

## Article

# Design of Low-Density Parity-Check Code Pair for Joint Source-Channel Coding Systems Based on Graph Theory

Yijie Lv <sup>1</sup>, Jiguang He <sup>2,3</sup>, Weikai Xu <sup>1,\*</sup> and Lin Wang <sup>1</sup>

<sup>1</sup> Department of Information and Communication Engineering, Xiamen University, Xiamen 361005, China; lvsejianke@126.com (Y.L.); wanglin@xmu.edu.cn (L.W.)

<sup>2</sup> Technology Innovation Institute, Abu Dhabi P.O. Box 9639, United Arab Emirates; jiguang.he@tii.ae

<sup>3</sup> Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland

\* Correspondence: xweikai@xmu.edu.cn

**Abstract:** In this article, a graph-theoretic method (taking advantage of constraints among sets associated with the corresponding parity-check matrices) is applied for the construction of a double low-density parity-check (D-LDPC) code (also known as LDPC code pair) in a joint source-channel coding (JSCC) system. Specifically, we pre-set the girth of the parity-check matrix for the LDPC code pair when jointly designing the two LDPC codes, which are constructed by following the set constraints. The constructed parity-check matrices for channel codes comprise an identity submatrix and an additional submatrix, whose column weights can be pre-set to be any positive integer numbers. Simulation results illustrate that the constructed D-LDPC codes exhibit significant performance improvement and enhanced flexible frame length (i.e., adaptability under various channel conditions) compared with the benchmark code pair.

**Keywords:** joint source-channel coding; low-density parity-check code; graph theory



**Citation:** Lv, Y.; He, J.; Xu, W.; Wang, L. Design of Low-Density Parity-Check Code Pair for Joint Source-Channel Coding Systems Based on Graph Theory. *Entropy* **2023**, *25*, 1189. <https://doi.org/10.3390/e25081189>

Academic Editor: T. Aaron Gulliver

Received: 29 June 2023

Revised: 26 July 2023

Accepted: 7 August 2023

Published: 10 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A joint source-channel coding (JSCC) system is capable of overcoming the shortcomings of a Shannon separation coding system and enhancing the performance of the digital communication system via joint decoding [1]. In general, a JSCC system can be realized with different combinations of source and channel codes. For instance, examples of these combinations are a variable-length code (VLC) cascaded with a convolutional code (CC) [2,3]; a VLC cascaded with a Turbo code [4–6]; a VLC cascaded with a low-density parity-check (LDPC) code [7,8]; and an LDPC code (falling into the category of fixed-length codes) cascaded with another LDPC code [9,10], termed as double LDPC (D-LDPC) codes.

Among these realizations, the D-LDPC code-oriented JSCC system has been intensively investigated since its first appearance in [9,10], where a joint extrinsic information transfer (JEXIT) chart was employed to enhance the waterfall performance of the code pair (one LDPC code for source code and another LDPC code for channel code). To reduce the hardware implementation complexity of the JSCC system and to improve its performance, double protograph LDPC (DP-LDPC) codes for JSCC were extensively studied in [11–16]. The source protograph LDPC codes were optimized to improve the error floor of the JSCC system by maximizing the source decoding threshold [12]. Increasing the mutual information (MI) among the source code and the channel code by redesigning the channel protograph LDPC codes leads to performance improvements in the waterfall region [13]. To reduce the transmission latency and decoding complexity, a concatenation of spatially coupled LDPC codes with sliding window decoding for JSCC was proposed in [17]. However, the aforementioned JSCC systems heavily rely on analyses of protograph LDPC codes and JEXIT. Moreover, the candidate frame length of the JSCC based on DP-LDPC code must be restricted to an integer fold of the column number of the base matrix.

Combinatorial mathematics is an important tool for constructing LDPC codes, which includes the graph-theoretic method, combinatoric designs, finite geometries, and difference sets [18–21]. As an essential construction method for LDPC codes, the channel LDPC codes constructed with the graph-theoretic method achieve promising performance [22].

In this paper, a construction method of the code pair is developed based on graph theory, which transforms parity-check matrices of the source code and the channel code into inter-set constraint problem and selects the sets which satisfy the constraint. In order to match the parity-check matrix of the source code and the generation matrix of the channel code in the encoding process, the channel code is first constructed during the construction of the code pair via the graph-theoretic method. In contrast to the conventional method, the proposed one can be obtained without constructing the base matrix and performing ‘copy and permute’ operations. Since the proposed channel code takes the systematic form, the generator matrix is easily obtained by transposing the parity-check matrix. In addition, the proposed code pair is constructed with set constraints from the graph-theoretic method, which brings a more adaptive/flexible frame length than the conventional protograph code pair.

