes, rate-loss of SC-LDPC codes is one of the main issues to be addressed. One way to mitigate the rate-loss is to attach additional variable nodes with an irregular degree distribution, where the degree distribution is optimized with a constraint that the belief propagation (BP) threshold should not be degraded by attaching variable nodes. However, it is observed that the degree distribution obtained with the BP threshold constraint induces degradation of the finite-length performance. In order to address the problem, we propose new optimization methods to attach additional variable nodes while minimizing performance degradation. The proposed optimization methods are based on several design techniques including the scaling law, local threshold, expected graph evolution, differential evolution algorithms, the use of a protograph structure, and puncturing codewords. Using the optimized structure for additional variable nodes, the rate-loss of SC-LDPC codes can be reduced by more than 53% without sacrificing the finite-length performance. It is also shown that the rate-loss mitigation can be translated into a performance improvement if the proposed and the conventional SC-LDPC codes are compared at the same code rate.**

***Index Terms*— Code rate, finite-length performance, low-density parity-check (LDPC) codes, rate-loss, spatially-coupled low-density parity-check (SC-LDPC) codes.**

I. Introduction

SPATIALLY-COUPLED low-density parity-check (SC-LDPC) codes achieve the channel capacity over general binary memoryless symmetric channels under iterative belief propagation (BP) decoding [1]. Compared to uncoupled block LDPC codes, the outstanding performance of SC-LDPC codes comes from their ability to realize wave-like decoding [2], which is triggered by boundary low-degree check nodes [3]. However, SC-LDPC codes suffer from a rate-loss problem in which the code rate of SC-LDPC codes is reduced from that of corresponding uncoupled block LDPC codes. Thus, many works have investigated the rate-loss problem of SC-LDPC codes. First, we can reduce the rate-loss to the desired level simply by adjusting code parameters such as the chain length L of SC-LDPC codes. Thus, in general, a large value of L is selected when designing capacity approaching SC-LDPC codes [3], [4] and decoding is performed using the windowed decoder [5], [6] to address the long blocklength. Increasing the chain length, however, causes the block error rate (BLER) to scale linearly as L increases [7]. Second, it is possible to construct an SC-LDPC code with a target rate by coupling block LDPC codes with a slightly higher rate. However, the SC-LDPC code cannot achieve the desired outstanding decoding performance due to the inferior performance of constituting block codes.

Another way to mitigate the rate-loss is modifying the code structure of SC-LDPC codes [8]–[12]. In [8], a modified SC-LDPC code structure is proposed to reduce the rate-loss but this code structure is not suitable for a sliding window decoder [5], which is a natural way to decode SC-LDPC codes. In [9]–[12], rate-loss is reduced by attaching additional variable nodes with regular [9], [10] or irregular degree distributions [11], [12]. By attaching extra variable nodes to the boundary check nodes, the code dimension is increased and accordingly the rate-loss can be kept to small even with a moderate value of L. In one study [11], they show that the rate-loss can be substantially mitigated by optimizing the degree distribution of additional variable nodes.

However, attaching additional variable nodes leads to a degradation in decoding performance. In [11], they design the degree distribution of additional variable nodes under the BP threshold constraint that the BP threshold should not change after attaching additional variable nodes. Then, the asymptotic decoding performance remains unchanged and it can be expected that the decoding performance will not be degraded. However, we observe that the SC-LDPC codes optimized in [11] show the performance degradation in terms of block error rate (BLER) of finite-length codes despite the fact that their asymptotic performances are equivalent in terms of the BP threshold.

With this problem in mind, we propose new optimization methods to attach additional variable nodes without sacrificing the finite-length performance. First, we separately define local decoding measures for additional variable nodes such as a local BLER and local threshold. Then, we derive the required local BLER to maintain the finite-length performance

Manuscript received February 25, 2019; revised July 11, 2019 and September 22, 2019; accepted October 27, 2019. Date of publication November 7, 2019; date of current version January 15, 2020. This work was supported by the Samsung Electronics Co., Ltd., in Korea. The associate editor coordinating the review of this article and approving it for publication was E. Paolini. *(Corresponding author: Hee-Youl Kwak.)*

H.-Y. Kwak is with Samsung Electronics, Co., Ltd., Hwasung 18448, South Korea (e-mail: ghy1228@gmail.com).

D.-Y. Yun and J.-S. No are with the Department of Electrical and Computer Engineering, INMC, Seoul National University, Seoul 08826, South Korea (e-mail: dyyun@ccl.snu.ac.kr; jsno@snu.ac.kr).

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

Digital Object Identifier 10.1109/TCOMM.2019.2952074

from the scaling law [7] of SC-LDPC codes. Finally, we obtain the degree distribution of additional variable nodes achieving the required local BLER by adjusting the local threshold of additional variable nodes with differential evolution algorithms [13].

In addition, a design constraint is proposed using the expected graph evolution [7], [14], [15], which is directly related to the finite-length performance. According to the analysis of SC-LDPC codes based on the expected graph evolution, decoding failures rarely occur in the initial phase of the decoding process [7]. However, if too many additional variable nodes are added at the boundary, a local minimum of the number of degree-1 check nodes occurs in the initial phase of the decoding process, which implies the finite-length performance degradation. Thus, we newly define the local minimum constraint that the local minimum of the number of degree-1 check nodes should not exist in the initial phase. With the local minimum constraint, we reduce the rate-loss further while attaining the required local BLER. Finally, the rate-loss is mitigated even further by employing a protograph structure for additional variable nodes with the precoding [16] and puncturing [17] techniques.

In summary, a variety of well-known design techniques is used to solve the problem of the finite-length performance degradation. Since the design techniques were derived for the block LDPC or conventional SC-LDPC codes, it is not a trivial problem to directly apply the design techniques for SC-LDPC codes with additional variable nodes is not a trivial problem. In other words, the contribution of this paper is to modify and combine well-known techniques to design good finite-length SC-LDPC codes with reduced rate-loss. From the experimental results, the proposed SC-LDPC codes with the optimized structure of additional variable nodes show the finite-length performance nearly identical to those of conventional SC-LDPC codes while the rate-loss is significantly mitigated. This makes it possible to construct capacity approaching SC-LDPC codes even with a moderate L value. In addition, we show that the gain in the code rate can be translated into a performance improvement when a comparison is made under the same code rate.

The remainder of the paper is organized as follows. Section II introduces the SC-LDPC code structure and its density evolution equations. In Section III, new optimization methods for the degree distribution of additional variable nodes are proposed and the finite-length performance of the proposed SC-LDPC codes is evaluated. In Section IV, the performances of the proposed SC-LDPC codes and conventional SC-LDPC codes are compared at the same code rate. Finally, conclusions are given in Section V.

II. Code Structure and Density Evolution Equations

A. (l, r, L) SC-LDPC Ensemble

The conventional SC-LDPC ensemble in [3] with variable and check node degrees $(3, 6)$ and chain length L is represented by the following base matrix

$$\overbrace{\begin{bmatrix} 1 & 1 & & & & & \\ 1 & 1 & \ddots & & & & \\ 1 & 1 & \ddots & 1 & 1 & & \\ & & \ddots & 1 & 1 & 1 & 1 \\ & & & 1 & 1 & 1 & 1 \\ & & & & & 1 & 1 \end{bmatrix}}^{2L}$$

whose size is $(L + 2) \times 2L$. We mainly consider the degree pair $(l, r) = (3, 6)$ as a running example in this paper. Each column and row of the base matrix correspond to a variable node and a check node in the protograph, respectively, and $2L$ variable nodes are grouped into L positions, where each position consists of two variable nodes. A parity check matrix of a code instance is obtained by lifting the base matrix [18] with lifting factor z. Let M denote the number of variable nodes at each position, i.e., $M = 2z$.

As illustrated in [3], the code structure of SC-LDPC codes that the degrees of the first and last two check nodes are lower than 6 results in the following properties. First, it induces rate-loss from the design rate of uncoupled LDPC codes. Second, boundary check nodes with low degrees trigger the wave-like propagation of reliable information, where two decoding waves propagate from both ends toward the inside on the Tanner graph. However, two decoding waves are unnecessary for the windowed decoder which utilizes only one decoding wave. Thus, a modified SC-LDPC ensemble, denoted by (l, r, L) ensemble, is introduced in [11] by folding the last check node. For example, the base matrix of the $(3, 6, L)$ ensemble is represented as

$$\overbrace{\begin{bmatrix} 1 & 1 & & & & & \\ 1 & 1 & \ddots & & & & \\ 1 & 1 & \ddots & 1 & 1 & & \\ & & \ddots & 1 & 1 & 1 & 1 \\ & & & 1 & 1 & 2 & 2 \end{bmatrix}}^{2L} \tag{1}$$

whose size is $(L + 1) \times 2L$. By folding the last check node, the decoding wave cannot be triggered from the right boundary and only one decoding wave is propagated from the left boundary in the graph. However, the rate-loss is halved compared to that of the conventional SC-LDPC ensemble for a given L. Considering the number of variable and check nodes in (1), the design rate of the $(3, 6, L)$ SC-LDPC ensemble is given as

$$R_{(3,6,L)} = 1 - \frac{(L+1)z}{2Lz} = \frac{1}{2} - \frac{1}{2L} \tag{2}$$

where the first term is the design rate of the uncoupled $(3, 6)$ regular LDPC ensemble and the second term $1/(2L)$ corresponds to the rate-loss, denoted by $\Delta R_{(3,6,L)}$. Note that the design rate and asymptotic properties under BP decoding of the $(3, 6, L)$ SC-LDPC ensemble represented by (1) are equivalent to those of the conventional SC-LDPC ensemble

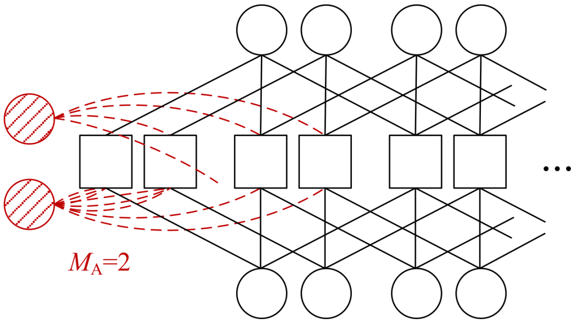


Fig. 1. Tanner graph of the $(3, 6, L, \lambda(x))$ ensemble with $z = 2$ and $\lambda(x) = 1/3x^3 + 2/3x^7$. The hatched circles corresponds to additional variable nodes.

with $2L$ due to the symmetric structure of the conventional SC-LDPC ensemble.

