Analysis and Improvement of Error-Floor Performance for JSCC Scheme Based on Double Protograph LDPC Codes

Qiwang Chen, Francis C. M. Lau, *Senior Member, IEEE*, Huihui Wu, *Member, IEEE*, Chen Chen

***Abstract*—In a joint source-channel coding (JSCC) system, excessive source compression can cause an error floor. In the original JSCC scheme based on double protograph LDPC (DP-LDPC) codes, only connections exist between check nodes (CNs) of the source protograph and variable nodes (VNs) of the channel protograph, and error floors are observed. In this paper, we investigate the joint protograph that forms the basis of a DP-LDPC code. The joint protograph consists of a source protograph, a channel protograph, and connections between the two protographs. Our main focus is on the complete-source-variable-linking (CSVL) protomatrix of the joint protograph, which determines the error floor of the DP-LDPC code. We propose a generalized source protograph extrinsic information transfer (GSP-EXIT) algorithm for evaluating the source decoding thresholds of the CSVL protomatrix. Based on the proposed algorithm, we analyze CSVL protomatrices with regular or irregular source protographs. We present design criteria for connections between VNs of the source protograph and CNs of the channel protograph. Such design rules will result in a higher source decoding threshold, which implies a lower error floor. We also propose a differential evolution algorithm for optimizing the source protograph. We present/compare analytical results and/with simulation results, and conclude that they are consistent.**

***Index Terms*—complete-source-variable-linking protomatrix, double protograph LDPC code, error floor, GSP-EXIT algorithm, joint source-channel coding.**

I. Introduction

In the coming Internet-of-Things (IoT) era, billions of sensors and devices will be connected to the Internet. Information will be passed from one device to another, and a massive amount of data traffic will be created. To ensure that data are being efficiently and effectively transmitted/received, source coding that compresses source information and channel coding that protects the data integrity should be designed to perform in an optimal manner. Using properly optimized schemes based on source-channel separation can obtain near-optimal performance when the transmission blocklength tends to infinite. But the complexity of encoding or decoding infinite-length blocks is too high and thus the transmission schemes are not practical. An alternate strategy is to use finite-length joint source-channel coding (JSCC) schemes [1], [2], [3], [4], which can provide significant improvement over the separated source-channel schemes in terms of system complexity and energy efficiency. Most JSCC schemes apply variable-length source-coding algorithms. Thus the length of the compressed source sequence may not match the requirement of the linear-block channel coding. The mismatch causes an increase in system complexity as well as error floors, i.e., the minimum bit-error-rate (BER) level even if the signal-to-noise ratio (SNR) increases [5].

A fixed-length to fixed-length JSCC scheme based on double low-density parity-check (D-LDPC) codes has been proposed in [4], where one code is used for source coding and another one for channel coding. Moreover, the check nodes (CNs) of the source LDPC code and the variable nodes (VNs) of the channel LDPC code are connected. Subsequently, the protograph concept [6], [7], [8] is incorporated into the D-LDPC codes, resulting in double protograph low-density parity-check (DP-LDPC) codes [9], [10], [11]. A joint protograph extrinsic information transfer (JP-EXIT) algorithm is also designed to analyze the channel threshold of DP-LDPC codes, i.e., error performance in the waterfall region [9]. With the use of the JP-EXIT algorithm, the channel threshold can be improved by (i) modifying the connections between the CNs of the source protograph and the VNs of the channel protograph [10], [11]; (ii) re-designing the channel protograph [9]; (iii) optimizing source protograph and channel protograph in a pairwise manner [12]; and (iv) allocating unequal power to the transmitted bits [13].

Error floors appear in D-LDPC codes when the source coding fails to match the source entropy. To lower the error floor, an information shortening algorithm for D-LDPC coding system has been designed [14]. Alternatively, source LDPC codes are designed to match source information with high entropy. In a spatially-coupled D-LDPC code, the error-floor performance gradually improves as the size of the decoding window increases [15]. Although these techniques can improve the error-floor performance, they need to sacrifice other performance metrics such as transmission efficiency.

In [16], a general D-LDPC encoding scheme is proposed. In this general coding scheme, the original source bits can join the channel coding procedures but are punctured before transmission. In other words, VNs of the source LDPC code

This work was supported by the National Natural Science Foundation of China (Grant No. 61901182), a grant from the RGC of the Hong Kong SAR, China (Project No. PolyU 152170/18E) and the Scientific Research Funds of Huaqiao University (20BS105). *(Corresponding author: Francis C. M. Lau)*

Qiwang Chen and Chen Chen are with Xiamen Key Laboratory of Mobile Multimedia Communications, College of Information Science and Engineering, Huaqiao University, Xiamen 361021 (e-mail: chenqiwang@hotmail.com, xmucecc@163.com);

Francis C. M. Lau is with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong (email: francis-cm.lau@polyu.edu.hk);

Huihui Wu was with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada (Email: huihui.wu.phd@gmail.com).

are connected to CNs of the channel LDPC code. This scheme therefore allows the transmitted (channel code) parity bits to capture more source information. It is observed that the introduction of the additional links improves the error-floor level but degrades the waterfall region performance. Several principles in designing the additional links have been proposed with an aim to improving the performance in the waterfall region [17], but then the performance in the error-floor region becomes degraded. In general, there is a lack of theoretical analyses on how the D-LDPC code design affects the error-floor level. Though a source protograph EXIT (SP-EXIT) algorithm has been proposed to evaluate a "source decoding threshold" and hence to predict the error-floor level, the algorithm assumes no connections between VNs of the source protograph and CNs of the channel protograph [18].

In this paper, we continue to use the "source decoding threshold" as an indicator of the error-floor level in a DP-LDPC code. We propose a generalized source protograph EXIT (GSP-EXIT) algorithm for evaluating the source decoding threshold of a DP-LDPC code. We also determine that only the *complete-source-variable-linking protomatrix* (i.e., part of joint protomatrix of the DP-LDPC code) is needed for the evaluation of the source decoding threshold. Using the GSP-EXIT algorithm, we investigate the error performance of different DP-LDPC code designs. We further propose an algorithm for optimizing the source decoding threshold. We verify our analytical findings with simulation results.

The novelty and contribution of this paper can be summarized as follows.

1) A GSP-EXIT algorithm is proposed for evaluating the source threshold of *complete-source-variable-linking (CSVL) protomatrices*. The theoretical source thresholds can be readily used to compare the relative error-floor levels of different DP-LDPC codes formed.
2) Given a regular or irregular source protomatrix, the effect of the source-variable-channel-check (SV-CC) linking protomatrix on the source threshold is fully evaluated. Design criteria of SV-CC linking protomatrices that give large source thresholds are also presented.
3) A differential evolution algorithm is proposed for designing optimal source protomatrices for a given SV-CC linking protomatrix. The proposed algorithm can optimize the source threshold of the overall code design.
4) Bit-error-rate performance of DP-LDPC codes are presented under different channel conditions.

This paper is organized as follows. Section II presents some other works related to JSCC systems. Section III reviews double protograph LDPC codes, including the representation of a protograph by a protomatrix, the structure of a double protograph, the corresponding DP-LDPC code construction, and the encoding and decoding processes. Section IV introduces the *complete-source-variable-linking (CSVL) protomatrix*, which consists of the source protomatrix and the source-variable-channel-check (SV-CC) linking protomatrix. This section also provides details of our proposed GSP-EXIT algorithm, which evaluates the source threshold of CSVL protomatrices. Section V investigates CSVL protomatrices with regular and irregular source protomatrices. Using the proposed GSP-EXIT algorithm, the source thresholds of the CSVL protomatrices are analyzed under different SV-CC-linking-protomatrix structures. DP-LDPC codes are constructed and simulations are performed to evaluate their bit error rates (BERs) under different conditions. In particular, the simulated error-floor levels are used to correlate with the source thresholds of the codes. This section further introduces an optimization algorithm for designing irregular source protomatrices under a given SV-CC-linking protomatrix, and presents the BER performance of the optimized codes. Section VI compares the BER performance of DP-LDPC codes under additive-white-Gaussian-noise (AWGN) and Rayleigh fading channels. Finally, Section VII concludes the paper.

II. Related Works

LDPC codes have been widely investigated for more than two decades [19], [20], [21]. They have been mainly studied as channel codes in, for example, an AWGN channel [6], [22], a non-ergodic blocking fading channel [7] and a multi-relay coded-cooperative network [8]. By replacing the single-parity-check codes in LDPC codes with Hadamard codes, LDPC-Hadamard codes and protograph-based LDPC-Hadamard codes are designed and shown to work as close as 0.16 dB compared to the ultimate Shannon limit (i.e., -1.59 dB) [23], [24]. Moreover, LDPC codes are used as source codes in Slepian-Wolf (SW) schemes [25], [26]. High rate LDPC codes are used in concatenation with Luby transform (LT) codes [27], forming the Raptor codes. Raptor codes have been applied in distributed source coding [28], video transmission over wireless networks [29], [30], and joint source-channel coding of video [31].

In [1], Huffman coding, a type of variable-length source code, is used in tandem with channel codes like recursive systematic convolutional codes and LDPC codes. It has been found that by performing joint iterative source/channel decoding, additional coding gains can be achieved. In [32], iterative joint source-channel decoding techniques that achieve near-capacity performance under H.264 multimedia communications are presented. In [2], a low-complexity JSCC scheme for transmitting JPEG-2000 images over a two-way multi-relay network has been studied. In [3], source messages are categorized into different classes and encoded by different channel codes. It is shown that as the number of classes increases, the coding scheme approaches the performance of JSCC scheme. In this paper, we focus on DP-LDPC codes.

III. Double Protograph LDPC Code

A. Joint Protograph

A protograph can be represented by a small protomatrix (or base matrix) $\mathbf{B} = \{b(i,j)\}$ where each column represents a VN, each row represents a CN, and the (i,j)-element $b(i,j)$ (a non-negative integer) indicates the number of parallel edges connecting the i-th CN and the j-th VN. A corresponding parity-check matrix $\mathbf{H}$ can be obtained by a two-step "copy-and-permute" (also called "'lifting") operation. First, the protomatrix is lifted by a factor of q_1 to remove all parallel

edges, i.e., each element $b(i,j)$ is replaced by a $q_1 \times q_1$ square matrix with (i) exactly $b(i,j)$ "1"s in each row and in each column, and (ii) "0"s elsewhere. Then, the resultant matrix is lifted again with a factor of q_2. In this step, each element "1" is replaced by a $q_2 \times q_2$ circulant permutation matrix, and each "0" is replaced by a $q_2 \times q_2$ zero matrix. In both lifting processes, the progressive-edge-growth (PEG) algorithm [33] is applied to maximize the girth (i.e., smallest cycle) of the resultant Tanner graph.

We refer to Fig. 1 where the joint protograph of a DP-LDPC code is shown. The joint protograph is represented by

$$\mathbf{B}_J = \left[\begin{array}{c|c} \mathbf{B}_s & \mathbf{B}_{L_1} \\ \mathbf{B}_{L_2} & \mathbf{B}_c \end{array} \right] \tag{1}$$

where

- $\mathbf{B}_s$ is a source protomatrix of size $m_s \times n_s$;
- $\mathbf{B}_c$ is a channel protomatrix of size $m_c \times n_c$;
- $\mathbf{B}_{L_1}$ is a *source-check-channel-variable (SC-CV) linking protomatrix* of size $m_s \times n_c$, indicating the connections between the CNs of $\mathbf{B}_s$ and the VNs of $\mathbf{B}_c$, and its structure can be represented by

$$\mathbf{B}_{L_1} = \left[\begin{array}{cc} \mathbf{I}_{m_s} & \mathbf{0} \end{array} \right], \tag{2}$$

  with $\mathbf{I}_{m_s}$ being an identity matrix of size $m_s \times m_s$, $\mathbf{0}$ being a zero matrix of appropriate size ($m_s \times m_c$ in this case), and $n_c = m_s + m_c$;
- $\mathbf{B}_{L_2}$ is a *source-variable-channel-check (SV-CC) linking protomatrix* of size $m_c \times n_s$ indicating the connections between the CNs of $\mathbf{B}_c$ and the VNs of $\mathbf{B}_s$.