## 2. Preliminaries of JSCC

### 2.1. Representations of JSCC

The D-LDPC system consists of two LDPC codes: one for source code and the other for channel code. As a linear block code, a parity-check matrix can be used to express the LDPC code. A systematic LDPC code can be completely characterized by a generator matrix  $\mathbf{G} \in \mathbb{B}^{k \times n}$  with  $\mathbb{B} = \{0, 1\}$ , which corresponds to a parity-check matrix  $\mathbf{H} \in \mathbb{B}^{(n-k) \times n}$  of the form in (1), where  $\mathbf{P}^T$  is the transpose of the matrix  $\mathbf{P}$  and  $\mathbf{I}_{n-k}$  is the identity matrix with dimension  $(n - k) \times (n - k)$ .

$$\mathbf{H} = [\mathbf{I}_{n-k} | \mathbf{P}^T] = \left[ \begin{array}{cccc|cccc} 1 & 0 & 0 & \cdots & 0 & p_{11} & p_{21} & \cdots & p_{k1} \\ 0 & 1 & 0 & \cdots & 0 & p_{12} & p_{22} & \cdots & p_{k2} \\ 0 & 0 & 1 & \cdots & 0 & p_{13} & p_{23} & \cdots & p_{k3} \\ & & & \vdots & & & & & \\ 0 & 0 & 0 & \cdots & 1 & p_{1(n-k)} & p_{2(n-k)} & \cdots & p_{k(n-k)} \end{array} \right]_{(n-k) \times n} \quad (1)$$

The parity-check matrix of the JSCC system is represented as

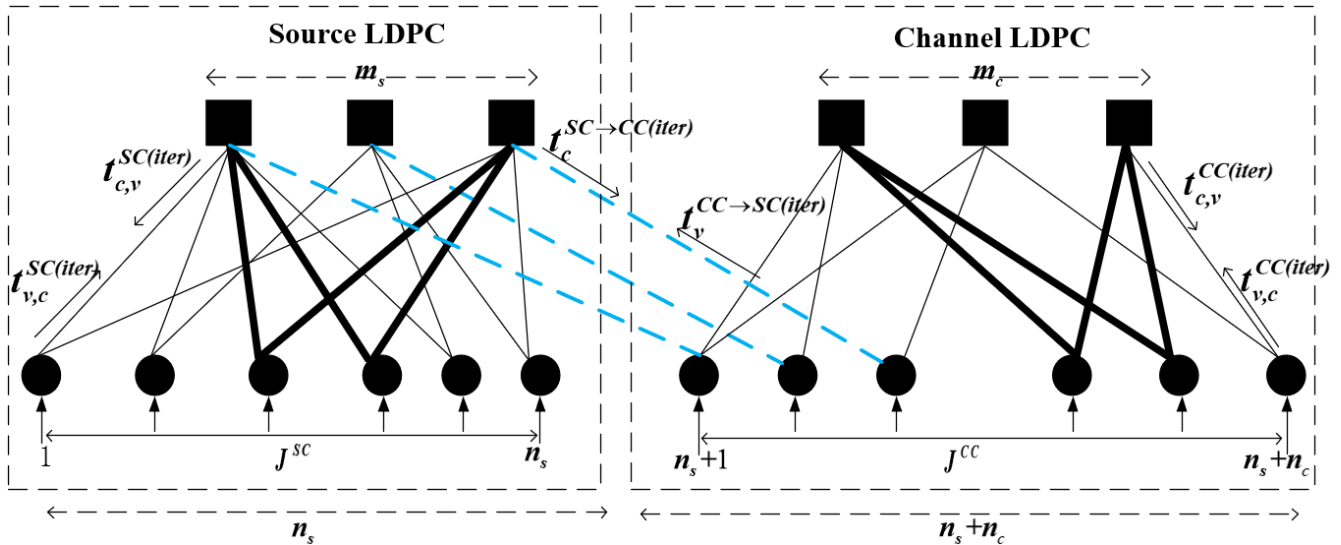
$$\mathbf{H}_J = \left[ \begin{array}{c|c} \mathbf{H}_S & \mathbf{H}_L \\ \hline \mathbf{0} & \mathbf{H}_C \end{array} \right], \quad (2)$$

which includes three non-zero submatrices, i.e.,  $\mathbf{H}_S$  with dimension  $m_s \times n_s$ ,  $\mathbf{H}_C$  with dimension  $m_c \times n_c$ , and  $\mathbf{H}_L$  with dimension  $m_s \times n_c$ . Therefore, the overall dimension of  $\mathbf{H}_J$  is  $(m_s + m_c) \times (n_s + n_c)$ . A simple example of a parity-check matrix  $\mathbf{H}_J$  is provided in (3).