B. $(l, r, L, \lambda(x))$ SC-LDPC Ensemble

To mitigate the rate-loss of the (l, r, L) SC-LDPC ensemble, the $(l, r, L, \lambda(x))$ SC-LDPC ensemble was proposed by attaching additional variable nodes with degree distribution $\lambda(x)$ to the (l, r, L) ensemble [11]. We refer to the part of the (l, r, L) ensemble in the $(l, r, L, \lambda(x))$ ensemble as the main region, that is, the $(l, r, L, \lambda(x))$ ensemble consists of the main region and attached additional variable nodes. For the $(3, 6, L, \lambda(x))$ ensemble, additional variable nodes are attached to the first $2z$ check nodes in a lifted Tanner graph with lifting factor z such that the degree of the $2z$ check nodes becomes 6. For example, Fig. 1 shows the Tanner graph of $(3, 6, L, \lambda(x))$ ensemble with $z = 2$ and $\lambda(x) = 1/3x^3 + 2/3x^7$. Let $M_{\rm A}$ denote the number of additional variable nodes in the lifted Tanner graph and $l_{\rm avg}$ denote the average degree of additional variable nodes, i.e., $l_{\rm avg} = 1/\int_0^1 \lambda(x)dx$. The number of sockets of additional variable nodes is expressed as $M_{\rm A} l_{\rm avg}$. Since all sockets of additional variable nodes are randomly connected to the first $2z$ check nodes, the following equation is satisfied as

$$M_{\rm A} l_{\rm avg} = 4z + 2z.$$

Then, the design rate of the $(3, 6, L, \lambda(x))$ ensemble is given as

$$R_{(3,6,L,\lambda(x))} = 1 - \frac{(L+1)z}{2Lz + M_{\rm A}} = \frac{1}{2} - \frac{1}{2}\frac{1 - 3/l_{\rm avg}}{L + 3/l_{\rm avg}} \tag{3}$$

where the rate-loss $\Delta R_{(3,6,L,\lambda(x))}$ is $(1 - 3/l_{\rm avg})/(2L + 6/l_{\rm avg})$, which is lower than $\Delta R_{(3,6,L)}$ for $l_{\rm avg} > 0$. The rate-loss $\Delta R_{(3,6,L,\lambda(x))}$ is reduced as $M_{\rm A}$ increases, or equivalently, as $l_{\rm avg}$ decreases. Let $\gamma_{\mathcal{A}}$ be the mitigation ratio of ensemble $\mathcal{A}$ defined by

$$\gamma_{\mathcal{A}} \triangleq \frac{\Delta R_{(3,6,L)} - \Delta R_{\mathcal{A}}}{\Delta R_{(3,6,L)}},$$

which will be used as a comparison measure of the rate-loss reduction.

C. DE Equations of the (3,6,L, $\lambda(x)$) Ensemble

In this paper, the channel is assumed to be the binary erasure channel (BEC) with erasure probability ϵ. Consider the base matrix $\mathbf{B}$ represented as

$$\mathbf{B} = \overbrace{\begin{bmatrix} 4 & 1 & 1 & & & & & \\ 2 & 1 & 1 & \ddots & & & & \\ & 1 & 1 & \ddots & 1 & 1 & & \\ & & & \ddots & 1 & 1 & 1 & 1 \\ & & & & 1 & 1 & 2 & 2 \end{bmatrix}}^{2L+1}, \tag{4}$$

which is a concatenation of the base matrix in (1) and $[4\,2\,0\ldots 0]^{\rm T}$. In terms of the protograph representation, the base matrix $\mathbf{B}$ corresponds to the $(3, 6, L, x^5)$ ensemble with z additional variable nodes of degree 6. However, the base matrix $\mathbf{B}$ is used not only to represent a specific ensemble, but also to derive the DE equations of the $(3, 6, L, \lambda(x))$ ensemble with general $\lambda(x)$ in a simple form. Note that, when constructing a code instance from the $(3, 6, L, \lambda(x))$ ensemble, we do not lift the base matrix $\mathbf{B}$ but follow the construction method described in Section II-B.

Let N_i denote the set of column indices of non-zero elements in the ith row in $\mathbf{B}$ and M_j denote the set of row indices of non-zero elements in the jth column in $\mathbf{B}$. Also, let $y_{i,j}^{(\ell)}$ denote the average erasure probability of messages sent from the ith check node to the jth variable node in $\mathbf{B}$ at iteration ℓ. Likewise, let $x_{i,j}^{(\ell)}$ denote the average erasure probability of messages sent from the jth variable node to the ith check node in $\mathbf{B}$ at iteration ℓ and $x_j^{(\ell)}$ denote the erasure probability of the jth variable node in $\mathbf{B}$ at iteration ℓ. Note that the first variable node in $\mathbf{B}$ corresponds to additional variable nodes. Set the initial conditions as $x_{i,j}^{(0)} = \epsilon$ if $B_{i,j} \neq 0$. Then, the evolution of $y_{i,j}^{(\ell)}$, $x_{i,j}^{(\ell)}$, and $x_j^{(\ell)}$ can be expressed as

$$\begin{aligned} y_{i,j}^{(\ell)} &= 1 - \prod_{j' \in N_i} \left(1 - x_{i,j'}^{(\ell-1)}\right)^{B_{i,j'} - \delta_{j,j'}} \\ x_{i,j}^{(\ell)} &= \begin{cases} \epsilon\lambda\left(\sum_{i'} \frac{B_{i',j} y_{i',j}^{(\ell)}}{\sum_{i''} B_{i'',j}}\right), & \text{for } (i,j) = (1,1) \text{ and } (2,1) \\ \epsilon \prod_{i' \in M_j} \left(y_{i',j}^{(\ell)}\right)^{B_{i',j} - \delta_{i,i'}}, & \text{otherwise} \end{cases} \\ x_j^{(\ell)} &= \begin{cases} \epsilon L\left(\sum_{i'} \frac{B_{i',j} y_{i',j}^{(\ell)}}{\sum_{i''} B_{i'',j}}\right), & \text{for } j = 1 \\ \epsilon \prod_{i' \in M_j} \left(y_{i',j}^{(\ell)}\right)^{B_{i',j}}, & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

where δ represented the Kronecker delta function and $L(x) = \int_0^x \lambda(z)\ dz / \int_0^1 \lambda(z)\ dz$ [19]. The BP threshold $\epsilon_{(3,6,L,\lambda(x))}$ is defined as the maximum ϵ for which $x_j^{(\ell)}$ goes to zero for all j as ℓ increases. Likewise, the BP threshold $\epsilon_{(3,6,L)}$ of the $(3, 6, L)$ ensemble is obtained in the same way by setting the erasure probability of messages from additional variable nodes equal to zero, i.e., $x_{i,j}^{(\ell)} = 0$ for $j = 1$. Note that $\epsilon_{(3,6,L)}$ for $L \geq 11$ converges to $\epsilon_{\text{SC}} \triangleq 0.4881$ [2], [3].[1]

III. Optimization of Degree Distribution $\lambda(x)$

A. Optimization of Degree Distribution $\lambda(x)$ in Previous Work

The design objective of the $(3, 6, L, \lambda(x))$ SC-LDPC ensemble is to optimize degree distribution $\lambda(x)$ that maximize $\gamma_{(3,6,L,\lambda(x))}$, or equivalently, minimize l_{avg} while preserving the performance of the $(3, 6, L)$ SC-LDPC ensemble. The approach proposed in [11] is to minimize l_{avg} with the constraint of $\epsilon_{(3,6,L,\lambda(x))} = \epsilon_{\text{SC}}$. The optimization problem is solved by linear programming and the resulting degree distribution $\lambda_1(x)$ is obtained as

$$\lambda_1(x) = 0.0193x^2 + 0.3439x^3 + 0.5310x^6 + 0.1058x^7.$$

The average variable node degree of $\lambda_1(x)$ is computed as $l_{\text{avg}} = 5.5117$ and $\gamma_{(3,6,20,\lambda_1(x))} = 0.5564$, i.e., the rate-loss is mitigated by 55.6%. Since the BP threshold $\epsilon_{(3,6,L,\lambda_1(x))}$ is unchanged from the BP threshold $\epsilon_{(3,6,L)}$ under the BP threshold constraint, the $(3, 6, L, \lambda_1(x))$ ensemble is equivalent to the $(3, 6, L)$ ensemble in terms of the asymptotic performance. However, the equivalent asymptotic performance does not guarantee the equivalent finite-length performance.