We denote $M_s = q_1 q_2 m_s$, $N_s = q_1 q_2 n_s$, $M_c = q_1 q_2 m_c$ and $N_c = q_1 q_2 n_c$. After the aforementioned two-step "copy-and-permute" operation, a $q_1 q_2 (m_s + m_c) \times q_1 q_2 (n_s + n_c)$ joint parity-check matrix $\mathbf{H}_J$ is obtained [33], i.e.,

$$\mathbf{H}_J = \left[\begin{array}{c|c} \mathbf{H}_s & \mathbf{H}_{L_1} \\ \mathbf{H}_{L_2} & \mathbf{H}_c \end{array} \right] = \left[\begin{array}{c|cc} \mathbf{H}_s & \mathbf{I}_{M_s} & \mathbf{0} \\ \mathbf{H}_{L_2} & \mathbf{H}_c & \end{array} \right] \tag{3}$$

where

- $\mathbf{H}_s$ is the *source coding matrix* of size $M_s \times N_s$ and corresponds to the source P-LDPC (protograph LDPC) code;
- $\mathbf{H}_c$ is the *channel coding matrix* of size $M_c \times N_c$ and corresponds to the channel P-LDPC code;
- $\mathbf{H}_{L_1}$ is the *source-check-channel-variable (SC-CV) linking matrix* of size $M_s \times N_c$ and represents the connections between the CNs of the source coding matrix and the VNs of channel coding matrix, $\mathbf{I}_{M_s}$ is an identity matrix of size $M_s \times M_s$, and $N_c = M_s + M_c$;
- $\mathbf{H}_{L_2}$ is the *source-variable-channel-check (SV-CC) linking matrix* of size $M_c \times N_s$ and represents the connections between the VNs of the source coding matrix and the CNs of channel coding matrix.

The overall code rate of the DP-LDPC code is given by

$$R_{\text{overall}} = \frac{N_s}{M_s} \times \frac{N_c - M_c}{N_c}. \tag{4}$$

In case N_p code bits (among the N_c code bits) are punctured and not sent through the channel, the rate becomes

$$R'_{\text{overall}} = \frac{N_s}{M_s} \times \frac{N_c - M_c}{N_c - N_p}. \tag{5}$$

B. Encoding Process

Let a source sequence $\mathbf{s} \in \{0,1\}^{1 \times N_s}$ be generated from a binary independent and identically distributed (i.i.d) Bernoulli source with the probability of a "1" equal to p. The encoding procedures with a non-zero SV-CC matrix $\mathbf{H}_{L_2}$ are described as follows [11].

1) Obtain the compressed source bits $\mathbf{c}$ of size $1 \times M_s$ by computing $\mathbf{c} = \mathbf{s}(\mathbf{H}_s)^T$, where $(\cdot)^T$ indicates the matrix transposition operation.
2) Combine $\mathbf{s}$ and $\mathbf{c}$ to form a vector $[\mathbf{s}\ \mathbf{c}]$ of size $1 \times (N_s + M_s)$.
3) Based on the parity-check matrix $[\mathbf{H}_{L_2}\ \mathbf{H}_c]$ of size $M_c \times (N_s + N_c)$, construct a corresponding generator matrix $\mathbf{G}$ of the form $[\mathbf{I}_{(N_s+M_s)}\ \mathbf{G}_p]$, where $\mathbf{I}_{(N_s+M_s)}$ is an identity matrix of size $(N_s + M_s) \times (N_s + M_s)$ and $\mathbf{G}_p$ is of size $(N_s + M_s) \times M_c$.
4) Obtain the parity bits $\mathbf{p}$ of size $1 \times M_c$ by computing $\mathbf{p} = [\mathbf{s}\ \mathbf{c}]\mathbf{G}_p$.
5) The channel codeword is given by $\mathbf{d} = [\mathbf{c}\ \mathbf{p}]$ and the code bits are sent to the channel after puncturing.
6) The vector combining both the source bits and the channel code bits, denoted by $\mathbf{u}$ and given by $\mathbf{u} = [\mathbf{s}\ \mathbf{d}] = [\mathbf{s}\ \mathbf{c}\ \mathbf{p}]$, can be readily shown to satisfy $\mathbf{u}(\mathbf{H}_J)^T = \mathbf{0}$.

C. Decoding Process

We assume a binary-phase-shift keying (BPSK) modulation and an AWGN channel model. In the decoding process, the log-likelihood-ratio (LLR) values of all VNs are first initialized. The LLRs corresponding to the original source bits are initialized with the value $z_s = \ln((1-p)/p)$. For each VN corresponding to a transmitted bit, its LLR is initialized with $z_c = 2r/\sigma^2$, where r is the received signal and σ^2 is the noise variance, and the LLR for any un-transmitted bit is 0. Here the noise variance is given by

$$\sigma^2 = \frac{1}{2 \times R_{\text{overall}} \times (E_s/N_0)} \tag{6}$$

where E_s is the average transmitted energy per source information bit and N_0 is the noise power spectral density. Each of the VN-to-CN message is then set with the initial LLR value of the corresponding VN. Subsequently with reference to Fig. 1, the iterative decoding algorithm is performed as follows [11].

1) Update all the CN-to-VN messages for each of the M_c CNs in the channel code of the DP-LDPC code.
2) Update all the VN-to-CN messages for each of the N_s VNs in the source code of the DP-LDPC code.
3) Update all the CN-to-VN messages for each of the M_s CNs in the source code of the DP-LDPC code.
4) Update all the VN-to-CN messages for each of the N_c VNs in the channel code of the DP-LDPC code.
5) Estimate the source bits and channel code bits, denoted by $\hat{\mathbf{u}}$, based on the a posterior LLRs at the VNs.
6) Repeat Step 1) to Step 5) unless (i) the estimated bits $\hat{\mathbf{u}}$ satisfy $\hat{\mathbf{u}}(\mathbf{H}_J)^T = \mathbf{0}$ or (ii) the maximum number of iterations is reached.

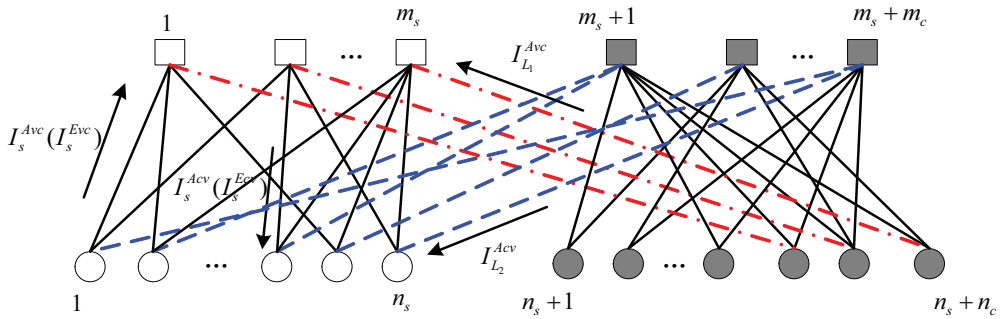

Fig. 1. The joint protograph of a DP-LDPC code, where the pecked lines (red) represent the edges in the SC-CV linking protomatrix $\mathbf{B}_{L_1}$ and the dash lines (blue) represent the edges in the SV-CC linking protomatrix $\mathbf{B}_{L_2}$.

IV. Generalized Source Protograph EXIT (GSP-EXIT) Algorithm

A. Error Floor and Source Decoding Threshold

In the source compression procedure $\mathbf{c} = \mathbf{s}(\mathbf{H}_s)^T$, it is possible that different source sequences $\mathbf{s}$ may result in the same compressed vector $\mathbf{c}$. Hence even when the transmission channel does not produce any errors, using only the source coding matrix $\mathbf{H}_s$ to try to reconstruct the original source sequence will result in errors. Hence an error-floor will appear at the high channel SNR region. For a given $\mathbf{H}_s$, a larger source statistic p implies a higher chance that two different source sequences result in the same compressed vector, and thus a higher error-floor level. To ensure a low error floor, $\mathbf{H}_s$ should match the source statistic p.

In [18], a SP-EXIT chart has been proposed to evaluate the source decoding threshold p_{th} of a source protograph (or the corresponding source protomatrix). In the SP-EXIT chart, the inner-code and outer-code EXIT curves correspond, respectively, to the VNs and CNs of the source protograph. Moreover, the shape of the inner-code curve is related to p. The gap between the inner-code curve and the outer-code curve is called the decoding tunnel. As p increases, the gap between the two curves becomes narrower. When the two curves touch each other except at the point $(1,1)$, the corresponding p is defined as the source decoding threshold p_{th}. In general, a larger p_{th} implies a lower error-floor level. The SP-EXIT algorithm in [18], however, can only analyze the cases where there is no connection between VNs of the source protograph and CNs of the channel protograph, i.e., $\mathbf{B}_{L_2} = \mathbf{0}$.

Referring to (3), besides the source coding matrix $\mathbf{H}_s$, the SV-CC matrix $\mathbf{H}_{L_2}$ connects the source bits. The design of $\mathbf{H}_{L_2}$ will hence affect the error performance and the error-floor level. Similar to $\mathbf{H}_s$, $\mathbf{H}_{L_2}$ should be designed to match the source statistic p. In order to analyze the matching criterion between p and the DP-LDPC code, we consider the source coding matrix $\mathbf{H}_s$ and the SV-CC matrix $\mathbf{H}_{L_2}$ together and define $\begin{bmatrix} \mathbf{H}_s \\ \mathbf{H}_{L_2} \end{bmatrix}$ as the *complete-source-variable-linking (CSVL) matrix*.

Instead of analyzing the CSVL matrix directly, we can investigate the corresponding protomatrix, i.e., $\begin{bmatrix} \mathbf{B}_s \\ \mathbf{B}_{L_2} \end{bmatrix}$, which is defined as the *CSVL protomatrix*. In the following, we propose a GSP-EXIT algorithm to evaluate the source threshold when $\mathbf{B}_{L_2}$ (and thus $\mathbf{H}_{L_2}$) is not a zero matrix.

B. GSP-EXIT Algorithm

We propose a GSP-EXIT algorithm which considers the source protomatrix $\mathbf{B}_s$ and the SV-CC linking protomatrix $\mathbf{B}_{L_2}$ together, i.e., the CSVL protomatrix. Referring to Fig. 1, we define six types of mutual information (MI) flowing into/within the source protograph:

- $I_s^{Evc}(i,j)$: extrinsic MI from j-th VN to i-th CN in $\mathbf{B}_s$;
- $I_s^{Ecv}(i,j)$: extrinsic MI from i-th CN to j-th VN in $\mathbf{B}_s$;
- $I_s^{Avc}(i,j)$: a-prior MI from j-th VN to i-th CN in $\mathbf{B}_s$;
- $I_s^{Acv}(i,j)$: a-prior MI from i-th CN to j-th VN in $\mathbf{B}_s$;
- $I_{L_1}^{Avc}(i,l)$: a-prior MI from l-th VN to i-th CN in $\mathbf{B}_{L_1}$;
- $I_{L_2}^{Acv}(k,j)$: a-prior MI from k-th CN to j-th VN in $\mathbf{B}_{L_2}$.

In [4], a function $J(\cdot)$ is defined as

$$J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(\xi-\sigma^2/2)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \cdot \log_2(1+e^{-\xi})d\xi$$

and $J^{-1}(\cdot)$ represents its inverse function. Another function $J_{BSC}(\cdot)$ is defined as

$$J_{BSC}(\mu,p) = (1-p) \times I(\mathbf{s};\chi^{(1-p)}) + p \times I(\mathbf{s};\chi^{(p)}),$$

where $I(\mathbf{s};\chi)$ is the MI between the source bits $\mathbf{s}$ and χ; $\chi^{(1-p)} \sim N(\mu + z_s, 2\mu)$ and $\chi^{(p)} \sim N(\mu - z_s, 2\mu)$ with $z_s = \ln((1-p)/p)$.