$$\mathbf{H}_J = \left[ \begin{array}{cccccc|cccccc} 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} \right]. \quad (3)$$

The Tanner graph associated with the parity-check matrix  $\mathbf{H}_J$  is depicted in Figure 1. Six types of MI in Figure 1 are defined as follows:

- \*  $t_{v,c}^{SC(iter)}$ : MI from the variable node (VN) of the source code to the check node (CN) of the source code.
- \*  $t_{c,v}^{SC(iter)}$ : MI from the CN of the source code to the VN of the source code.
- \*  $t_c^{SC \rightarrow CC(iter)}$ : MI from the CN of the source code to the VN of the channel code.
- \*  $t_v^{CC \rightarrow SC(iter)}$ : MI from the VN of the channel code to the CN of the source code.
- \*  $t_{v,c}^{CC(iter)}$ : MI from the VN of the channel code to the CN of the channel code.
- \*  $t_{c,v}^{CC(iter)}$ : MI from the CN of the channel code to the VN of the channel code.



**Figure 1.** Tanner graphical representation of the JSCC system realized according to exemplary  $\mathbf{H}_J$  in (3).

## 2.2. Encoding and Decoding of JSCC Systems

### 2.2.1. Encoder

The source sequence follows the binomial Bernoulli ( $p$ ) distribution, whose entropy is expressed as

$$H = -p \log_2 p - (1 - p) \log_2 (1 - p), \tag{4}$$

with  $p$  ( $p \neq 1/2$ ) being the probability of ‘1’.

The encoding process for the LDPC code pair is represented by

$$\mathbf{c} = \mathbf{G}_C^T \mathbf{o} = \mathbf{G}_C^T \mathbf{H}_S \mathbf{s}, \tag{5}$$

where  $\mathbf{s} \in \mathbb{B}^{n_s}$  denotes the source sequence, while  $\mathbf{o} \in \mathbb{B}^{m_s}$  stands for the compressed source sequence.  $\mathbf{G}_C \in \mathbb{B}^{(n_c - m_c) \times n_c}$  is the generator matrix of the channel code. By following the dimension of  $\mathbf{H}_S$ , the following constraint

$$n_c - m_c = m_s \tag{6}$$

needs to be satisfied.

The definition of the rate for the low-density parity-check code pair is given by

$$R_J = \frac{n_s \times (n_c - m_c)}{m_s \times (n_c - n_{punc})}, \tag{7}$$

where  $n_{punc}$  is the number of punctured variable nodes in the channel code.

### 2.2.2. Decoder

The joint decoding algorithm is applied at the receiver to recover the original source sequence  $\mathbf{s}$ . The initial source information  $J^{\text{SC}}$  is denoted by  $\ln((1-p)/p)$ , depending on the source statistics. The initial channel information is denoted by  $J^{\text{CC}} = 2y_i/\sigma^2$ , where  $y_i = (1 - 2c_i) + n_i$  is the additive white Gaussian noise (AWGN).

## 3. Construction of LDPC Code Pairs

### 3.1. Definitions and Theorems

In Figure 1, the cycles of source and channel codes of the JSCC system in the Tanner graph are underlined with bold lines. The minimum cycle length in a given bipartite graph is the so-called girth of the graph. These cycles directly affect the performance of waterfall and error-floor regions for the LDPC code pair. Particularly, short-length cycles are more detrimental, since the information sent out by the message-passing decoder comes back after a small number of hops to the same node that sent it. This results in biases in the decoding algorithm. All in all, the short-length cycles are particularly deleterious to decoding performance, and thus in the design of the code pair, one should intentionally avoid them. Due to the correspondence between the non-zero elements in the parity-check matrix and the edges in Figure 1, it is possible to avoid short cycles during the construction of the LDPC code pair through structural design. Specifically, each column of the parity-check matrix for both the source and channel codes can be represented by a set, with the row indices of the non-zero elements forming this set. Consequently, short cycles are effectively avoided by designing the relationships between these sets.

A graph consists of vertices (or nodes) and the edges (lines or branches) that connect the vertices. In the sequel, we provide multiple definitions and theorems related to graphs.