Let $\mathbb{P}_{(3,6,L)}$ and $\mathbb{P}_{(3,6,L,\lambda(x))}$ be the BLERs of the $(3, 6, L)$ and $(3, 6, L, \lambda(x))$ ensembles, respectively, under windowed decoding [5] with window size $W = 12$. For the windowed decoding scheme of the $(3, 6, L, \lambda(x))$ ensemble, the window decoder receives the channel values of the first $(W - 1)M$ variable nodes in the $(3, 6, L)$ ensemble together with the channel values of the M_{A} additional variable nodes during the first windowed decoding. Then, the number of bits included in the first window becomes $(W-1)M+M_{\text{A}}$, which is lower than the number of bits WM included in the first window when decoding the $(3, 6, L)$ ensemble. Thus, it is fair to compare the $(3, 6, L)$ and $(3, 6, L, \lambda(x))$ ensembles in terms of windowed decoding. After this initial decoding process, the windowed decoding scheme of the $(3, 6, L, \lambda(x))$ ensemble follows the conventional windowed decoding scheme. Note that we set the window size W to a sufficiently large value so that the windowed decoding performance does not differ significantly from the BP decoding performance, which is required to utilize the scaling law of SC-LDPC codes in the next subsection.

Fig. 2 shows the BLERs of the $(3, 6, 20)$ and $(3, 6, 20, \lambda_1(x))$ SC-LDPC ensembles with $M = 1{,}000$, where the finite-length performance of the $(3, 6, 20, \lambda_1(x))$ ensemble is degraded compared to that of the $(3, 6, 20)$ ensemble. To investigate the cause of the performance degradation, we define the local BLER $\overline{\mathbb{P}}_{(3,6,L,\lambda(x))}$ as the probability that at least one of the additional variable nodes is not recovered. If all additional variable nodes are recovered during the decoding process of the $(3, 6, L, \lambda(x))$ ensemble, the probability that at least one of the remaining variable nodes in the main region is not recovered is equivalent to $\mathbb{P}_{(3,6,L)}$. Thus, $\mathbb{P}_{(3,6,L,\lambda(x))}$ is given as

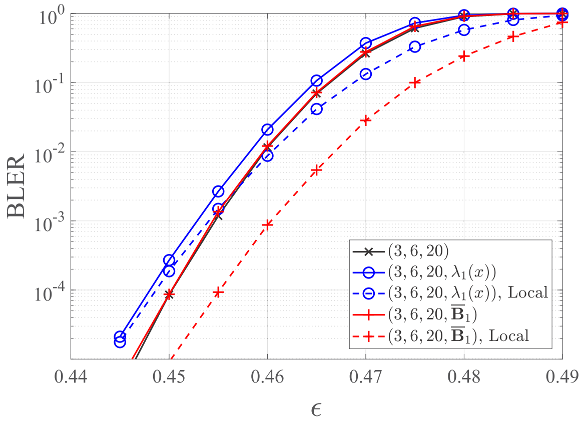


Fig. 2. Block erasure rates of the $(3, 6, 20)$ SC-LDPC code, the $(3, 6, 20, \lambda_1(x))$ SC-LDPC code in [11], and the proposed $(3, 6, 20, \overline{\mathbf{B}}_1)$ code along with their local BLERs.

$$\begin{aligned} \mathbb{P}_{(3,6,L,\lambda(x))} &= \overline{\mathbb{P}}_{(3,6,L,\lambda(x))} + \left(1 - \overline{\mathbb{P}}_{(3,6,L,\lambda(x))}\right) \mathbb{P}_{(3,6,L)} \\ &\approx \mathbb{P}_{(3,6,L)}, \text{ for } \overline{\mathbb{P}}_{(3,6,L,\lambda(x))} \ll \mathbb{P}_{(3,6,L)} \end{aligned} \tag{6}$$

where the last approximation is satisfied if the local BLER $\overline{\mathbb{P}}_{(3,6,L,\lambda(x))}$ is sufficiently lower than that of the $(3, 6, L)$ ensemble. In other words, the local performance of additional variable nodes should be much better than the performance of the main region in order to maintain the finite-length performance. However, Fig. 2 shows that $\overline{\mathbb{P}}_{(3,6,20,\lambda_1(x))}$ is instead higher than $\mathbb{P}_{(3,6,20)}$ for $\epsilon \leq 0.455$, which leads to the performance degradation of $\mathbb{P}_{(3,6,L,\lambda(x))}$. In contrast, the performance of the proposed $(3, 6, 20, \overline{\mathbf{B}}_1)$ code, which will be introduced in Section III-E, is nearly identical to that of the $(3, 6, 20)$ SC-LDPC code. In the following subsections, we will introduce the design procedure to obtain the proposed SC-LDPC codes.

B. Required Local BLER of Additional Variable Nodes

While minimizing the finite-length performance degradation, we aim to find $\lambda(x)$ for which mitigation ratio γ is maximized. To be specific, for given target BLER $\mathbb{P}^*$ and parameter k, we need to find $\lambda(x)$ achieving $\mathbb{P}_{(3,6,L,\lambda(x))} = k\mathbb{P}_{(3,6,L)} = k\mathbb{P}^*$, where $k > 1$ is set to a value near 1. The design procedure to find such $\lambda(x)$ is summarized below.

1) First, we obtain the target erasure probability ϵ^* at which $\mathbb{P}_{(3,6,L)} = \mathbb{P}^*$ using the scaling law of the SC-LDPC ensemble.
2) Second, the required local BLER $\overline{\mathbb{P}}^*$ to achieve $\mathbb{P}_{(3,6,L)} = \mathbb{P}^*$ is obtained from (8).
3) Finally, we can obtain $\lambda(x)$ to achieve $\overline{\mathbb{P}} = \overline{\mathbb{P}}^*$ at ϵ^* through differential evolution by adjusting an optimization parameter called the local threshold $\overline{\epsilon}_{(3,6,L,\lambda(x))}$.

[1] Since the BP threshold of the conventional SC-LDPC ensemble converges to the MAP threshold of underlying LDPC codes, 0.4881, for $L \geq 22$, the same effect arises for the $(3, 6, L)$ ensemble for $L \geq 11$.

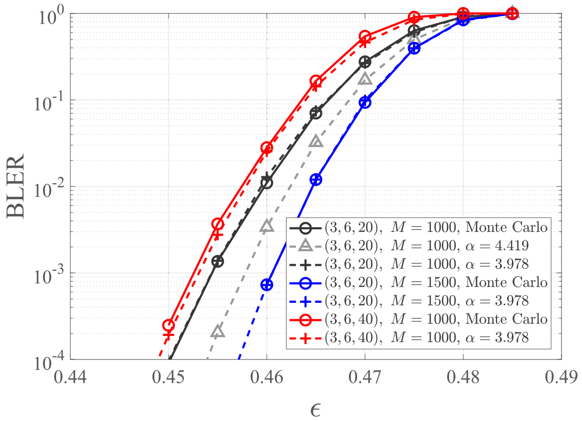


Fig. 3. Block erasure rates of the $(3,6,L)$ SC-LDPC codes for $(L,M)=(20,1000)$, $(L,M)=(20,1500)$, and $(L,M)=(40,1000)$ from Monte Carlo simulation (soild lines) along with their estimated performances (dashed lines) from the scaling law with scaling parameter $\alpha=4.419$ and $\alpha=3.978$.

The scaling law of the $(3,6,L)$ ensemble is described as [7], [14]

$$\mathbb{P}_{(3,6,L)} \approx \frac{\theta\epsilon L}{\sqrt{2\pi}\alpha\sqrt{M}(\epsilon_{\text{SC}}-\epsilon)}\exp\left(-M\frac{\alpha^2(\epsilon_{\text{SC}}-\epsilon)^2}{2}\right) \tag{7}$$

where α and θ are the scaling parameters. For the conventional SC-LDPC ensemble with two-sided decoding waves, the scaling parameters are given as $\alpha=6.25$ and $\theta=0.57$ [14]. For the $(3,6,L)$ ensemble with one-sided decoding wave, the scaling parameter α becomes $\alpha=6.25/\sqrt{2}\approx 4.419$ [7]. In Fig. 3, we plot the estimated BLER from the scaling law along with the actual performance computed by a Monte Carlo simulation for the $(3,6,20)$ ensemble with $M=1000$. As shown in Fig 3, the estimated performance with $\alpha=4.419$ and the actual performance differ somewhat because the derivation of the scaling parameter $\alpha=4.419$ in [14] is based on various assumptions resulting in an incorrect prediction of the actual performances. To capture the actual performance more correctly, we adjust the scaling parameter α by comparing the Monte Carlo simulation result. Fig. 3 shows that the estimated performance with $\alpha=0.9\times 4.419\approx 3.978$ can predict the actual performance more correctly. In addition, the refined scaling parameter $\alpha=3.978$ enables one to predict the actual performance for the other parameters such as $(L,M)=(20,1500)$ and $(L,M)=(40,1000)$. Thus, we use the refined scaling parameter $\alpha=3.978$ henceforth. After obtaining the refined alpha value from a Monte Carlo simulation of a code instance once, we can predict the finite-length performance for other code ensembles without additional Monte Carlo simulations. Thus, using the refined scaling law, we can obtain ϵ^* at which $\mathbb{P}_{(3,6,L)}=\mathbb{P}^*=10^{-4}$ as described in Table I. Note that ϵ^* decreases as L increases given that the finite-length performance of the $(3,6,L)$ ensemble is degraded as L increases.