1) Inner-code curve: We consider $\mathbf{B}_s$ and $\mathbf{B}_{L_2}$ together, i.e., we use $\begin{bmatrix} \mathbf{B}_s \\ \mathbf{B}_{L_2} \end{bmatrix}$. Given (i) $I_s^{Acv}(i,j) \in [0,1]$ for all i and j; (ii) $I_{L_2}^{Acv}(k,j) \in [0,1]$ for all k and j; and (iii) p; $I_s^{Evc}(i,j)$ is calculated as follows. For $i = 1,2,..,m_s$ and $j = 1,2,...,n_s$, $I_s^{Evc}(i,j) = 0$ if $b_s(i,j) = 0$. Otherwise $I_s^{Evc}(i,j)$ is computed using

$$I_s^{Evc}(i,j) = J_{BSC}\Big(\Gamma_s^{vc}(\mathbf{B}_s,i,j) + \Gamma_{L_2}(\mathbf{B}_{L_2},j), p\Big), \quad (7)$$

where

$$\Gamma_s^{vc}(\mathbf{B}_s,i,j) = \sum_{t\neq i} b_s(t,j) \times [J^{-1}(I_s^{Acv}(t,j))]^2 + (b_s(i,j)-1) \times [J^{-1}(I_s^{Acv}(i,j))]^2 \quad (8)$$

$$\Gamma_{L_2}(\mathbf{B}_{L_2},j) = \sum_{k=1}^{m_c} b_{L_2}(k,j) \times [J^{-1}(I_{L_2}^{Acv}(k,j))]^2. \quad (9)$$

To obtain an inner-code curve, we assume (i) the same $I_s^{Acv}(i,j)$ for all i and j, i.e., $I_s^{Acv}(i,j) = I_s^{Acv}$; and (ii) the same $I_{L_2}^{Acv}(k,j)$ for all k and j, i.e., $I_{L_2}^{Acv}(k,j) = I_{L_2}^{Acv}$. Then, (8) and (9) are simplified to, respectively,

$$\begin{aligned}\Gamma_s^{vc}(\mathbf{B}_s,i,j) =&[J^{-1}(I_s^{Acv})]^2\left[\sum_{t\neq i} b_s(t,j) + (b_s(i,j)-1)\right]\\ =&[J^{-1}(I_s^{Acv})]^2(w_{s,*,j}-1)\end{aligned} \tag{10}$$

and

$$\begin{aligned}\Gamma_{L_2}(\mathbf{B}_{L_2},j) =&[J^{-1}(I_{L_2}^{Acv})]^2\sum_{k=1}^{m_c} b_{L_2}(k,j)\\ =&[J^{-1}(I_{L_2}^{Acv})]^2 w_{L_2,*,j}\end{aligned} \tag{11}$$

with $w_{s,*,j}$ and $w_{L_2,*,j}$ being the weight of the j-th column in $\mathbf{B}_s$ and $\mathbf{B}_{L_2}$, respectively. Then the average I_s^{Evc} is computed by

$$I_s^{Evc} = \sum_i\sum_j I_s^{Evc}(i,j)\times b_s(i,j) / \sum_i\sum_j b_s(i,j) \tag{12}$$

for every set of $\{I_s^{Acv}, I_{L_2}^{Acv}, p\}$.

2) Outer-code curve: We consider $\mathbf{B}_s$ and $\mathbf{B}_{L_1}$ together, i.e., we use $[\mathbf{B}_s\ \mathbf{B}_{L_1}]$. Given (i) $I_s^{Avc}(i,j) \in [0,1]$ for all i and j; and (ii) $I_{L_1}^{Avc}(i,l) \in [0,1]$ for all i and l; the average I_s^{Ecv} is calculated as follows. For $i = 1,2,...,m_s$ and $j = 1,2,...,n_s$, $I_s^{Ecv}(i,j) = 0$ if $b_s(i,j) = 0$. Otherwise, $I_s^{Ecv}(i,j)$ is computed using

$$I_s^{Ecv}(i,j) = 1.0 - J\Big(\Gamma_s^{cv}(\mathbf{B}_s,i,j) + \Gamma_{L_1}(\mathbf{B}_{L_1},i)\Big) \tag{13}$$

where

$$\begin{aligned}\Gamma_s^{cv}(\mathbf{B}_s,i,j) =&\sum_{t\neq j} b_s(i,t)\times[J^{-1}(1-I_s^{Avc}(i,t))]^2\\ &+(b_s(i,j)-1)\times[J^{-1}(1-I_s^{Avc}(i,j))]^2\end{aligned} \tag{14}$$

$$\Gamma_{L_1}(\mathbf{B}_{L_1},i) = \sum_{l=1}^{n_c} b_{L_1}(i,l)\times[J^{-1}(1-I_{L_1}^{Avc}(i,l))]^2. \tag{15}$$

To obtain an outer-code curve, we assume the same $I_s^{Avc}(i,j)$ for all i and j, i.e., $I_s^{Avc}(i,j) = I_s^{Avc}$. We further assume that perfect LLR messages are passed from the *l*-th VN to *i*-th CN in $\mathbf{B}_{L_1}$, i.e., $I_{L_1}^{Avc}(i,l) = 1$ for all i and l. Then, (14) and (15) are simplified to, respectively,

$$\begin{aligned}\Gamma_s^{cv}(\mathbf{B}_s,i,j) =&[J^{-1}(1-I_s^{Avc})]^2\left[\sum_{t\neq j} b_s(i,t) + (b_s(i,j)-1)\right]\\ =&[J^{-1}(1-I_s^{Avc})]^2(w_{s,i,*}-1)\end{aligned} \tag{16}$$

and

$$\Gamma_{L_1}(\mathbf{B}_{L_1},i) = 0\ \forall i \text{ when } I_{L_1}^{Avc}(i,l) = 1\ \forall i,l \tag{17}$$

with $w_{s,i,*}$ being the weight of the i-th row in $\mathbf{B}_s$. The average I_s^{Ecv} is computed by

$$I_s^{Ecv} = \sum_i\sum_j I_s^{Ecv}(i,j)\times b_s(i,j) / \sum_i\sum_j b_s(i,j). \tag{18}$$

for a given value of $I_s^{Avc} \in [0,1]$.

3) GSP-EXIT curves: Using the above steps, we can obtain (i) I_s^{Evc} for every set of $\{I_s^{Acv}, I_{L_2}^{Acv}, p\}$, and (ii) I_s^{Ecv} for every $I_s^{Avc} \in [0,1]$. Using the fact that $I_s^{Evc} = I_s^{Avc}$ and $I_s^{Ecv} = I_s^{Acv}$, we can plot the GSP-EXIT curves on the same figure, i.e.,

- plot I_s^{Evc} versus I_s^{Acv} when I_s^{Acv} increases from 0 to 1 while fixing $I_{L_2}^{Acv}$ and p;
- plot I_s^{Avc} versus I_s^{Ecv} curve when I_s^{Avc} increases from 0 to 1.

C. Source Decoding Threshold Estimation

As mentioned in Sect. IV-A, the SP-EXIT chart is used to evaluate the source decoding threshold p_{th} only when $\mathbf{B}_{L_2} = \mathbf{0}$, i.e., $I_{L_2}^{Acv} = 0$. In our proposed GSP-EXIT algorithm, the scenario where $\mathbf{B}_{L_2} \neq \mathbf{0}$ and hence $I_{L_2}^{Acv} \neq 0$ is also analyzed. We consider the following CSVL protomatrix

$$\begin{bmatrix}\mathbf{B}_s^{reg}\\ \hline \mathbf{B}_{L_2}\end{bmatrix} = \left[\begin{array}{cccccccc} 1&0&1&1&1&0&1&1\\ 1&1&0&1&1&1&0&1\\ 1&1&1&0&1&1&1&0\\ 0&1&1&1&0&1&1&1\\ \hline b_{L_2}^{1,1}&b_{L_2}^{1,2}&b_{L_2}^{1,3}&b_{L_2}^{1,4}&b_{L_2}^{1,5}&b_{L_2}^{1,6}&b_{L_2}^{1,7}&b_{L_2}^{1,8}\\ b_{L_2}^{2,1}&b_{L_2}^{2,2}&b_{L_2}^{2,3}&b_{L_2}^{2,4}&b_{L_2}^{2,5}&b_{L_2}^{2,6}&b_{L_2}^{2,7}&b_{L_2}^{2,8}\\ b_{L_2}^{3,1}&b_{L_2}^{3,2}&b_{L_2}^{3,3}&b_{L_2}^{3,4}&b_{L_2}^{3,5}&b_{L_2}^{3,6}&b_{L_2}^{3,7}&b_{L_2}^{3,8}\\ b_{L_2}^{4,1}&b_{L_2}^{4,2}&b_{L_2}^{4,3}&b_{L_2}^{4,4}&b_{L_2}^{4,5}&b_{L_2}^{4,6}&b_{L_2}^{4,7}&b_{L_2}^{4,8}\end{array}\right], \tag{19}$$

where

- $\mathbf{B}_s^{reg}$ is a 4×8 regular protomatrix with column weight being 3 and row weight being 6;
- $\mathbf{B}_{L_2}$ is of size 4×8, $b_{L_2}^{k,j} = b_{L_2}(k,j)$ represents the (k,j)-th element of $\mathbf{B}_{L_2}$, and unless otherwise stated $b_{L_2}^{k,j} = 0$ for all k and j.

We consider the case when $b_{L_2}^{2,6} = b_{L_2}^{3,7} = b_{L_2}^{4,8} = 2$ and $p = 0.12$. We plot the GSP-EXIT chart in Fig. 2 under the condition that $I_{L_1}^{Avc} = 1$, i.e., perfect LLR messages are passed from the VNs of the channel protograph to the CNs of the source protograph via $\mathbf{B}_{L_1}$. Referring to the figure, the decoding tunnel between the inner-code curves (indicated with blue lines) and the outer-code curve (indicated with a dashed line) signifies the resilience of the source information and correlates with the level of error-floor — the wider the tunnel, the lower the error floor. The decoding tunnel is closed when $I_{L_2}^{Acv} = 0.0$ (representing the case $\mathbf{B}_{L_2} = \mathbf{0}$) and is open when $I_{L_2}^{Acv} \geq 0.4$. The results indicate that a non-zero $\mathbf{B}_{L_2}$ can improve the error-floor performance of the JSCC scheme. Moreover, the tunnel becomes wider and wider as $I_{L_2}^{Acv}$ is gradually increased to 1.

We further consider the scenario when perfect LLR messages are also passed from the CNs of the channel protograph to the VNs of the source protograph via $\mathbf{B}_{L_2}$, i.e., $I_{L_2}^{Acv} = 1$, and (11) becomes

$$\Gamma_{L_2}(\mathbf{B}_{L_2},j) = w_{L_2,*,j}. \tag{20}$$

Combining with the previous assumption that perfect LLR messages are passed from the VNs of the channel protograph to the CNs of the source protograph via $\mathbf{B}_{L_1}$, we therefore assume all LLR messages passed from the channel decoder to the source decoder are perfect, i.e, $I_{L_1}^{Avc} = 1$ and $I_{L_2}^{Acv} = 1$.

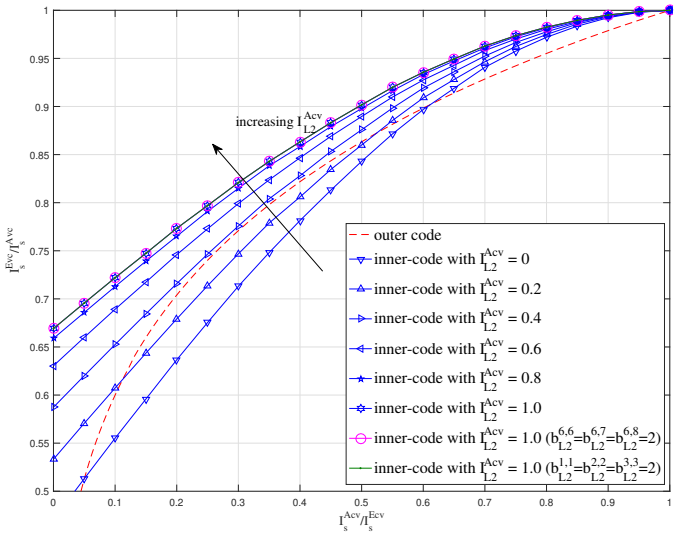

Fig. 2. The GSP-EXIT chart for the CSVL protomatrix (19). $\mathbf{B}_{L_2}$ contains three non-zero columns with only one "2" in each non-zero column: blue lines denote the inner-code curves with $b_{L_2}^{4,6} = b_{L_2}^{5,7} = b_{L_2}^{6,8} = 2$ at different $I_{L_2}^{Acv}$ values; pink line and green line denote the inner-code curves with $b_{L_2}^{6,6} = b_{L_2}^{6,7} = b_{L_2}^{6,8} = 2$ and $b_{L_2}^{1,2} = b_{L_2}^{2,2} = b_{L_2}^{3,3} = 2$, respectively, at $I_{L_2}^{Acv} = 1$. The red (dashed) line denotes the outer-code curve. $I_{L_1}^{Avc} = 1$ and $p = 0.12$.