Next, we compute the required local BLER $\overline{\mathbb{P}}^*$ such that $\mathbb{P}_{(3,6,L,\lambda(x))}$ achieves $k\mathbb{P}^*$ at erasure probability ϵ^*. From (8), the required local BLER $\overline{\mathbb{P}}^*$ is represented as

TABLE I

Erasure Probability ϵ^* at Which $\mathbb{P}_{(3,6,L)}=\mathbb{P}^*=10^{-4}$ for Various L and M

	$L=20$	$L=40$	$L=60$	$L=80$
$M=1000$	0.4500	0.4488	0.4482	0.4477
$M=1500$	0.4570	0.4560	0.4555	0.4551

$$\overline{\mathbb{P}}^* = \frac{\mathbb{P}_{(3,6,L,\lambda(x))}-\mathbb{P}_{(3,6,L)}}{1-\mathbb{P}_{(3,6,L)}} = \frac{(k-1)\mathbb{P}^*}{1-\mathbb{P}^*}. \tag{8}$$

Then our objective can be described as finding $\lambda(x)$ which achieves the required local BLER $\mathbb{P}^*$ at ϵ^*.

C. Minimizing the Rate-Loss While Achieving the Required Local BLER

In order to find $\lambda(x)$ achieving the required local BLER $\mathbb{P}^*$ at the target erasure probability ϵ^*, first we define the local threshold $\overline{\epsilon}_{(3,6,L,\lambda(x))}$ of the $(3,6,L,\lambda(x))$ ensemble as the maximum ϵ for which the erasure probability $x_1^{(\ell)}$ of additional variable nodes goes to below a small value such as 10^{-3} as ℓ increases. As the BP threshold $\epsilon_{(3,6,L,\lambda(x))}$ represents the asymptotic performance of the overall codeword, the local threshold $\overline{\epsilon}_{(3,6,L,\lambda(x))}$ is a performance measure specialized for additional variable nodes. In general, $\overline{\epsilon}_{(3,6,L,\lambda(x))}$ needs to be higher than ϵ_{SC} to achieve the required local BLER $\overline{\mathbb{P}}^* \ll \mathbb{P}^*$. However, for $\lambda_1(x)$, the local threshold $\overline{\epsilon}_{(3,6,L,\lambda_1(x))}$ is equal to $\epsilon_{(3,6,L)}=0.4881$, which implies that $\lambda_1(x)$ is not a proper degree distribution to achieve the required local BLER $\overline{\mathbb{P}}^*$.

A differential evolution algorithm, presented in Algorithm 1, is proposed to find a degree distribution achieving the input local threshold $\overline{\epsilon}$. Similar to the differential evolution algorithms in [20]–[22], Algorithm 1 is initialized with a set of randomly generated degree distributions and generate the improved set of degree distributions through the mutation, crossover, and selection steps. We apply the method of DE/rand/1/bin with per-vector dither [23] for the mutation step as the differential evolution algorithm in [22]. During the selection step, a trial degree distribution is selected if it satisfies two constraints which one is related to the local threshold and the other is related to average variable node degree. Due to the constraint on the average variable node degree, a degree distribution with a lower average degree is generated as the generation progresses.

As a running example, we set $L=20$, $M=1000$, $\mathbb{P}^*=10^{-4}$, and $k=1.1$. From Table I, ϵ^* is 0.45 and the required local BLER $\overline{\mathbb{P}}^*$ becomes roughly 10^{-5} by (8). By varying the input local threshold $\overline{\epsilon}$ of Algorithm 1, we can find a degree distribution of additional variable nodes achieving $\overline{\mathbb{P}}^*$ at ϵ^*. In detail, we increase $\overline{\epsilon}$ by 0.005 from ϵ_{SC} until the obtained degree distribution achieves $\overline{\mathbb{P}}^*$ at ϵ^*. The other input parameters are fixed as $N_p=100, G=10{,}000, l_{\min}=3, l_{\max}=10, F=0.5, p_c=0.85$. For $\overline{\epsilon}=0.4936$, we obtain the degree distribution $\lambda_2(x)=0.7009x^5+0.2991x^6$ from Algorithm 1, where the local BLER $\overline{\mathbb{P}}_{(3,6,20,\lambda_2(x))}$ is lower

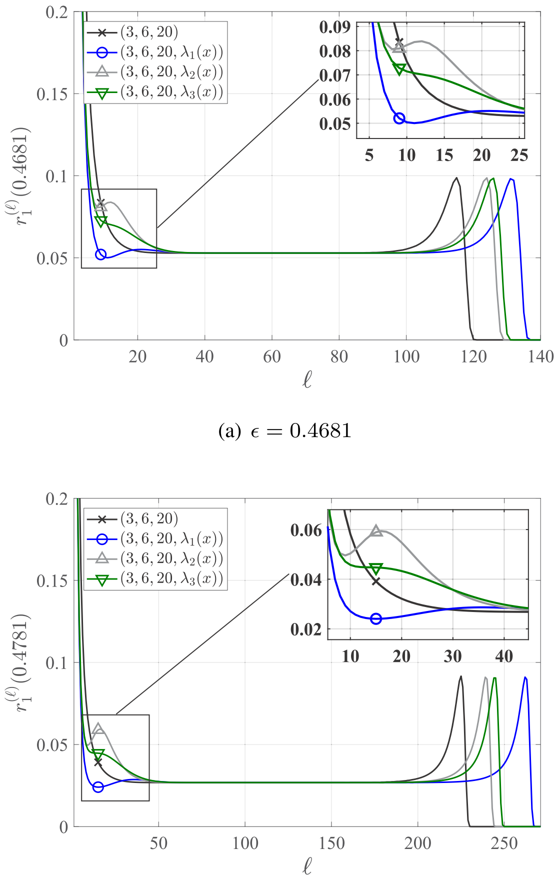
Algorithm 1 Differential Evolution Algorithm to Design $\lambda(x)$

```
Input: $N_p, G, l_{\min}, l_{\max}, F, p_c, \overline{\epsilon}$
Initialization: Generate a set of degree distributions, $\{\lambda_1(x), \ldots, \lambda_{N_p}(x)\}$, randomly with minimum degree $l_{\min}$ and maximum degree $l_{\max}$, where $\lambda_i(x) = \sum_{j=l_{\min}}^{l_{\max}} \lambda_{i,j} x^{j-1}$.
for $g = 1 : G$ do
    Mutation: For each $i \in \{1, \ldots, N_p\}$, generate a mutant polynomial $m_i(x) = \sum_{j=l_{\min}}^{l_{\max}} m_{i,j} x^{j-1}$ as
        $m_i(x) = \lambda_{r_1}(x) + (F + \beta(1-F))(\lambda_{r_2}(x) - \lambda_{r_3}(x))$
    where $r_1, r_2$, and $r_3$ are randomly-chosen distinct values in the range $\{1, \ldots, N_P\}$ and $\beta$ is a random variable distributed uniformly on $[0,1]$. If a coefficient of $m_i(x)$ is negative, it is set to be zero.
    Crossover: For each $i \in \{1, \ldots, N_p\}$, generate a trial degree distribution $t_i(x) = \sum_{j=l_{\min}}^{l_{\max}} t_{i,j} x^{j-1}$ as
        $t_{i,j} = \begin{cases} \lambda_{i,j} \text{ with probability } p_c \\ m_{i,j} \text{ with probability } 1-p_c \end{cases}$
    for $j \in \{l_{\min}, \ldots, l_{\max}\}$. If $t_i(1)$ is not 1, we randomly select non-zero coefficients of $t_i(x)$ and adjust the coefficients to satisfy $t_i(1) = 1$.
    Selection: Using the DE equations (5), for each $t_i(x)$, check the following constraints:
        i) $x_1^{(\ell)}$ goes to below $10^{-3}$ at input local threshold $\overline{\epsilon}$.
        ii) $1/\int_0^1 t_i(x)dx < 1/\int_0^1 \lambda_i(x)dx$.
    If the constraints are satisfied, then set $\lambda_i(x) = t_i(x)$.
end for
Select $\lambda_i(x)$ such that $1/\int_0^1 \lambda_i(x)dx$ is the minimum.
```

than $\overline{\mathbb{P}}^* = 10^{-5}$ at $\epsilon = \epsilon^* = 0.45$. However, the corresponding mitigation ratio $\gamma_{(3,6,20,\lambda_2(x))}$ is 0.4908, which is lower than $\gamma_{(3,6,20,\lambda_1(x))} = 0.5564$. It means that the $(3,6,L,\lambda_2(x))$ ensemble achieves the required local BLER at the cost of the reduced mitigation ratio compared to the results for $\lambda_1(x)$.

D. Local Minimum Constraint for Optimization Algorithm

In this subsection, we show that the trade-off between the local BLER and the mitigation ratio can be improved by using the expected graph evolution. The BP and local thresholds are the upper limit of the channel parameters for which decoding is successful. In contrast, the expected graph evolution [15] shows the decoding behavior for a given channel parameter, which can be utilized in an analysis of the finite-length performance. In [7], the expected number of degree-1 check nodes under peeling decoding is obtained by solving a system of differential equations. Also, in [24], the upper bound of the expected number of degree-1 check nodes is obtained from DE, which requires less computational complexity compared to solving a system of differential equations. Due to the low computational complexity, a constraint obtained from the expected graph evolution can be used for the proposed optimization algorithm.



(a) $\epsilon = 0.4681$

(b) $\epsilon = 0.4781$

Fig. 4. Evolution of $r_1^{(\ell)}(\epsilon)$ of the $(3,6,20)$ and $(3,6,20,\lambda(x))$ SC-LDPC ensembles with $\lambda_1(x)$, $\lambda_2(x)$, and $\lambda_3(x)$ at $\epsilon = 0.4681$ and $\epsilon = 0.4781$.