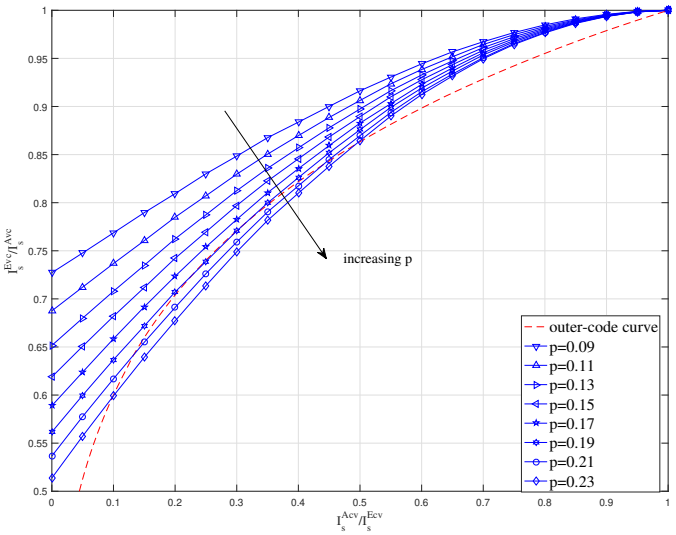

Fig. 3. The GSP-EXIT chart for the CSVL protomatrix (19) at different source statistic p. $\mathbf{B}_{L_2}$ contains three non-zero columns with only one "2" in each non-zero column: $b_{L_2}^{4,6} = b_{L_2}^{5,7} = b_{L_2}^{6,8} = 2$. The red (dashed) line denotes the outer-code curve. $I_{L_1}^{Avc} = 1$ and $I_{L_2}^{Acv} = 1.0$ and $p = 0.12$.

Fig. 3 shows the GSP-EXIT chart for different source statistic p. We can observe that as p increases, the decoding tunnel between the inner-code curve and the outer-code curve becomes narrower. The value of p when the tunnel is nearly closed is the source decoding threshold p_{th}. As shown in Fig. 3, the inner-code curve and outer-code curve almost touch each other when $p = 0.19$, and thus $p_{th} = 0.19$.

Given $\mathbf{B}_s$ and $\mathbf{B}_{L_2}$, the pseudo-code used for finding the source decoding threshold p_{th} is shown in **Algorithm 1**. The parameters used are as follows: maximum number of iterations $t_{max} = 200$, step size $p' = 10^{-3}$ and tolerance $\delta = 10^{-7}$.

V. CSVL Protomatrices with Regular and Irregular Source Protographs

In this section, we continue to study CSVL protomatrices under the condition that all LLR messages passing from the channel decoder to the source decoder are perfect, i.e, $I_{L_1}^{Avc} = 1$ and $I_{L_2}^{Acv} = 1$. To be more general, we analyze the CSVL protomatrix assuming both regular and irregular source protomatrices. We apply (7), (10), and (20) to (12) for evaluating the inner-code curve I_s^{Evc}; and apply (13), (16), and (17) to (18) for evaluating the outer-code curve I_s^{Ecv}.

A. Regular Source Protomatrix

We consider a regular source protomatrix, i.e., all column weights are the same and all row weights are the same. We also denote the column weight by $w_{s,c}$ and row weight by $w_{s,r}$; and hence $w_{s,*,j} = w_{s,c}\ \forall j$ and $w_{s,i,*} = w_{s,r}\ \forall i$. The $\mathbf{B}_s^{reg}$ in (19) is one example, which has a size of 4×8 with all column weights (degree of VNs) being $w_{s,c} = 3$ and all row weights (degree of CNs) being $w_{s,r} = 6$.

For a regular $\mathbf{B}_s$, (16) is simplified to

$$\Gamma_s^{cv}(\mathbf{B}_s, i, j) = [J^{-1}(1 - I_s^{Avc})]^2(w_{s,r} - 1). \tag{21}$$

Substituting (13), (17), and (21) to (18), it can be shown that the outer-code curve I_s^{Ecv} becomes a function depending only on I_s^{Avc} and $w_{s,r}$, i.e.,

$$I_s^{Ecv} = 1.0 - J\Big([J^{-1}(1 - I_s^{Avc})]^2(w_{s,r} - 1)\Big). \tag{22}$$

Considering the inner-code curve, (10) is simplified to

$$\Gamma_s^{vc}(\mathbf{B}_s, i, j) = [J^{-1}(I_s^{Avc})]^2(w_{s,c} - 1)\ \forall i, j\ . \tag{23}$$

Moreover, (20) depends only on the column weight of $\mathbf{B}_{L_2}$, but not on the distribution of the non-zero entries in the same column. Substituting (7), (20) and (23) to (12), the inner-code curve I_s^{Evc} becomes

$$I_s^{Evc} = \sum_j J_{BSC}\Big([J^{-1}(I_s^{Avc})]^2(w_{s,c} - 1) + w_{L_2,*,j}, p\Big)/n_s, \tag{24}$$

which is a function of I_s^{Acv}, $w_{s,c}$, the column-weight distribution of $\mathbf{B}_{L_2}$ ($\{w_{L_2,*,j}\}$), p and n_s (number of columns in $\mathbf{B}_s$).

1) $\mathbf{B}_{L_2}$ with the same column-weight distribution: (24) indicates that $\mathbf{B}_{L_2}$ with the same column-weight distribution produces the same inner-code curve. To verify this, we consider the following three different $\mathbf{B}_{L_2}$ that have the same column-weight distribution.

- $b_{L_2}^{4,6} = b_{L_2}^{5,7} = b_{L_2}^{6,8} = 2$, and 0 elsewhere
- $b_{L_2}^{6,6} = b_{L_2}^{6,7} = b_{L_2}^{6,8} = 2$, and 0 elsewhere
- $b_{L_2}^{1,1} = b_{L_2}^{2,2} = b_{L_2}^{3,3} = 2$, and 0 elsewhere

In these three cases, three columns have a weight of 2 and all other columns have a weight of 0. Fig. 2 plots the inner-code curves for the above three cases based on the $\mathbf{B}_s^{reg}$ in (19) and $p = 0.12$. It can be observed that all the three curves overlap completely, verifying our theoretical findings.

Algorithm 1 GSP-EXIT algorithm for evaluating the source decoding threshold p_{th}

Given $\mathbf{B}_s$ and $\mathbf{B}_{L_2}$, evaluate the source decoding threshold p_{th}.
Initialization: Set the maximum number of iterations t_{max}, step size p' and tolerance δ. Set $I_{L_1}^{Avc}(i,l) = I_{L_1}^{Avc} = 1\ \forall i,l$ and $I_{L_2}^{Acv}(k,j) = I_{L_2}^{Acv} = 1\ \forall k,j$.
for a given p (assume $p > p_{th}$ and hence too large) **do**
    Set $t = 1$; $I_s^{Avc}(i,j) = I_s^{Avc} = 0\ \forall i,j$; $I_s^{Acv}(i,j) = I_s^{Acv} = 0\ \forall i,j$; $I_s^{Evc} = 0$; $I_s^{Ecv} = 0$.
    while $\left((1.0 - I_s^{Evc}) > \delta \text{ or } (1.0 - I_s^{Ecv}) > \delta\right)$ and $t \le t_{max}$ **do**
        for $i = 1, ..., m_s$, $j = 1, ..., n_s$ **do**
            Compute $I_s^{Evc}(i,j)$ using (7).
            Set $I_s^{Avc}(i,j) = I_s^{Evc}(i,j)$.
            Compute I_s^{Evc} using (12).
        end for
        for $i = 1, ..., m_s$, $j = 1, ..., n_s$ **do**
            Compute $I_s^{Ecv}(i,j)$ using (13).
            Set $I_s^{Acv}(i,j) = I_s^{Ecv}(i,j)$.
            Compute I_s^{Ecv} using (18).
        end for
        $t = t + 1$.
    end while
    if $\left((1.0 - I_s^{Evc}) \le \delta \text{ and } (1.0 - I_s^{Ecv}) \le \delta\right)$ **then**
        $p_{th} = p$, goto **final** [Comment: Largest p to allow the iterations to converge almost to $(1,1)$ is found.]
    else
        $p = p - p'$. [Comment: p too large and needs to be decreased].
    end if
end for
final

2) Number of non-zero columns in $\mathbf{B}_{L_2}$: Supposing the weight of a column in $\mathbf{B}_{L_2}$ is either 0 or a constant value, say 2, (24) shows that more columns with the constant weight will produce a higher I_s^{Evc}. As a result, the decoding tunnel between the corresponding inner-code curve and the outer-code curve becomes wider, resulting in a larger source decoding threshold p_{th}. Fig. 4 plots the GSP-EXIT chart for the CSVL protomatrix (19) at $p = 0.12$. $\mathbf{B}_{L_2}$ contains 1 to 5 non-zero column(s) with only one entry "2" in each non-zero column. Table I lists the corresponding threshold versus the number of non-zero columns. The results verify our theory that more columns with the constant weight of 2 produces a wider decoding tunnel, and hence a larger source decoding threshold.

In summary, for a regular source protomatrix, the source decoding threshold p_{th} is determined by the distribution of column weights in $\mathbf{B}_{L_2}$. In general, more non-zero-weight columns in $\mathbf{B}_{L_2}$ gives a higher p_{th} and hence a lower error floor.

3) Bit-error-rate (BER) simulations: Next we perform bit-error-rate (BER) simulations. The CSVL protomatrix in (19)

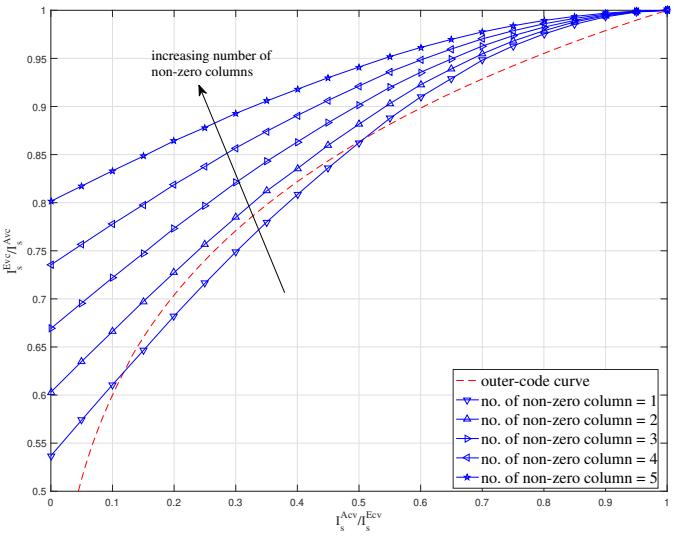

Fig. 4. The GSP-EXIT chart for the CSVL protomatrix (19) at $p = 0.12$. $\mathbf{B}_{L_2}$ contains 1 to 5 non-zero column(s) with only one entry "2" in each non-zero column. The red (dashed) line denotes the outer-code curve. $I_{L_1}^{Avc} = 1$ and $I_{L_2}^{Acv} = 1$.

TABLE I
SOURCE DECODING THRESHOLD p_{th} FOR THE CSVL PROTOMATRIX (19). $\mathbf{B}_{L_2}$ CONTAINS 1 TO 5 NON-ZERO COLUMN(S) WITH ONLY ONE ENTRY "2" IN EACH NON-ZERO COLUMN.

Number of non-zero columns in $\mathbf{B}_{L_2}$	p_{th}
1	0.1029
2	0.1345
3	0.1875
4	0.2790
5	0.4973

is used jointly with

$$\begin{bmatrix} \mathbf{B}_{L_1} \\ \hline \mathbf{B}_c^{reg} \end{bmatrix} = \left[\begin{array}{cccccccc} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ \hline 1&1&1&0&1&1&1&0 \\ 1&1&0&1&1&1&0&1 \\ 1&0&1&1&1&0&1&1 \\ 0&1&1&1&0&1&1&1 \end{array}\right], \tag{25}$$

in which the channel protomatrix $\mathbf{B}_c^{reg}$ is of size 4×8; and is regular with column weight 3 and row weight 8. For $\mathbf{B}_{L_2}$, each non-zero column has a weight of 2 and contains two "1"s. The positions of "1" are selected with an aim to (i) connecting more VNs in $\mathbf{B}_s$ and CNs in $\mathbf{B}_c^{reg}$; and (ii) minimizing the channel threshold (SNR_{th}) (e.g., evaluated by using the JP-

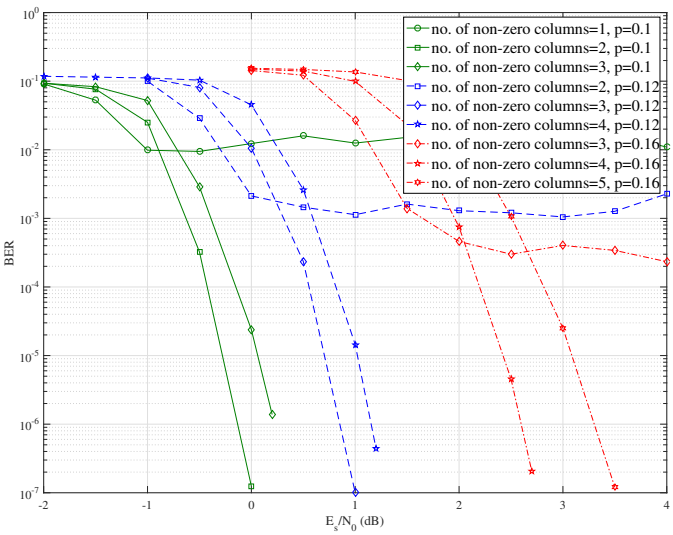

Fig. 5. BER performance of the DP-LDPC code derived from (19) and (25) at source statistic $p = 0.1, 0.12, 0.16$. The number of non-zero columns in $\mathbf{B}_{L_2}$ varies from 1 to 3 for $p = 0.1$; 2 to 4 for $p = 0.12$; and 3 to 5 for $p = 0.16$. Each non-zero column in $\mathbf{B}_{L_2}$ has a weight of 2.