Let $r_1^{(\ell)}(\epsilon)$ be the upper bound of the normalized number of degree-1 check nodes connected to variable nodes in the main region of the $(3,6,L,\lambda(x))$ ensemble at iteration ℓ with erasure probability ϵ under BP decoding. Then $r_1^{(\ell)}(\epsilon)$ is given as [14]

$$r_1^{(\ell)}(\epsilon) = \sum_{j=2}^{2L+1} x_j^{(\ell-1)} - \sum_{j=2}^{2L+1} x_j^{(\ell)}$$

where $x_j^{(\ell)}$ is obtained by the DE equations in (5). Fig. 4 shows the evolution of $r_1^{(\ell)}(\epsilon)$ at $\epsilon = \epsilon_{\text{SC}} - 2 \times 10^{-2} = 0.4681$ and $\epsilon = \epsilon_{\text{SC}} - 10^{-2} = 0.4781$, respectively. For the $(3,6,20)$ ensemble, there exist three phases in the evolution of $r_1^{(\ell)}(\epsilon)$; the initial phase, the critical phase, and the third phase. In [7], the scaling law of SC-LDPC codes is derived with the assumption that decoding failures rarely occur in the initial phase. The assumption is justified considering the property that the value of $r_1^{(\ell)}(\epsilon)$ in the initial phase decreases steadily to the value of the critical phase. However, for the $(3,6,20,\lambda_1(x))$ and $(3,6,20,\lambda_2(x))$ ensembles, a local minimum appears in the initial phase as an additional factor of decoding failures,

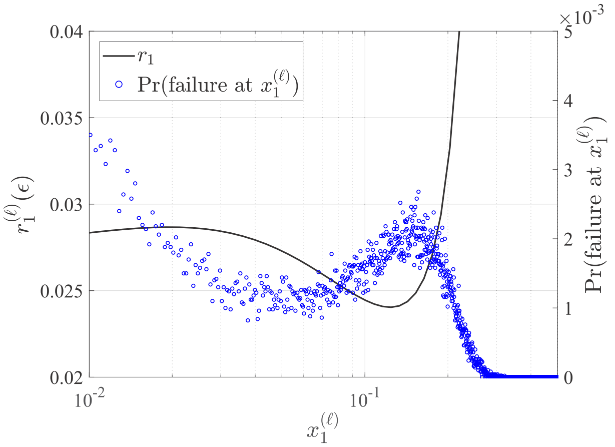


Fig. 5. $r_1^{(\ell)}(\epsilon)$ and $\mathrm{Pr}(\text{failure at } x_1^{(\ell)})$ against $x_1^{(\ell)}$ at $\epsilon = 0.4781$ for the $(3, 6, 20, \lambda_1(x))$ SC-LDPC ensemble.

which implies that the finite-length performance is predicted to be degraded.

The predication of the finite-length performance actually requires not only the expected value of $r_1^{(\ell)}(\epsilon)$ but also the second order moments of $r_1^{(\ell)}(\epsilon)$ [7]. However, computing the second order moments requires a large amount of computations, which is undesirable for the optimization process. Moreover, the expectation of $r_1^{(\ell)}(\epsilon)$ is sufficient to find a design condition affecting decoding failures. In order to verify that the emerged local minimum becomes a factor of decoding failures, we plot the evolution of $r_1^{(\ell)}(0.4781)$ as a function of $x_1^{(\ell)}$ for the $(3, 6, 20, \lambda_1(x))$ ensemble in Fig. 5. In addition, we compute the empirical probability $\mathrm{Pr}(\text{failure at } x_1^{(\ell)})$, which is defined as the probability that the average erasure probability of additional variable nodes is equal to $x_1^{(\ell)}$ when the decoder declares a decoding failure in the finite-length code simulation. In Fig. 5, we plot $\mathrm{Pr}(\text{failure at } x_1^{(\ell)})$ with a different y-axis, where $\mathrm{Pr}(\text{failure at } x_1^{(\ell)})$ is obtained from 10^5 transmissions of the $(3, 6, 20, \lambda_1(x))$ ensemble with $z = 2{,}000$. By comparing the two plots in Fig. 5, we can confirm that decoding failures are likely to occur at the local minimum of $r_1^{(\ell)}(\epsilon)$, which implies that the local minimum is a significant factor of decoding failures.

We define the local minimum constraint such that $r_1^{(\ell)}(\epsilon)$ should not have a local minimum in the initial phase for $\epsilon < \epsilon_{(3,6,L)}$. By including the local minimum constraint in the selection step of Algorithm 1,[2] the optimized degree distribution $\lambda_3(x)$ achieving the required local BLER $\overline{\mathbb{P}}^*$ can be obtained with the local threshold $\overline{\epsilon} = 0.4901$ as

$$\lambda_3(x) = 0.01067x^3 + 0.63926x^4 + 0.35007x^9$$

and $\gamma_{(3,6,20,\lambda_3(x))} = 0.5088$. Fig. 4 shows that a local minimum of $r_1^{(\ell)}(\epsilon)$ does not exist for the $(3, 6, 20, \lambda_3(x))$ ensemble. By removing the harmful factor affecting the finite-length performance, the mitigation ratio $\gamma_{(3,6,20,\lambda_3(x))} = 0.5088$ is improved compared to $\gamma_{(3,6,20,\lambda_2(x))} = 0.4908$. In the next subsection, a design method of attached additional variable nodes is introduced to increase γ even further by employing a protograph structure.

E. Optimizing the (3,6,L,$\overline{\mathbf{B}}$) Ensemble

As noted in Section II-C, the base matrix in (4) is used to represent the DE equations of the $(3, 6, L, \lambda(x))$ ensemble in a simple form. However, if the base matrix in (4) is lifted with lifting factor z, the resulting codes correspond to the $(3, 6, L, x^5)$ ensemble with z additional variable nodes of degree 6. In other words, the node and edge distributions of additional variable nodes can be represented by a base matrix instead of $\lambda(x)$. Let $\mathbf{B}'$ denote the attached base matrix determining the structure of additional variable nodes. For example, $\mathbf{B}' = [4\,2\,0\,\ldots\,0]^{\mathrm{T}}$ for the base matrix in (4). By changing the non-zero entries of $\mathbf{B}'$, we can optimize the structure for additional variable nodes. For $\mathbf{B}' = [4\,2\,0\,\ldots\,0]^{\mathrm{T}}$, however, the number of non-zero entries to be optimized is only two. In order to increase the search space, we first consider the pre-lifting base matrix of (1) with lifting factor z_1, where $z = z_1 \times z_2$, after which we attach $\mathbf{B}'$ of size $2z_1 \times z_{\mathrm{A}}$ to the pre-lifted base matrix as

$$\left[\begin{array}{c:c:cc:cc:c}
1 & \mathbf{P} & & & & & \\
\hdashline
 & \mathbf{B}' & \Pi_{1,1} & \Pi_{1,2} & & & \\
 & & \Pi_{2,1} & \Pi_{2,2} & \Pi_{2,3} & \Pi_{2,4} & \\
\hdashline
 & & \Pi_{3,1} & \Pi_{3,2} & \Pi_{3,3} & \Pi_{3,4} & \ddots \\
 & & & & \Pi_{4,3} & \Pi_{4,4} & \ddots \\
 & & & & & & \ddots
\end{array}\right] \tag{9}$$

where $\mathbf{P}$ is a matrix of size $1 \times z_{\mathrm{A}}$ and $\Pi_{i,j}$ is a random permutation matrix of size $z_1 \times z_1$. Note that we include a degree-1 variable node using the precoding technique [16] while the last variable node in $\mathbf{B}'$ is punctured to maintain the design rate. Let $\overline{\mathbf{B}} = [\mathbf{P}; \mathbf{B}']$ and the $(3, 6, L, \overline{\mathbf{B}})$ SC-LDPC ensemble denote the ensemble corresponding to (9). Since the matrix size of (9) is $(1 + (L+1)z_1) \times (1 + z_{\mathrm{A}} + 2Lz_1)$, the design rate of the $(3, 6, L, \overline{\mathbf{B}})$ ensemble is given as

$$\begin{aligned}
R_{(3,6,L,\overline{\mathbf{B}})} &= 1 - \frac{(L+1)z_1z_2}{2Lz_1z_2 + z_{\mathrm{A}}z_2} \\
&= \frac{1}{2} - \frac{1}{2}\frac{1 - \frac{z_{\mathrm{A}}}{2z_1}}{L + \frac{z_{\mathrm{A}}}{2z_1}}.
\end{aligned}$$

Compared to the $(3, 6, L, \lambda(x))$ ensemble, the $(3, 6, L, \overline{\mathbf{B}})$ ensemble has advantages such as the practical advantage of using the protograph structure and the possibility of improving the local performance of additional variable nodes by employing degree-1 variable nodes. However, the number of additional variable nodes is restricted to multiples of z_{A} after lifting the base matrix (9), placing a restriction on the range of the mitigation ratio $\gamma_{(3,6,L,\overline{\mathbf{B}})}$.