EXIT algorithm in [9]). For example, $\mathbf{B}_{L_2}$ is given by

$$\mathbf{B}_{L_2} = \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0 \\ 1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0 \end{bmatrix}$$

when the first two columns are selected to be the non-zero columns. Lifting factors of $q_1 = 4$ and $q_2 = 100$ are used to form the DP-LDPC code based on (19) and (25). Thus, $N_s = N_c = 3200$, and $M_s = M_c = 1600$. We assume an AWGN channel and BPSK modulation. In the decoding process, a maximum of 100 iterations is used and a maximum of 2×10^6 blocks is simulated for every E_s/N_0.

Fig. 5 plots the BER results. At source statistic $p = 0.1$, an error floor occurs when there is only one non-zero column in $\mathbf{B}_{L_2}$; the error floor disappears when there are two or three non-zero columns. Similarly, at $p = 0.12$ ($p = 0.16$), an error floor occurs when there are two (three) non-zero columns in $\mathbf{B}_{L_2}$; the error floor disappears when there are three (four) or four (five) non-zero columns. The results are consistent with our finding in Table I that more non-zero columns in $\mathbf{B}_{L_2}$ provides a higher p_{th} and hence a smaller chance of having an error floor. Fig. 5 also shows that more non-zero columns in $\mathbf{B}_{L_2}$ gives a worse BER in the waterfall region. Thus, depending on p and the actual operating region, a smaller number of non-zero columns in $\mathbf{B}_{L_2}$ may be selected. For example, at $p = 0.16$ and $E_s/N_0 \in$ [1 dB, 2 dB], we may decide to use a $\mathbf{B}_{L_2}$ with three (instead of four or five) non-zero columns.

B. Irregular Source Protomatrix

We consider an irregular source protomatrix, i.e., the rows and columns of $\mathbf{B}_s$ can have different weights. We use the rate-1/2 source protograph in [18] as an example and denote the corresponding protomatrix as $\mathbf{B}_s^{[18]}$. $\mathbf{B}_s^{[18]}$ has been optimized in [18] under $\mathbf{B}_{L_2} = \mathbf{0}$ and has a source threshold of $p_{th} = 0.0977$ (value also listed in Table IV under $\mathbf{B}_s^{[18]}$ and $\psi_0 = \{\varnothing\}$). With $\mathbf{B}_s^{[18]}$, the CSVL protomatrix can be written as

$$\begin{bmatrix} \mathbf{B}_s^{[18]} \\ \hline \mathbf{B}_{L_2} \end{bmatrix} = \begin{bmatrix} 3\,2\,1\,1\,0\,1\,0\,0 \\ 2\,3\,1\,0\,1\,0\,1\,0 \\ 3\,3\,0\,0\,0\,0\,0\,1 \\ 3\,0\,1\,2\,2\,1\,1\,1 \\ \hline \mathbf{B}_{L_2} \end{bmatrix}, \tag{26}$$

where $\mathbf{B}_{L_2}$ is assumed to be the same size as that in (19), i.e., 4×8.

1) Selection of columns with non-zero weights: In the following, we define ψ as the set of columns in $\mathbf{B}_{L_2}$ with non-zero weights. For example, $\psi = \{1, 3, 5\}$ indicates that the weights of the 1st, 3rd and 5th columns in $\mathbf{B}_{L_2}$ are non-zero while the weights of other columns are zero. As in the previous section, we assume that there is only one entry with a value of "2" in each non-zero-weight column while other entries in the same column are "0".

We evaluate the source threshold of the CSVL protomatrix in (26) when the number of columns with non-zero weights equals $1, 2, 3$ and 4. Denoting $C_n^k = \frac{n!(n-k)!}{k!}$, the number of the non-zero columns in $\mathbf{B}_{L_2}$ being 1, 2, 3 and 4 have, respectively, a total of $C_8^1 = 8$, $C_8^2 = 28$, $C_8^3 = 56$ and $C_8^4 = 70$ possible combinations. For each of these combinations, we evaluate the source threshold using the GSP-EXIT algorithm, i.e., **Algorithm 1**. Table II lists the 4 combinations with the best p_{th} and another 4 combinations with the worst values for each case.

Referring to the table, if only one column in $\mathbf{B}_{L_2}$ is allowed to have non-zero weights, the 3rd column should be selected because it can achieve the highest source decoding threshold, i.e., ψ={3} and $p_{th} = 0.1156$. Note that the corresponding 3rd column in $\mathbf{B}_s^{[18]}$ has neither the largest or smallest column weight. Table II also shows that the best p_{th} increases as the number of non-zero columns in $\mathbf{B}_{L_2}$ increases from 1 to 4. The result is consistent with that in the previous section where a regular source protomatrix has been considered. However, if the non-zero columns in $\mathbf{B}_{L_2}$ are randomly selected, p_{th} may decrease with the number of non-zero columns, e.g., compare p_{th} values for the following sets — $\{3\}$*v.s.*$\{1, 8\}$, $\{2, 4\}$*v.s.*$\{1, 2, 8\}$, $\{1, 3, 5\}$*v.s.*$\{5, 6, 7, 8\}$. Another observation from Table II is that the best non-zero-column combination in each case does not contain the column numbers $6, 7$ or 8 in $\mathbf{B}_s$, which correspond to weight-2 columns in the irregular source protomatrix $\mathbf{B}_s^{[18]}$. Comparing the results in Table I and Table II also indicates that under the same number of non-zero columns in $\mathbf{B}_{L_2}$, the CSVL protomatrix with the irregular $\mathbf{B}_s^{[18]}$ can achieve a higher p_{th} than that with the regular $\mathbf{B}_s^{reg}$.

In summary, for a given source protomatrix $\mathbf{B}_s$ with irregular weight distribution, (i) the number of non-zero-weight columns in $\mathbf{B}_{L_2}$ and (ii) the columns in $\mathbf{B}_{L_2}$ that should be given non-zero weights, need to be selected with care in order to maximize the source decoding threshold. In particular, to obtain the best source decoding threshold, non-zero-weight

TABLE II

SET OF COLUMNS ψ IN $\mathbf{B}_{L_2}$ WITH NON-ZERO WEIGHTS VERSUS SOURCE DECODING THRESHOLD p_{th}. THE NUMBER OF NON-ZERO-WEIGHT COLUMNS IN $\mathbf{B}_{L_2}$ EQUALS 1, 2, 3, AND 4. ONLY THE 4 SETS SHOWING THE BEST (HIGHEST) p_{th} AND THE 4 SETS SHOWING THE WORST p_{th} ARE LISTED FOR EACH CASE. THE CSVL PROTOMATRIX IN (26) IS USED.

ψ with cardinality 1	p_{th}	ψ with cardinality 2	p_{th}	ψ with cardinality 3	p_{th}	ψ with cardinality 4	p_{th}
$\{\mathbf{3}\}$	**0.1156**	$\{\mathbf{2,4}\}$	**0.1498**	$\{\mathbf{1,3,5}\}$	**0.2076**	$\{\mathbf{1,2,4,5}\}$	**0.3632**
$\{4\}$	0.1117	$\{2,5\}$	0.1494	$\{1,5,7\}$	0.2047	$\{1,2,3,4\}$	0.3352
$\{5\}$	0.1117	$\{1,5\}$	0.1493	$\{1,3,4\}$	0.2037	$\{1,2,3,5\}$	0.3352
$\{6\}$	0.1101	$\{1,4\}$	0.1482	$\{2,3,4\}$	0.2029	$\{1,2,4,6\}$	0.3352
$\{7\}$	0.1101	$\{6,7\}$	0.1201	$\{6,7,8\}$	0.1393	$\{4,5,6,7\}$	0.1650
$\{8\}$	0.1058	$\{1,2\}$	0.1058	$\{4,6,7\}$	0.1388	$\{1,2,7,8\}$	0.1650
$\{2\}$	0.1035	$\{1,8\}$	0.1058	$\{5,6,7\}$	0.1386	$\{3,6,7,8\}$	0.1626
$\{1\}$	0.1031	$\{1,7\}$	0.1057	$\{1,2,8\}$	0.1058	$\{5,6,7,8\}$	0.1584

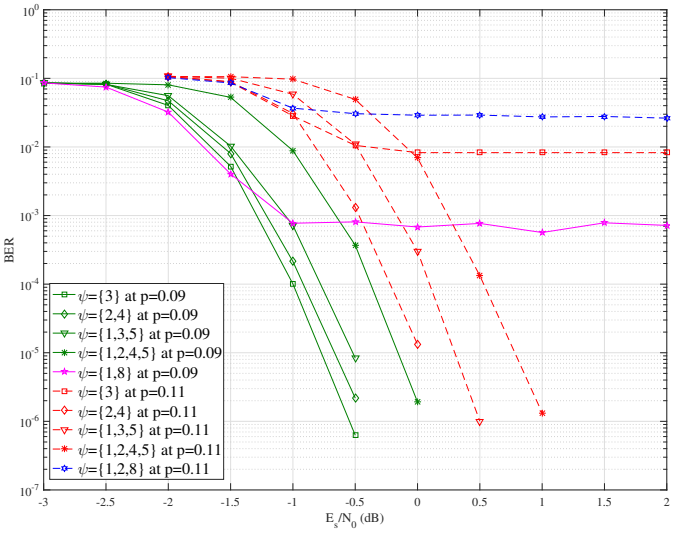

Fig. 6. BER performance of the DP-LDPC code derived from (26) and (25) at source statistic $p = 0.09$ and 0.11. The number of non-zero columns in $\mathbf{B}_{L_2}$ varies from 1 to 4. ψ indicates the set of non-zero columns and each non-zero column has a weight of 2.

columns in $\mathbf{B}_{L_2}$ are not recommended to align with weight-2 columns in the irregular source protomatrix.

2) BER simulations: We simulate the BER performance of the DP-LDPC code formed by (26) and (25). The same simulation setting (e.g., column weight, lifting factors) as in Sect. V-A3 is used. Fig. 6 plots the BER results. We consider the cases when the optimal column sets under different number of non-zero columns in $\mathbf{B}_{L_2}$ are applied, i.e., we consider $\psi = \{3\}$, $\psi = \{2,4\}$, $\psi = \{1,3,5\}$ and $\psi = \{1,2,4,5\}$. At source statistic $p = 0.09$, no error floor occurs and the BER performance in the waterfall region degrades as the number of non-zero columns increases. At source statistic $p = 0.11$, similar observations can be obtained except that $\psi = \{3\}$ gives an error floor. We also consider two column sets in Table II that are among the worst ones — $\psi = \{1,8\}$ and $\psi = \{1,2,8\}$. At $p = 0.09$, $\psi = \{1,8\}$ shows an error floor, and is outperformed by $\psi = \{3\}$ when E_s/N_0 is beyond -1.5 dB. At $p = 0.11$, $\psi = \{1,2,8\}$ shows an error floor, and is always outperformed by $\psi = \{3\}$ and $\psi = \{2,4\}$ over all E_s/N_0. The BER results are consistent with our finding in Table II — when the optimal non-zero-column combinations are selected, more non-zero columns in $\mathbf{B}_{L_2}$ provides a higher p_{th} and hence a smaller chance of having an error floor. It also verifies that $\mathbf{B}_{L_2}$ with a higher number of non-zero columns can give a much worse BER performance if the non-zero-column combination is randomly selected. Again, like what we have observed in Sect. V-A3, the selection of ψ here will depend on p and the operating region. For example, at $p = 0.09$ and $p = 0.11$, respectively, $\psi = \{3\}$ and $\psi = \{2,4\}$ should be selected.