To increase the available values of the mitigation ratio, we consider puncturing $\overline{\beta}$ fraction of the additional variable nodes. After lifting the base matrix in (9) with lifting factor z_2, the numbers of degree-1 variable nodes and additional variable nodes in $\overline{\mathbf{B}}$ are z_2 and $M_{\mathrm{A}} = z_{\mathrm{A}}z_2$, respectively. Then, $\overline{\beta}$ fraction of the z_2 degree-1 variable nodes and the

[2] To reduce the optimization complexity, we check the local minimum constraint at the two points of $\epsilon = \epsilon_{\mathrm{SC}} - 2 \times 10^{-2}$ and $\epsilon = \epsilon_{\mathrm{SC}} - 10^{-2}$.

first $(z_{\mathrm{A}}-1)z_2$ additional variable nodes are punctured, where punctured variable nodes are selected randomly. The number of randomly punctured variable nodes is $p=\overline{\beta}z_{\mathrm{A}}z_2$ and the resulting design rate of the $(3,6,L,\overline{\mathbf{B}})$ ensemble is given as

$$\begin{aligned} R_{(3,6,L,\overline{\mathbf{B}})} &= 1-\frac{(L+1)z_1z_2-p}{(2Lz_1z_2+z_{\mathrm{A}}z_2)-p} \\ &= \frac{1}{2}-\frac{1}{2}\frac{1-\frac{z_{\mathrm{A}}}{2z_1}-\overline{\beta}\frac{z_{\mathrm{A}}}{2z_1}}{L+\frac{z_{\mathrm{A}}}{2z_1}-\overline{\beta}\frac{z_{\mathrm{A}}}{2z_1}}. \end{aligned} \tag{10}$$

Using the DE equations for protograph based ensembles [3], the optimization of $\overline{\mathbf{B}}$ and $\overline{\beta}$ can be performed in the similar way of the optimization of $\lambda(x)$ as shown in Algorithm 2. For input local thresholds $\overline{\epsilon}=0.4901, \overline{\epsilon}=0.4896$, and $\overline{\epsilon}=0.4891$, the obtained base matrices $\overline{\mathbf{B}}_1$, $\overline{\mathbf{B}}_2$, and $\overline{\mathbf{B}}_3$ are given as

$$\overline{\mathbf{B}}_1=\begin{bmatrix} 1&0&0&0&2\\ 1&0&0&0&2\\ 0&1&0&0&2\\ 0&1&1&0&1\\ 0&0&0&2&1\\ 1&0&1&1&1\\ 1&0&2&1&0\\ 1&0&2&1&0\\ 2&2&0&0&0\\ 0&2&0&0&1\\ 0&0&0&0&2 \end{bmatrix}$$

$$\overline{\mathbf{B}}_2=\begin{bmatrix} 0&0&1&0&2\\ 1&1&0&2&0\\ 0&0&1&0&2\\ 0&1&0&0&2\\ 2&0&0&0&1\\ 0&2&0&0&1\\ 0&0&2&1&0\\ 1&0&1&1&1\\ 1&0&1&1&1\\ 0&0&0&1&1\\ 0&0&1&0&1 \end{bmatrix}$$

$$\overline{\mathbf{B}}_3=\begin{bmatrix} 1&0&0&0&2\\ 0&0&1&0&2\\ 1&1&1&0&1\\ 0&1&0&1&2\\ 1&0&0&0&2\\ 0&0&0&1&1\\ 0&0&1&0&1\\ 2&0&0&1&0\\ 0&2&0&1&1\\ 1&1&2&0&0\\ 1&0&0&2&0 \end{bmatrix}$$

where $z_1=5$ and $z_{\mathrm{A}}=5$. The corresponding $\overline{\beta}$'s for $\overline{\mathbf{B}}_1$, $\overline{\mathbf{B}}_2$, and $\overline{\mathbf{B}}_3$ are $\overline{\beta}_1=0.05$ $\overline{\beta}_2=0.07$, and $\overline{\beta}_3=0.09$, respectively. The $(3,6,L,\overline{\mathbf{B}}_1)$ ensemble is comparable to the $(3,6,L,\lambda_3(x))$ ensemble because their local thresholds are equivalent and both ensembles are optimized with the local minimum constraint. However, mitigation ratio $\gamma_{(3,6,20,\overline{\mathbf{B}}_1)}=0.5360$ is improved relative to $\gamma_{(3,6,20,\lambda_3(x))}=0.5088$. Compared to the $(3,6,L,\lambda_1(x))$ ensemble, mitigation ratio $\gamma_{(3,6,20,\overline{\mathbf{B}}_1)}=0.5360$ is slightly lower than $\gamma_{(3,6,20,\lambda_1(x))}=0.5564$. However, as shown in Fig. 2, the BLER of the $(3,6,20,\overline{\mathbf{B}}_1)$ ensemble is nearly identical to that of the $(3,6,20)$ ensemble unlike the $(3,6,20,\lambda_1(x))$ ensemble. Table II displays γ together with the ratio of the BLER of ensemble $\mathcal{A}$ to that of the $(3,6,20)$ ensemble at $\epsilon=0.45$. In Table II, we compare the ensembles introduced so far. While the $(3,6,20,\lambda_1(x))$ ensemble proposed in [11] shows significant performance degradation, the performance loss is not noticeable for the ensembles proposed in this paper. The performance is maintained by introducing the local threshold and differential evolution algorithms. Comparing the $(3,6,20,\lambda_1(x))$ ensemble and $(3,6,L,\lambda_2(x))$ ensemble shows that there is a trade-off between the rate-loss mitigation ratio and the finite-length performance. The trade-off is improved for the $(3,6,L,\lambda_3(x))$ ensemble by introducing the expected graph evolution and for the $(3,6,L,\overline{\mathbf{B}}_1)$ ensemble by introducing the protograph structure.

Algorithm 2 Design Algorithm of $\overline{\mathbf{B}}$

```
Input: $N_p, G, F, l_{\min}, l_{\max}, e_{\max}, p_c, \overline{\epsilon}^*$
1: Initialization: For generation $g=0$, generate a set of base matrices of size $2z_1\times z_1$, $\{\mathbf{B}_1,\ldots,\mathbf{B}_{N_p}\}$, where each entry of $\mathbf{B}_t$ is randomly chosen in the set of $\{0,\ldots,e_{\max}\}$. For $t\in\{1,\ldots,N_p\}$, obtain the local threshold $\overline{\epsilon}(\mathbf{B}_t)$ and the local minimum $r_1^*(\epsilon,\mathbf{B}_t)$ of $r_1^{(\ell)}(\epsilon)$ in the initial phase, where set $r_1^*(\epsilon,\mathbf{B}_t)=1$ if there is no local minimum. If $\overline{\epsilon}(\mathbf{B}_t)\geq\overline{\epsilon}^*$, find the maximum puncturing fraction $\beta_{\mathrm{A}}(\mathbf{B}_t)$ to achieve $\overline{\epsilon}(\mathbf{B}_t)=\overline{\epsilon}^*$.
2: for $g=1:G$ do
3:   Mutation: For each $t\in\{1,\ldots,N_p\}$, generate a mutant matrix $\mathbf{M}_t$ as follows:
       $[\mathbf{M}_t]_{i,j}=[\mathbf{B}_{r_1}]_{i,j}+(F+\beta(1-F))([\mathbf{B}_{r_2}]_{i,j}-[\mathbf{B}_{r_3}]_{i,j})$
     where $[\mathbf{X}]_{i,j}$ denote the $(i,j)$ component of matrix $\mathbf{X}$. If a entry of $\mathbf{M}_t$ is not integer or larger than $e_{\max}$, the entry is replaced by the closet integer value among $\{0,1,\ldots,e_{\max}\}$.
4:   Crossover: For each $t\in\{1,\ldots,N_p\}$, generate a trial matrix $\mathbf{T}_t$ as follows:
       $[\mathbf{T}_t]_{i,j}=\begin{cases}[\mathbf{B}_t]_{i,j} \text{ with probability } p_c\\ [\mathbf{M}_t]_{i,j} \text{ with probability } 1-p_c.\end{cases}$
5:   Selection: For each $\mathbf{T}_t$, obtain the local threshold $\overline{\epsilon}(\mathbf{T}_t)$ and the local minimum $r_1^*(\epsilon,\mathbf{T}_t)$. If $\overline{\epsilon}(\mathbf{T}_t)\geq\overline{\epsilon}^*$, find the maximum puncturing fraction $\beta_{\mathrm{A}}(\mathbf{T}_t)$ to achieve $\overline{\epsilon}(\mathbf{T}_t)=\overline{\epsilon}^*$. Update $\mathbf{B}_t$ to $\mathbf{T}_t$ when one of the following conditions is satisfied.
       1) If $r_1^*(\epsilon,\mathbf{B}_t)\neq 1, r_1^*(\epsilon,\mathbf{T}_t)>r_1^*(\epsilon,\mathbf{B}_t)$.
       2) If $r_1^*(\epsilon,\mathbf{B}_t)=1, \overline{\epsilon}(\mathbf{B}_t)<\overline{\epsilon}^*, \overline{\epsilon}(\mathbf{T}_t)>\overline{\epsilon}(\mathbf{B}_t)$.
       3) If $r_1^*(\epsilon,\mathbf{B}_t)=1, \overline{\epsilon}(\mathbf{B}_t)\geq\overline{\epsilon}^*, \beta_{\mathrm{A}}(\mathbf{T}_t)>\beta_{\mathrm{A}}(\mathbf{B}_t)$.
6: end for
   Select $\mathbf{B}_t$ such that $\beta_{\mathrm{A}}(\mathbf{B}_t)$ is the maximum.
```

By following the same design procedure used when $L=20$ and $M=1000$, we can optimize the $(3,6,L,\overline{\mathbf{B}})$ ensemble for other parameters of L and M. Note that the optimized results

TABLE II

COMPARING THE $(3, 6, 20, \lambda_1(x))$, $(3, 6, 20, \lambda_2(x))$, $(3, 6, 20, \lambda_3(x))$, AND $(3, 6, 20, \overline{\mathbf{B}}_1)$ SC-LDPC ENSEMBLES WITH RESPECT TO MITIGATION RATIO γ AND THE RATIO OF BLERS AT $\epsilon = 0.45$

Ensemble $\mathcal{A}$	$\gamma_{\mathcal{A}}$	$\mathbb{P}_{\mathcal{A}}/\mathbb{P}_{(3,6,20)}$
$(3, 6, 20, \lambda_1(x))$ [11]	0.5564	3.14
$(3, 6, 20, \lambda_2(x))$	0.4908	1.01
$(3, 6, 20, \lambda_3(x))$	0.5088	1.01
$(3, 6, 20, \overline{\mathbf{B}}_1)$	0.5360	1.01

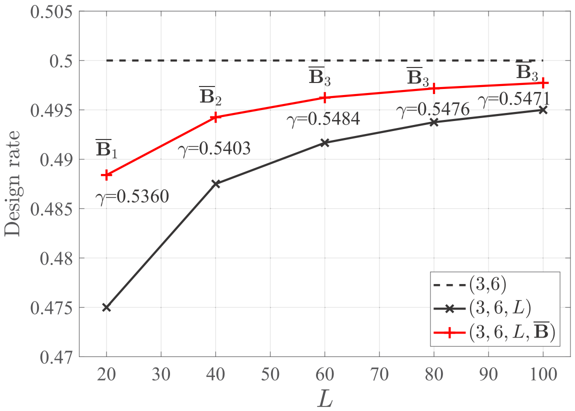


Fig. 6. Design rates of the $(3, 6)$ regular LDPC ensemble, the $(3, 6, L)$ SC-LDPC ensemble, and the $(3, 6, L, \overline{\mathbf{B}})$ SC-LDPC ensemble along with optimized $\overline{\mathbf{B}}$ and corresponding γ for various values of L when $M = 1000$.

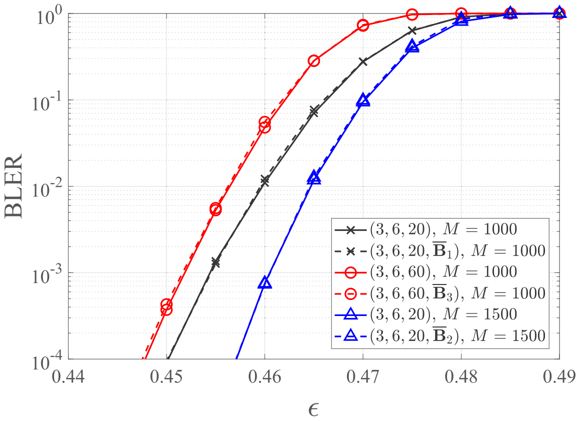


Fig. 7. Block error rates of the $(3, 6, L)$ SC-LDPC codes and the proposed $(3, 6, 20, \overline{\mathbf{B}})$ codes for $(L, M) = (20, 1000)$, $(L, M) = (60, 1000)$, and $(L, M) = (20, 1500)$.

depend on L and M, as target erasure probability ϵ^* changes according to the selection of L and M. As an example, for $L = 60$ and $M = 1000$, the required local threshold is 0.4891, which is lower than the required local threshold 0.4901 for the case when $(L, M) = (20, 1000)$. Thus, we select $\overline{\mathbf{B}}_3$ for the $(3, 6, 60, \overline{\mathbf{B}})$ ensemble instead of $\overline{\mathbf{B}}_1$. Fig. 7 shows that performance degradation is also not observed for the case when $(L, M) = (60, 1000)$. In Fig. 6, we plot the design rates of the $(3, 6, L)$ and the $(3, 6, L, \overline{\mathbf{B}})$ ensembles for various values of L. At each point of the design rate of the optimized $(3, 6, L, \overline{\mathbf{B}})$ ensemble, the corresponding $\overline{\mathbf{B}}$ and $\gamma_{(3,6,L,\overline{\mathbf{B}})}$ are shown, where the mitigation ratio is higher than 0.53 for all values of L. In other words, the rate-loss is mitigated by more than 53% while the performance degradation is not noticeable for the both case $(L, M) = (20, 1000)$ and $(60, 1000)$. In addition, Fig. 7 shows that the same result is obtained for $(L, M) = (20, 1500)$.

Remark 1: Comparing the conventional SC-LDPC ensemble and the proposed SC-LDPC ensemble in terms of degree distribution, one can notice that two SC-LDPC ensembles have different degree distributions. While the conventional SC-LDPC ensemble has boundary low-degree check nodes, where the wave-like propagation of reliable message is triggered [3], the boundary check nodes for the proposed SC-LDPC ensemble is no longer low-degree because of the attached additional variable nodes. Only after recovering the additional variable nodes, the degree of boundary check nodes becomes low as the conventional SC-LDPC codes. In this point of view, the design objective of additional variable nodes can be expressed by adding variable nodes as much as possible while ensuring recovering of additional variable nodes. In order to do that, we first derived the local BLER of additional variable nodes required for the correct decoding without performance degradation in Section III-B, and then we found the degree distribution of additional variable nodes that satisfies the required local BLER in Section III-C. Thus, additional variable nodes with the obtained degree distribution are correctly decoded without scarifying the decoding performance and the wave-like propagation is triggered as the conventional SC-LDPC codes. In addition, a design constraint called local minimum constraint is proposed to further optimize the additional variable nodes while guaranteeing the correct decoding in Section III-D.

Remark 2: Note that our proposed design method can be applied to other SC-LDPC codes as long as the degree of boundary check nodes is lower than other check nodes. Thus, the rate-loss of other SC-LDPC codes such as irregular SC-LDPC codes [3] and non-uniformly coupled SC-LDPC codes [25] can be mitigated by using the proposed design method in a similar manner.

IV. PERFORMANCE COMPARISON

Thus far, the performances of the $(3, 6, L)$ and $(3, 6, L, \overline{\mathbf{B}})$ ensembles have been compared for the same value of L since we have been focusing on whether or not performance degradation occurs by attaching additional variable nodes. In this section, the performances of the two ensembles are compared at the same design rate.

To achieve a given design rate, the $(3, 6, L)$ ensemble requires a large value of L compared to the $(3, 6, L, \overline{\mathbf{B}})$ ensemble. From the design rates in (2) and (10), the relationship between L_1 and L_2 satisfying $\Delta R_{(3,6,L_1)} = \Delta R_{(3,6,L_2,\overline{\mathbf{B}})}$ is given as

$$L_1 = \frac{L_2 + \frac{z_A}{2z_1} - \overline{\beta}\frac{z_A}{2z_1}}{1 - \frac{z_A}{2z_1} - \overline{\beta}\frac{z_A}{2z_1}}. \quad (11)$$

For example, the design rates of the $(3, 6, 43)$, $(3, 6, 87)$, and $(3, 6, 123)$ ensembles are almost the same as those of the $(3, 6, 20, \overline{\mathbf{B}}_1)$, $(3, 6, 40, \overline{\mathbf{B}}_2)$, and $(3, 6, 60, \overline{\mathbf{B}}_3)$ ensembles, respectively. In Fig. 8, the BLERs of each pair of ensembles are compared for $M = 1000$, which shows that the proposed

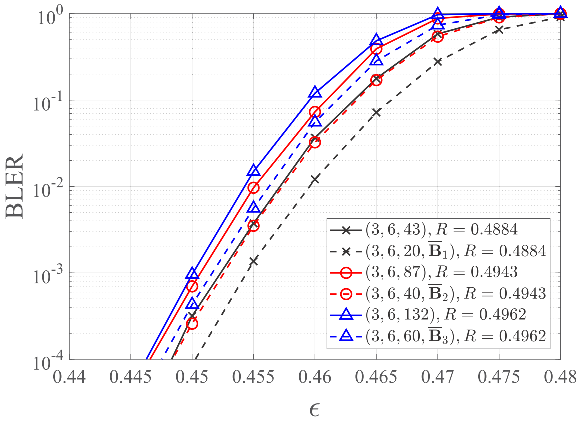


Fig. 8. Comparison of BLERs of the $(3,6,L)$ and $(3,6,L,\overline{\mathbf{B}})$ ensembles for different values of L and the same design rate.

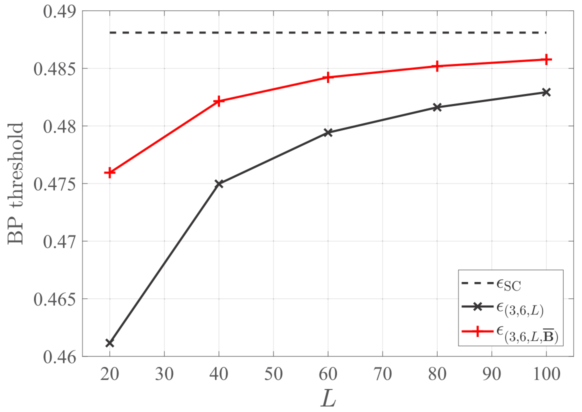


Fig. 9. BP thresholds of the punctured $(3,6,L)$ and $(3,6,L,\overline{\mathbf{B}})$ ensembles to achieve the target design rate 0.5 by puncturing.