Next, we compare the BER performance of the DP-LDPC codes with the same number of non-zero columns in $\mathbf{B}_{L_2}$ but under different non-zero column sets. We consider $\mathbf{B}_{L_2}$ with the non-zero column sets shown in Table III, which also lists the corresponding p_{th} of the CSVL protomatrix based on (26). Figs. 7, 8, and 9 plot, respectively, the BER curves at $p = 0.10$, $p = 0.12$ and $p = 0.14$, under $\mathbf{B}_{L_2}$ with one, two and three non-zero columns. The results again verify our previous analytical finding that for $\mathbf{B}_{L_2}$ with the same number of non-zero columns, different non-zero column sets can produce very different BER results and error-floor levels. For example, in Fig. 7 case-1 achieves an error-floor level of 2×10^{-4} while case-4 achieves an error-floor level of around 2×10^{-2}; in Fig. 9 case-9 does not shown any error floor while case-12 achieves an error-floor level of 10^{-2}. The BER results also confirms that a higher p_{th} produces a lower error floor.

C. Optimization of Irregular Source Protomatrices

In the previous two subsections, we have analyzed the effect of non-zero $\mathbf{B}_{L_2}$ on the source decoding threshold p_{th} under a regular or irregular source protomatrix. With the same number of non-zero columns in $\mathbf{B}_{L_2}$, results have shown that the CSVL protomatrix with the irregular $\mathbf{B}_s^{[18]}$ can achieve a higher p_{th} than that with the regular $\mathbf{B}_s^{reg}$. Optimized in [18] for p_{th} (and hence low error floor) under the assumption that $\mathbf{B}_{L_2} = \mathbf{0}$, the irregular $\mathbf{B}_s^{[18]}$ used in the previous subsection may not be optimal for $\mathbf{B}_{L_2} \neq \mathbf{0}$. In this subsection, we attempt to optimize $\mathbf{B}_s$ for a given $\mathbf{B}_{L_2}$.

1) Optimization algorithm: For easy comparison with results in the previous subsection, we use similar $\mathbf{B}_{L_2}$ designs here, i.e., $\mathbf{B}_{L_2}$ is assumed to have one, two or three non-zero columns and each non-zero column has a weight of 2. Without loss of generality, we further assume that the non-zero-column sets are given by $\psi_1 = \{1\}$, $\psi_2 = \{1,2\}$ and $\psi_3 = \{1,2,3\}$, corresponding to, respectively, one, two and three non-zero

TABLE III
NON-ZERO COLUMN SET ψ OF $\mathbf{B}_{L_2}$ AND THE CORRESPONDING SOURCE DECODING THRESHOLD p_{th} OF THE CSVL PROTOMATRIX IN (26).

Case no.	Non-zero column set ψ	p_{th}
case-1	$\{3\}$	0.1156
case-2	$\{7\}$	0.1101
case-3	$\{8\}$	0.1058
case-4	$\{1\}$	0.1031
case-5	$\{2,4\}$	0.1498
case-6	$\{2,7\}$	0.1444
case-7	$\{3,8\}$	0.1361
case-8	$\{7,8\}$	0.1299
case-9	$\{1,3,5\}$	0.2076
case-10	$\{2,6,8\}$	0.1686
case-11	$\{3,5,7\}$	0.1586
case-12	$\{4,5,8\}$	0.1472

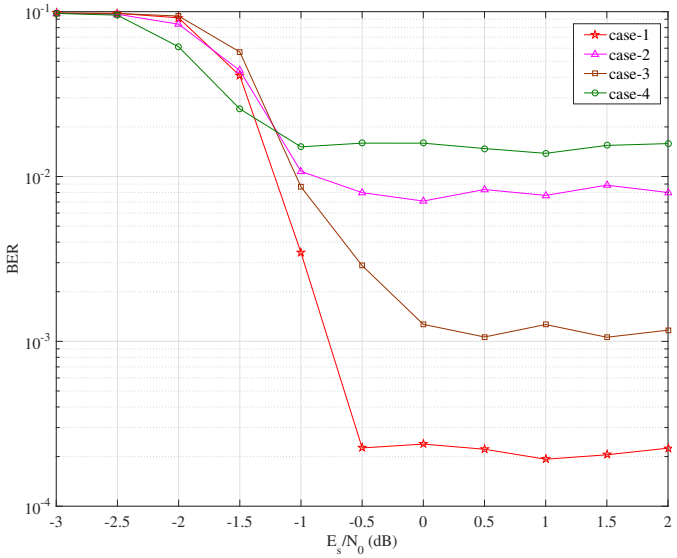

Fig. 7. BER performance of the DP-LDPC codes formed by (26) and (25). $\mathbf{B}_{L_2}$ in (26) contains one non-zero column and the non-zero column sets are shown in Table III. Source statistic $p = 0.10$.

columns; and the empty non-zero-column set is denoted by $\psi_0 = \{\varnothing\}$, corresponding to $\mathbf{B}_{L_2} = \mathbf{0}$.

As for $\mathbf{B}_s$, we retain the weight-2 columns of $\mathbf{B}_s^{[18]}$ because such columns can improve error-floor performance [18]. We also retain them in the same columns, i.e., 6th, 7th, and 8th columns, because our previous subsection results indicate that non-zero-weight columns in $\mathbf{B}_{L_2}$ are not recommended to align with weight-2 columns in the irregular source protomatrix. Thus, our optimized irregular protomatrix $\mathbf{B}_s^{opt}$ has the following structure

$$\mathbf{B}_s^{opt} = \begin{bmatrix} b_s^{1,1} & b_s^{1,2} & b_s^{1,3} & b_s^{1,4} & b_s^{1,5} & 1 & 0 & 0 \\ b_s^{2,1} & b_s^{2,2} & b_s^{2,3} & b_s^{2,4} & b_s^{2,5} & 0 & 1 & 0 \\ b_s^{3,1} & b_s^{3,2} & b_s^{3,3} & b_s^{3,4} & b_s^{3,5} & 0 & 0 & 1 \\ b_s^{4,1} & b_s^{4,2} & b_s^{4,3} & b_s^{4,4} & b_s^{4,5} & 1 & 1 & 1 \end{bmatrix}, \tag{27}$$

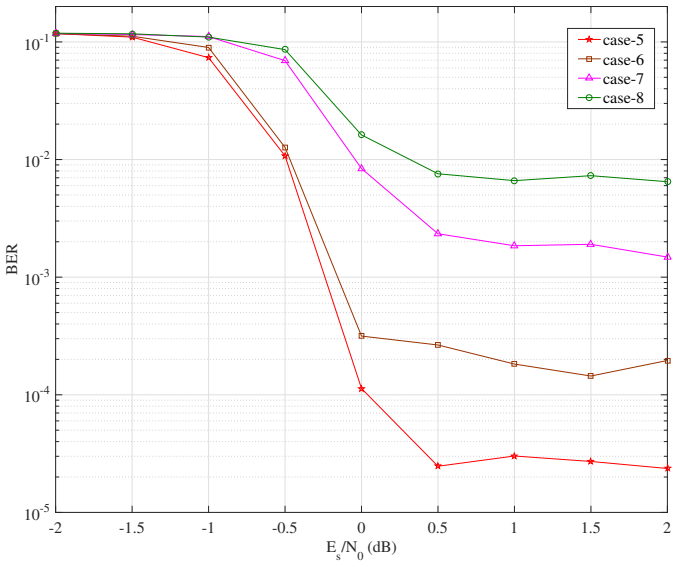

Fig. 8. BER performance of the DP-LDPC codes formed by (26) and (25). $\mathbf{B}_{L_2}$ in (26) contains two non-zero columns and the non-zero column sets are shown in Table III. Source statistic $p = 0.12$.

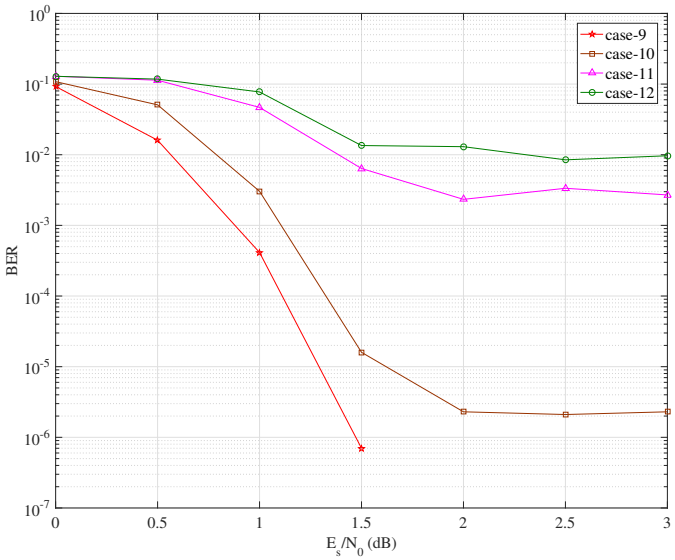

Fig. 9. BER performance of the DP-LDPC codes formed by (26) and (25). $\mathbf{B}_{L_2}$ in (26) contains three non-zero columns and the non-zero column sets are shown in Table III. Source statistic $p = 0.14$.

where $b_s^{i,j} = b_s(i,j)$ denotes the (i,j)-th element of $\mathbf{B}_s^{opt}$.

To search for the remaining elements in $\mathbf{B}_s^{opt}$, we propose a differential evolution [34], [35] searching algorithm, details of which is given in **Algorithm 2**. Moreover, the source decoding threshold p_{th} found by the GSP-EXIT algorithm is used as the cost function, i.e.,

$$\begin{aligned} &\max_{\mathbf{B}_s, \mathbf{B}_{L_2}} \ \Phi(\mathbf{B}_s, \mathbf{B}_{L_2}) \\ &\text{s.t.} \quad \Omega(\mathbf{B}_s, \mathbf{B}_{L_2}) = 1 \end{aligned} \tag{28}$$

where $\Phi(\mathbf{B}_s, \mathbf{B}_{L_2})$ returns the source threshold p_{th}; $\Omega(\mathbf{B}_s, \mathbf{B}_{L_2}) = 1$ when all the required conditions in **Algorithm 2** are satisfied, and $\Omega(\mathbf{B}_s, \mathbf{B}_{L_2}) = 0$ otherwise.

TABLE IV
SOURCE DECODING THRESHOLD (p_{th}) UNDER DIFFERENT COMBINATIONS OF $\mathbf{B}_s$ AND $\mathbf{B}_{L_2}$. $\mathbf{B}_s$ IS GIVEN AS IN (26), AND (29) TO (31). $\mathbf{B}_{L_2}$ CONTAINS ONE, TWO OR THREE NON-ZERO COLUMNS AND EACH NON-ZERO COLUMN HAS A WEIGHT OF 2. ψ DENOTES THE SET OF NON-ZERO COLUMNS IN $\mathbf{B}_{L_2}$.

Set of non-zero columns	$\mathbf{B}_s^{[18]}$	$\mathbf{B}_s^{opt-1}$	$\mathbf{B}_s^{opt-2}$	$\mathbf{B}_s^{opt-3}$
$\psi = \{\varnothing\}$	**0.0977**	0.0867	0.0859	0.0764
$\psi = \{1\}$	0.1156	**0.1227**	0.1152	0.1018
$\psi = \{1, 2\}$	0.1498	0.1409	**0.1543**	0.1439
$\psi = \{1, 2, 3\}$	0.2076	0.2048	0.2033	**0.2148**

The searching space and complexity increase with the parameters b_s^{max}, K_c and K_g. In this study, we set $b_s^{max} = 3$, $K_c = 5000$, $K_g = 5000$ to limit the searching space. We also assume a cross-over probability of 0.85, i.e., $p_c = 0.85$.

2) Optimized results and comparisons: Using **Algorithm 2**, the optimized source protographs corresponding to the non-zero-column sets (of $\mathbf{B}_{L_2}$) $\psi_1 = \{1\}$, $\psi_2 = \{1, 2\}$ and $\psi_3 = \{1, 2, 3\}$ are found to be, respectively,

$$\mathbf{B}_s^{opt-1} = \begin{bmatrix} 3 & 3 & 0 & 2 & 0 & 1 & 0 & 0 \\ 2 & 3 & 1 & 3 & 1 & 0 & 1 & 0 \\ 2 & 2 & 0 & 3 & 0 & 0 & 0 & 1 \\ 1 & 2 & 2 & 0 & 2 & 1 & 1 & 1 \end{bmatrix}, \tag{29}$$

$$\mathbf{B}_s^{opt-2} = \begin{bmatrix} 3 & 1 & 0 & 3 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 3 & 2 & 0 & 1 & 0 \\ 3 & 2 & 2 & 3 & 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 3 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{30}$$

and

$$\mathbf{B}_s^{opt-3} = \begin{bmatrix} 3 & 0 & 2 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 3 & 3 & 0 & 1 & 0 \\ 0 & 3 & 3 & 2 & 3 & 0 & 0 & 1 \\ 2 & 0 & 3 & 1 & 2 & 1 & 1 & 1 \end{bmatrix}. \tag{31}$$

Using $\mathbf{B}_s^{[18]}$ and the above optimized $\mathbf{B}_s$, we form CSVL protomatrices under $\psi_0 = \{\varnothing\}$, $\psi_1 = \{1\}$, $\psi_2 = \{1, 2\}$ and $\psi_3 = \{1, 2, 3\}$ and evaluate the corresponding p_{th}. The results are listed in Table IV. Note that the p_{th} values corresponding to $\mathbf{B}_s^{[18]}$ (and ψ_1, ψ_2 and ψ_3) are simply the optimal values listed in Table II. We can observe in Table IV that each $\mathbf{B}_s$ is optimized at one particular non-zero-column set. For example, the $\mathbf{B}_s^{opt-2}$ achieves the highest p_{th} under ψ_2, and has a smaller p_{th} than (i) $\mathbf{B}_s^{opt-1}$ under ψ_1; and (ii) $\mathbf{B}_s^{opt-3}$ under ψ_3. We can also see than $\mathbf{B}_s^{[18]}$ is only optimized at ψ_0.

In Figs. 10 to 12, we plot the BER curves of the DP-LDPC codes formed by some of the above CSVL protomatrices together with (25). Fig. 10, Fig. 11 and Fig. 12 correspond to the results when $\mathbf{B}_{L_2}$ consists of one, two and three non-zero columns, respectively. Compared with using the optimized $\mathbf{B}_s$ in [18] ($\mathbf{B}_s^{[18]}$) in the DP-LDPC codes, using the $\mathbf{B}_s$ optimized by **Algorithm 2** ($\mathbf{B}_s^{opt-1}$ to $\mathbf{B}_s^{opt-3}$) results in lower error floors under the same source statistic p. The error floor improvements are expected to be larger when the source statistic p decreases. Moreover, if the search space is made larger, say by increasing b_s^{max} from 3 to 4, the improvement in the source threshold and hence error floor will be more significant.

Algorithm 2 Differential evolution searching algorithm for the source protograph $\mathbf{B}_s^{opt}$

GIVEN:

- an initial source protograph $\mathbf{B}_s$ with some fixed columns (6th, 7th, and 8th columns in our study) and all other entries set to 1;
- $\mathbf{B}_{L_2}$ under $\psi_1 = \{1\}$, $\psi_2 = \{1, 2\}$ or $\psi_3 = \{1, 2, 3\}$;
- the number of candidate matrices K_c;
- the number of generations K_g;
- the crossover probability p_c;
- $\mathbf{B}_s^{k_c(k_g)}$ denote the k_c-th candidate source protograph matrix in the k_g-th generation.

REQUIRED CONDITIONS:

- The minimum column weight of the source protograph is 3 (except the fixed columns).
- The minimum row weight of the source protograph is 2.
- Except the fixed entries, the other entries in $\mathbf{B}_s$ has a maximum value of b_s^{max}.

Step 1 *INITIALIZATION*: Set $k_g = 0$ and $\mathbf{B}_s^{1(k_g)} = \mathbf{B}_s$. For $k_c = 2, 3, \ldots, K_c$, generate $\mathbf{B}_s^{k_c(k_g)}$ by replacing the elements in $\mathbf{B}_s^{1(0)}$ (except those in the fixed columns) with random integers between 0 and b_s^{max}. Repeat the process until all $\mathbf{B}_s^{k_c(k_g)}$ satisfy the required conditions.

Step 2 *MUTATION*: For $k_c = 1, 2, \ldots, K_c$, generate a mutation matrix $\mathbf{M}_s^{k_c(k_g)}$ using

$$\mathbf{M}_s^{k_c(k_g)} = \Theta\Big(\mathbf{B}_s^{r_1(k_g)} + 0.5(\mathbf{B}_s^{r_2(k_g)} - \mathbf{B}_s^{r_3(k_g)})\Big)$$

where r_1, r_2 and r_3 are randomly integers in the range $[1, K_c]$, and $\Theta(\mathbf{B})$ is an operation that replaces each element in $\mathbf{B}$ with an integer closest to its absolute value.

Step 3 *CROSSOVER*: For $k_c = 1, 2, \ldots, K_c$, the (i, j)-th entry of a candidate matrix $\mathbf{N}_s^{k_c(k_g)}$ is set as the (i, j)-th element of $\mathbf{M}_s^{k_c(k_g)}$ with a crossover probability p_c, or as the (i, j)-th element of $\mathbf{B}_s^{k_c(k_g)}$ with probability $(1 - p_c)$.

Step 4 *SELECTION*: For $k_c = 1, 2, \ldots, K_c$, protographs of the $(k_g + 1)$-th generation are set according to

$$\mathbf{B}_s^{k_c(k_g+1)} = \begin{cases} \mathbf{N}_s^{k_c(k_g)} & \text{if } \Phi(\mathbf{N}_s^{k_c(k_g)}, \mathbf{B}_{L_2}) \times \Omega(\mathbf{N}_s^{k_c(k_g)}, \mathbf{B}_{L_2}) \\ & \qquad > \Phi(\mathbf{B}_s^{k_c(k_g)}, \mathbf{B}_{L_2}) \\ \mathbf{B}_s^{k_c(k_g)} & \text{otherwise,} \end{cases}$$

where $\Phi(\mathbf{B}_s, \mathbf{B}_{L_2})$ returns the source threshold p_{th}; $\Omega(\mathbf{B}_s, \mathbf{B}_{L_2}) = 1$ when all the required conditions are satisfied, and $\Omega(\mathbf{B}_s, \mathbf{B}_{L_2}) = 0$ otherwise.

Step 5 *TERMINATION*: Steps 2 to 4 are executed for K_g generations and the protograph with the highest source decoding threshold p_{th} is chosen as the optimal protograph $\mathbf{B}_s^{opt}$.

Next, we compare the decoding complexity of the DP-LDPC codes based on $\mathbf{B}_s^{[18]}$, $\mathbf{B}_s^{opt-1}$, $\mathbf{B}_s^{opt-2}$ and $\mathbf{B}_s^{opt-3}$. It is shown that the decoding complexity of DP-LDPC codes depends mainly on the average degree of CNs [36]. Our results have indicated that the average row weights of $\mathbf{B}_s^{[18]}$, $\mathbf{B}_s^{opt-1}$,

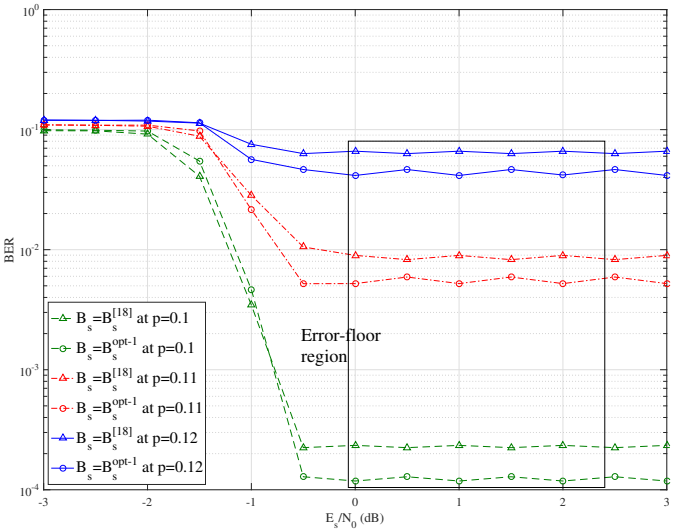

Fig. 10. BER performance of the DP-LDPC codes formed by $\mathbf{B}_s^{[18]}/\mathbf{B}_s^{opt-1}$, $\mathbf{B}_{L_2}$, and (25). $\mathbf{B}_{L_2}$ contains one non-zero column, i.e., $\psi_1 = \{1\}$. Source statistic $p = 0.10, 0.11$ and 0.12.

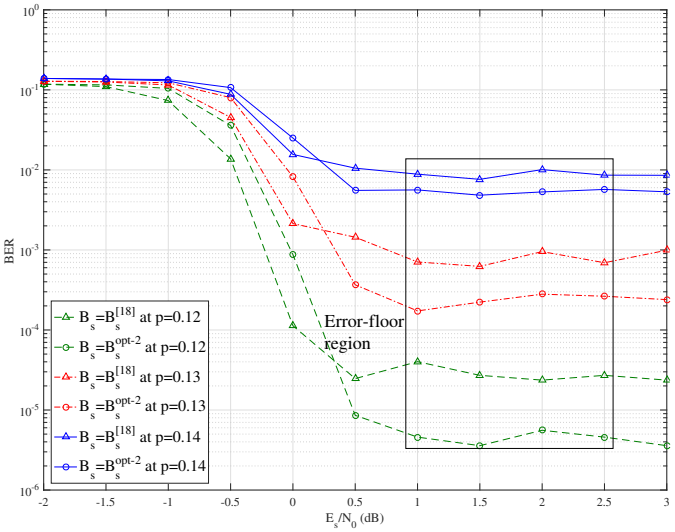

Fig. 11. BER performance of the DP-LDPC codes formed by $\mathbf{B}_s^{[18]}/\mathbf{B}_s^{opt-2}$, $\mathbf{B}_{L_2}$, and (25). $\mathbf{B}_{L_2}$ contains two non-zero columns, i.e., $\psi_2 = \{1, 2\}$. Source statistic $p = 0.12, 0.13$ and 0.14.

$\mathbf{B}_s^{opt-2}$ and $\mathbf{B}_s^{opt-3}$ are, respectively, $8.5, 9.0, 8.75$, and 9.375. Thus, we expect a modest increase in decoding complexity when the optimized codes are used. As for the encoding complexity, the increase will be very minimal because only logical operations on "0"s and "1"s are involved in the process.

VI. BER Performance over a Rayleigh Fading Channel

In this section, we evaluate the BER performance of the DP-LDPC codes over a flat Rayleigh fading channel model. Under this model, the received signal r is given by $r = hx+n$ where x is the BPSK-modulated signal, n is the AWGN noise, and h is a normalized Rayleigh fading factor with $\mathbb{E}[h^2] = 1$ ($\mathbb{E}[\cdot]$ is the expectation operation). Moreover, the probability

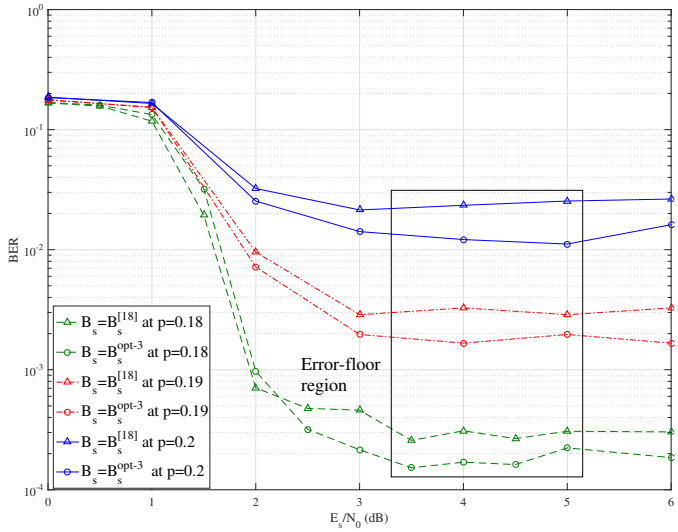

Fig. 12. BER performance of the DP-LDPC codes formed by $\mathbf{B}_s^{[18]}/\mathbf{B}_s^{opt-3}$, $\mathbf{B}_{L_2}$, and (25). $\mathbf{B}_{L_2}$ contains three non-zero columns, i.e., $\psi_3 = \{1, 2, 3\}$. Source statistic $p = 0.18, 0.19$ and 0.2.

density function of h is given by $f(h) = 2h\exp(-h^2)$. If the side information (SI) h is perfectly known by the receiver, the LLR of the received signal can be calculated by $z_c = 2rh/\sigma^2$; otherwise the LLR is approximated by $z_c \approx 2r\mathbb{E}[h]/\sigma^2$, where $\mathbb{E}[h] = 0.8862$ [37].