$(3,6,L,\overline{\mathbf{B}})$ ensemble is superior to the conventional $(3,6,L)$ ensemble for all pairs. The performance improvement of the $(3,6,L,\overline{\mathbf{B}})$ ensemble comes from the small value of L because the BLER scales linearly with the chain length L [7]. We expect that the performance improvement observed in Fig. 8 continues for all pairs of L_1 and L_2 satisfying (11) because L_1 is a linear function of L_2.

Furthermore, the random puncturing technique in [17] can be used to match the design rate. Let β be the fraction of punctured bits for each position. For the additional variable nodes of the $(3,6,L,\overline{\mathbf{B}})$ ensemble, the total fraction of punctured bits becomes $\overline{\beta}+\beta$. According to [17], the design rate $R(\beta)$ and the BP threshold $\epsilon(\beta)$ of a punctured code with puncturing fraction β are expressed in terms of the design rate $R(0)$ and the BP threshold $\epsilon(0)$ of the unpunctured mother code as

$$\begin{aligned} R(\beta) &= \frac{R(0)}{1-\beta} \\ \epsilon(\beta) &= 1 - \frac{1-\epsilon(0)}{1-\beta}. \end{aligned} \tag{12}$$

Using (12) and the design rates of the $(3,6,L)$ ensemble and the $(3,6,L,\overline{\mathbf{B}})$ ensemble in (2) and (10), the required value of β to achieve target design rate R^* can be obtained for each value of L. From the obtained value of β, the BP thresholds of the punctured $(3,6,L)$ and $(3,6,L,\overline{\mathbf{B}})$ ensembles can also be obtained. Fig. 9 shows the numerically calculated BP thresholds of the punctured $(3,6,L)$ and $(3,6,L,\overline{\mathbf{B}})$ ensembles to achieve a target design rate $R^* = 0.5$ by puncturing. For the $(3,6,L,\overline{\mathbf{B}})$ ensemble, the required β is lower than that of the $(3,6,L)$ ensemble due to the reduced rate-loss, and accordingly the BP threshold of the $(3,6,L,\overline{\mathbf{B}})$ ensemble is much close to that of the corresponding unpunctured ensemble ϵ_{SC}. The high BP threshold directly influences the finite-length performance as shown in Fig. 10, where the punctured $(3,6,60)$ and $(3,6,60,\overline{\mathbf{B}}_3)$ codes are compared. As a reference, the performance of $(3,6)$ block LDPC code is included in Fig. 10. In addition, we include the decoding performance of the punctured $(3,6,60,\lambda_3(x))$ SC-LDPC code in Fig. 10, which is similar to the performance of the $(3,6,60,\overline{\mathbf{B}}_3)$ code. However, the $(3,6,60,\overline{\mathbf{B}}_3)$ code is more hardware friendly than the $(3,6,60,\lambda_3(x))$ code because additional variable nodes in the $(3,6,60,\overline{\mathbf{B}}_3)$ code is represented by the protograph representation. In addition, we include the performance of the $(3,7,12)$ SC-LDPC code in Fig. 10, which shows that constructing an SC-LDPC code from block LDPC codes with a slightly higher rate is not a proper way to achieve the target rate. Instead, the result shows that the best way to obtain the target rate is to puncture some variable nodes of the SC-LDPC code with additional variable nodes. Lastly, the performance of the punctured $(3,6,12)$ SC-LDPC code with code rate 0.5 is evaluated for a fair comparison, which does not show improved performance.

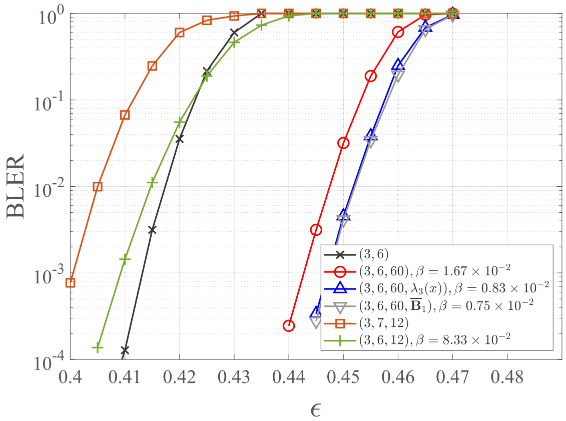


Fig. 10. Comparison of BLERs of the punctured $(3,6,60)$, $(3,6,60,\lambda_3(x))$, and $(3,6,60,\overline{\mathbf{B}})$ SC-LDPC codes with $M = 1{,}000$. In addition, the decoding performances of the $(3,6)$ block LDPC code, the $(3,7,12)$ SC-LDPC code, and the punctured $(3,6,12)$ SC-LDPC code are included. The code rate of all codes is equal to 0.5. The decoding of the SC-LDPC codes is performed by windowed decoding with $W = 12$ and constraint length 12,000.

V. Conclusion

In this paper, a design method is proposed to mitigate the rate-loss of SC-LDPC codes without degradation of the finite-length performance. Based on a comprehensive study on the finite-length performance of the SC-LDPC codes with additional variable nodes, we proposed the design method and obtained the optimized structure for additional variable nodes that significantly mitigate the rate-loss. For variable and check node degrees $(3,6)$ as an example, we showed that the rate-loss can be reduced by more than 53% while the finite-length performance is not degraded. In other words, the proposed design method increases the practical value of the SC-LDPC code in that it solves one of the major drawbacks of the SC-LDPC code.

References

[1] S. Kudekar, T. J. 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.

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

[3] 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.

[4] L. Schmalen, V. Aref, J. Cho, D. Suikat, D. Rösener, and A. Leven, "Spatially coupled soft-decision error correction for future lightwave systems," *J. Lightw. Technol.*, vol. 33, no. 5, pp. 1109–1116, Mar. 1, 2015.

[5] 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.

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

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

[8] Z. Yang, Y. Fang, G. Han, G. Cai, and F. C. M. Lau, "Design and analysis of punctured terminated spatially coupled protograph LDPC codes with small coupling lengths," *IEEE Access*, vol. 6, pp. 36723–36731, 2018.

[9] S. Kudekar, C. Méassony, T. J. Richardson, and R. Urbanke, "Threshold saturation on BMS channels via spatial coupling," in *Proc. 6th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Brest, France, Sep. 2010, pp. 309–313.

[10] S. Cammerer, V. Aref, L. Schmalen, and S. T. Brink, "Triggering wave-like convergence of tail-biting spatially coupled LDPC codes," in *Proc. Ann. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2016, pp. 93–98.

[11] M. R. Sanatkar and H. D. Pfister, "Increasing the rate of spatially-coupled codes via optimized irregular termination," in *Proc. 9th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Sep. 2016, pp. 31–35.

[12] H. Kwak, J. Kim, and J.-S. No, "Rate-loss reduction of SC-LDPC codes by optimizing reliable variable nodes via expected graph evolution," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 2930–2934.

[13] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.

[14] M. Stinner and P. M. Olmos, "On the waterfall performance of finite-length SC-LDPC codes constructed from protographs," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 345–361, Feb. 2016.

[15] M. G. Luby *et al.*, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[16] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[17] D. G. M. Mitchell, M. Lentmaier, A. E. Pusane, and D. J. Costello, "Randomly punctured LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 408–421, Feb. 2016.

[18] J. Thorpe, "Low density parity check (LDPC) codes constructed from protographs," JPL, California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 42-154, 2003.

[19] T. J. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, U.K.: Cambridege Univ. Press, 2008.

[20] A. Shokrollahi and R. Storn, "Design of efficient erasure codes with differential evolution," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2000, p. 5.

[21] A. K. Pradhan, A. Thangaraj, and A. Subramanian, "Construction of near-capacity protograph LDPC code sequences with block-error thresholds," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 27–37, Jan. 2016.

[22] C. Tang, M. Jiang, C. Zhao, and H. Shen, "Design of protograph-based LDPC codes with limited decoding complexity," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2570–2573, Dec. 2017.

[23] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[24] M. Stinner, L. Barletta, and P. M. Olmos, "Finite-length scaling based on belief propagation for spatially coupled LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 2109–2113.

[25] L. Schmalen, V. Aref, and F. Jardel, "Non-uniformly coupled LDPC codes: Better thresholds, smaller rate-loss, and less complexity," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 376–380.

Hee-Youl Kwak received the B.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2013 and 2019, respectively. He is currently a Senior Engineer with Samsung Electronics, Co., Ltd., Gyeonggi-do, South Korea. His area of research interests include error-correcting codes, coding theory, and coding for memory.

Dae-Young Yun received the B.S. degree in electrical and computer engineering (ECE) from Seoul National University, Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree with ECE Department. His current research interests include error correcting codes and coding for memory.

Jong-Seon No (S'80–M'88–SM'10–F'12) received the B.S. and M.S.E.E. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1988. He was a Senior MTS with Hughes Network Systems from 1988 to 1990. He was an Associate Professor with the Department of Electronic Engineering, Konkuk University, Seoul, from 1990 to 1999. He joined the faculty of the Department of Electrical and Computer Engineering, Seoul National University, in 1999, where he is currently a Professor. His area of research interests include error-correcting codes, sequences, cryptography, LDPC codes, interference alignment, and wireless communication systems. He was a recipient of the IEEE Information Theory Society Chapter of the Year Award in 2007. From 1996 to 2008, he served as the Founding Chair of the Seoul Chapter of the IEEE Information Theory Society. He was the General Chair of Sequence and Their Applications 2004, Seoul. He also served as the General Co-Chair of the International Symposium on Information Theory and Its Applications 2006 and the International Symposium on Information Theory 2009, Seoul. He has been the Co-Editor-in-Chief of IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS since 2012.