In Fig. 13, we plot the BER curves of DP-LDPC codes formed by $\mathbf{B}_s^{[18]}$, $\mathbf{B}_{L_2}$ under case-1 or case-5 shown in Table III, and (25). We can observe that a Rayleigh fading channel degrades the BER performance compared with an AWGN channel. The performance loss is more severe when no SI is available. For example, we consider case-1 under a source statistic $p = 0.10$ and a BER 10^{-3}. Under an AWGN channel $E_s/N_0 = -0.8$ dB is required. Under a Rayleigh fading channel, E_s/N_0 degrades by about 2.5 dB if SI is available; and by 3.8 dB if SI is not available. We also observe that the error floor remains at the same level, regardless of AWGN or fading channels. It is because the error floor is caused by source compression rather than the channel fading when E_s/N_0 is large. Similar observations are found for case-5.

In Fig. 14, we plot the BER curves of DP-LDPC codes formed by (i) $\mathbf{B}_s^{opt-1}$ and $\mathbf{B}_{L_2}$ with $\psi_1 = \{1\}$, or (ii) $\mathbf{B}_s^{opt-2}$ and $\mathbf{B}_{L_2}$ with $\psi_2 = \{1, 2\}$; and (25) under AWGN and Rayleigh fading channels. Similar observations as above are found, i.e., a Rayleigh fading channel degrades the BER performance compared with an AWGN channel; and the performance loss is more severe when no SI is available.

VII. Conclusion

In this paper, we investigated the error-floor issue of DP-LDPC codes. We proposed a generalized source protograph EXIT (GSP-EXIT) algorithm for evaluating the source decoding threshold of DP-LDPC codes. With the GSP-EXIT algorithm, we optimized DP-LDPC code structures (with both regular or irregular source protomatrices) and achieved lower error floors. Our analytical findings were also validated

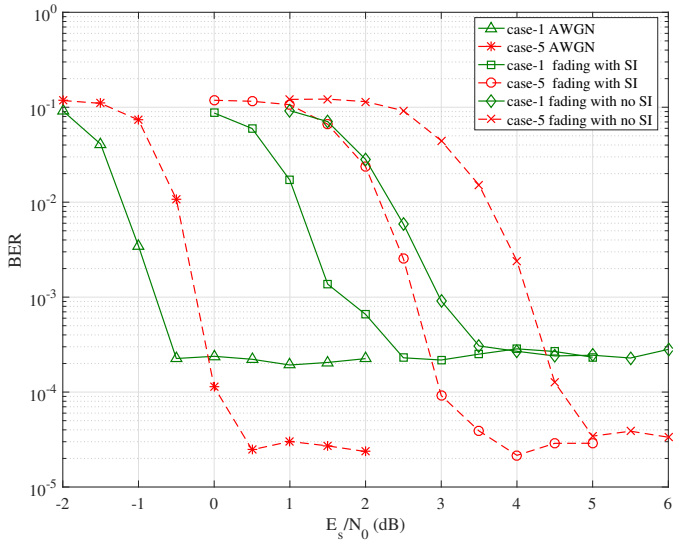

Fig. 13. BER curves of DP-LDPC codes formed by $\mathbf{B}_s^{[18]}$, $\mathbf{B}_{L_2}$ under case-1 and case-5 in Table III, and (25) under AWGN and Rayleigh fading channels. Solid curves: case-1 at source statistic $p = 0.10$; dashed curves: case-5 at source statistic $p = 0.12$.

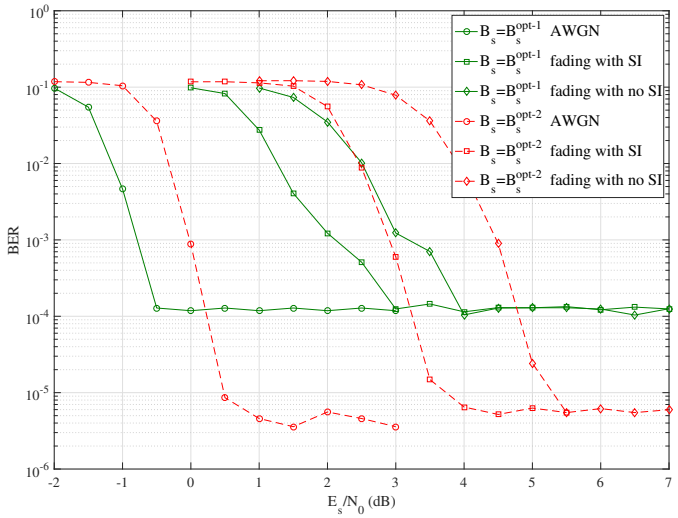

Fig. 14. BER curves of DP-LDPC codes formed by (a) $\mathbf{B}_s^{opt-1}$ and $\mathbf{B}_{L_2}$ with $\psi_1 = \{1\}$, or (b) $\mathbf{B}_s^{opt-2}$ and $\mathbf{B}_{L_2}$ with $\psi_2 = \{1, 2\}$; and (25) under AWGN and Rayleigh fading channels. Solid curves: $\mathbf{B}_s^{opt-1}$ and $\mathbf{B}_{L_2}$ with $\psi_1 = \{1\}$ at source statistic $p = 0.10$; dashed curves: $\mathbf{B}_s^{opt-2}$ and $\mathbf{B}_{L_2}$ with $\psi_2 = \{1, 2\}$ at source statistic $p = 0.12$.

by simulations results. Our study again shown dependency between error-floor levels and waterfall region performances of DP-LDPC codes. In the future, we will try to establish a relationship between these two performance indicators and propose a joint optimization techniques under some given conditions.

REFERENCES

[1] A. Zribi, R. Pyndiah, S. Zaibi, F. Guilloud, and A. Bouallegue,"Low-complexity soft decoding of Huffman codes and iterative joint source channel decoding," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1669–1679, June 2012.

[2] C. Bi and J. Liang, "Joint source-channel coding of JPEG 2000 image transmission over two-way multi-relay networks," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3594–3608, July 2017.

[3] I. Bocharova, A.G. Fabregas, B.D. Kudryashov, A. Martinez, A.T. Campo and G. Vazquez-Vilar,, "Source-channel coding with multiple classes," in *Proc. IEEE International Symposium on Information Theory*, Honolulu, USA, Aug. 2014.

[4] M. Fresia, F. Perez-cruz, H. V. Poor and S. Verdu, "Joint source and channel coding," *IEEE Signal Processing Mag.*, vol. 27, no. 6, pp. 104–113, Nov. 2010.

[5] Y. Jang, J. Jeong, and D. Yoon,"Bit error floor of MPSK in the presence of phase error", *IEEE Trans. Vehi. Tech.*, vol. 65, no. 5, pp. 3782–3786, May 2016.

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

[7] Y. Fang, P. Chen, G. Cai, F. C. M. Lau, S. C. Liew, and G. Han, "Outage limit-approaching channel coding for future wireless communications: Root-protograph low-density parity-check codes," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 85–93, 2019.

[8] Y. Fang, S. Liew, and T. Wang, "Design of distributed protograph LDPC codes for multi-relay coded-cooperative networks," *IEEE Trans. Wireless Commun.* vol. 16, no. 11, pp. 7235–7251, Nov. 2017.

[9] Q. Chen, L. Wang, S. Hong, and Z. Xiong, "Performance improvement of JSCC scheme through redesigning channel codes," *IEEE Commun. Letters*, vol. 20, No. 6, pp. 1088–1091, June 2016.

[10] S. Liu, C. Chen, L. Wang, and S. Hong, "Edge connection optimization for JSCC system based on DP-LDPC codes," *IEEE Wireless Commun. Letters*, vol. 8, no. 4, pp. 996–999, Aug. 2019.

[11] S. Hong, Q. Chen, and L. Wang, "Performance analysis and optimization for edge connection of JSCC system based on double protograph LDPC codes," *IET Commun.*, vol. 12, Iss. 2, pp. 214–219, Jan. 2017.

[12] C. Chen, L. Wang, and F. C. M. Lau, "Joint optimization of protograph LDPC code pair for joint source and channel coding," *IEEE Trans. on Commun.*, vol. 66, no. 8, pp. 3255–3267, Aug. 2018.

[13] J. He, Y. Li, G. Wu, S. Qian, Q. Xue, and T. Matsumoto, "Performance improvement of joint source-channel coding with unequal power allocation," *IEEE Wireless Commun. Letters*, vol. 6, no. 5, pp. 582–585, June 2017.

[14] H. Neto and W. Henkel,"Information shortening for joint source-channel coding schemes based on low-density parity-check codes," *8th International Symp. on Turbo Codes and Iterative Info. Proc.* (*ISTC*), 2014, Bremen, Germany.

[15] A. Golmohammadi and D. Mitchell, "Concatenated spatially coupled LDPC codes for joint source-channel coding," *Proc. IEEE International Symposium Information Theory (ISIT)*, 2018, Vail, USA.

[16] H. Neto and W. Henkel, "Multi-edge optimization of low-density parity-check codes of joint source-channel coding," in *9th International ITG Conference on Systems, Communication and Coding*, 2013, Munich, Germany.

[17] Q. Chen, S. Hong, and Y. Chen, "Design of linking matrix in JSCC scheme based on double protograph LDPC codes," *IEEE Access*, vol. 7, pp. 92176–92183, July 2019.

[18] C. Chen, L. Wang, and S. Liu, "The design of protograph LDPC codes as source codes in a JSCC system," *IEEE Commun. Letters*, vol. 22, no. 4, pp. 672–675, April 2018.

[19] D. J. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in *Proc. 5th IMA Conf. Cryptography Coding, Number 1025 Lecture Notes Comput. Sci.*, pp. 100–111, Oct. 1995.

[20] T. J. Richardson and R. L. Urbanke, "Capacity of low-density parity-check codes under message passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[21] Y. Fang, G. A. Bi, Y. L. Guan, and F. C. M. Lau, "A survey on protograph LDPC codes and their applications," *IEEE Commun. Surv. Tut.*, vol. 17, no. 4, pp. 1989–2016, 2015.

[22] Z. Yang, Y. Fang, G. Zhang, F. C. M. Lau, S. Mumtaz and D. B. da Costa, "Analysis and optimization of tail-biting spatially coupled protograph LDPC codes for BICM-ID systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 390–404, Jan. 2020.

[23] G. Yue, L. Ping, and X. Wang, "Generalized low-density parity-check codes based on Hadamard constraints," *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 1058–1079, March 2007.

[24] P. W. Zhang, F. C. M. Lau, and C.-W. Sham "Protograph-based LDPC-Hadamard Codes," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Seoul, Korea, 2020, pp. 1–6.

[25] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Letters*, vol. 6, no. 10, pp. 440–443, Oct. 2002.

[26] F. Ye, E. Dupraz, Z. Mheich, and K. Amis, "Optimized rate-adaptive protograph-based LDPC codes for source coding with side information," *IEEE Trans. on Commun.*, vol. 67, no. 6, pp.3879–3889, June 2019.

[27] M. Luby, "LT codes," *in Proc. 43rd Symp. Found. Comput. Sci.* (FOCS), Washington DC, USA, 2002, pp. 271–280.

[28] M. Fresia, L. Vandendorpe and H. V. Poor, "Distributed source coding using Raptor codes for hidden Markov sources," *IEEE Trans. on Signal Process.*, vol. 57, no. 7, pp. 2868–2875, July 2009.

[29] J. Wu, C. Yuen, M. Wang, J. Chen and C. W. Chen, "TCP-oriented Raptor coding for high-frame-rate video transmission over wireless networks," *IEEE J. Sel. Areas in Commun.*, vol. 34, no. 8, pp. 2231–2246, Aug. 2016.

[30] J. Wu, B. Cheng and M. Wang, "Improving multipath video transmission with Raptor codes in heterogeneous wireless networks," *IEEE Trans. on Multimedia*, vol. 20, no. 2, pp. 457–472, Feb. 2018.

[31] Q. Xu, V. Stankovic, and Z. Xiong, "Distributed joint source-channel coding of video using Raptor codes," *IEEE J. Sel. Areas in Commun.*, vol. 25, no. 4, pp. 851–861, May 2007.

[32] Nasriuminallsh and L. Hanzo, "Near-capacity H.264 multimedia communications using iterative joint source-channel decoding," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 538–564, 2nd Quart., 2012.

[33] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, pp. 386–398, Jan. 2005.

[34] 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.

[35] 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.

[36] Q. Chen and L. Wang, "Design and analysis of joint source channel coding schemes over non-standard coding channels," *IEEE Trans. on Vehi. Tech.*, vol. 69, no. 5, pp. 5369–5380, May 2020.

[37] J. Hou, P. H. Siegel, and L. B. Milstein, "Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels," *IEEE. J. Sel. Areas on Commun.*, vol. 19, no. 5, pp. 924–934, May 2001.