**Definition 1.** A connected graph,  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , includes vertex-set  $\mathcal{V} = \{v_1, v_2, \dots\}$  with elements called vertices and edge-set  $\mathcal{E} = \{(v_i, v_j)\}$  with pairs of vertices called edges. The end vertices of the edge are the vertices  $v_i$  and  $v_j$  that are connected to an edge  $(v_i, v_j)$ .

**Definition 2.** Let  $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n$  be  $n$  candidate sets,  $\mathcal{T}_i$  contains  $W$  elements from  $\{1, \dots, m\}$ , where  $i \in \{1, \dots, n\}$ . Then, for any positive integer  $t$ ,  $\cup^t$  denotes the union of  $t$  sets selected from the set of all pairwise intersections of the candidate sets, i.e.,  $\mathcal{T}_i \cap \mathcal{T}_j$  for  $1 \leq i, j \leq n$ .

**Theorem 1.** Let  $\mathcal{E}(n)$  denote a set with  $n$  elements and  $\mathcal{T}_i, \mathcal{T}_j$  be subsets of  $\mathcal{E}(n)$ . If  $\forall t \in \{2, 3, \dots, s\}$ ,  $\max\left(|\cup_{\substack{\mathcal{T}_i, \mathcal{T}_j \subseteq \mathcal{E}(n) \\ i \neq j}}^t (\mathcal{T}_i \cap \mathcal{T}_j)|\right) < t$  (the union is not calculated when  $t$  takes the value 2, i.e., calculating the intersection of  $\mathcal{T}_i$  and  $\mathcal{T}_j$  only), then the cycle length of the Tanner graph is larger than  $2s$ , i.e., the girth of the Tanner is larger than or equal to  $2(s + 1)$ .

**Proof.** When  $s = 2$ ,  $\max(|(\mathcal{T}_i \cap \mathcal{T}_j)|) < 2$ , it means that the number of common elements between any two subsets is less than 2. In this case, the cycle length of the Tanner graph exceeds 4. When  $s > 2$ , an increase in its value means that new connection(s) between the check node and the variable node is added to the Tanner graph. For each additional edge in the Tanner graph, the cycle length and the girth will increase by 2. This is due to the fact that the total of the degrees of the vertices for the graph is twice the number of edges, which implies that the number of common elements between the subsets will increase by 1. That is, if there exists a length- $2N$  cycle, then  $t = N$ ,  $\max\left(|\cup_{\substack{\mathcal{T}_i, \mathcal{T}_j \subseteq \mathcal{E}(n) \\ i \neq j}}^t (\mathcal{T}_i \cap \mathcal{T}_j)|\right) = N$ . For example, if there exists a length-6 cycle,  $\exists \mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$  satisfying  $|(\mathcal{T}_1 \cap \mathcal{T}_2) \cup (\mathcal{T}_2 \cap \mathcal{T}_3) \cup (\mathcal{T}_1 \cap \mathcal{T}_3)| = 3$ , which contradicts with Theorem 1 (since according to it,  $|(\mathcal{T}_1 \cap \mathcal{T}_2) \cup (\mathcal{T}_2 \cap \mathcal{T}_3) \cup (\mathcal{T}_1 \cap \mathcal{T}_3)| < 3$ ).  $\square$

**Theorem 2.** According to Theorem 1, the LDPC code pair is constructed for optimizing the performance of the JSCC system. Each column of the parity-check matrix can be represented by a subset (a subset refers to a group of row indices representing the positions of non-zero elements in a column of the parity-check matrix for the source or channel codes, excluding the identity matrix). The maximum number of columns in the LDPC source code is determined via

$$P_s \leq \left\lfloor \frac{m_c(m_c - 1)(m_c^2 - m_c - W_c^2 + W_c)}{W_s W_c^2 (W_s - 1)(W_c - 1)^2} \right\rfloor, \quad (8)$$

where the column weights for the source and channel LDPC codes are denoted as  $W_s$  and  $W_c$ .

**Proof.** The parity-check matrix of the systematic channel LDPC code consists of an identity matrix  $\mathbf{I}$  and a matrix  $\mathbf{P}_c$ . The parity-check matrix of the source code is formed with matrix  $\mathbf{P}_s$ . The construction of matrix  $\mathbf{P}_s$  and matrix  $\mathbf{P}_c$  are based on Theorem 1. If there does not exist any length-4 cycles,  $\forall \mathcal{T}_i, \mathcal{T}_j \subseteq \mathcal{E}(n), \max(|(\mathcal{T}_i \cap \mathcal{T}_j)|) < 2$ , the number of columns of the matrix  $\mathbf{P}$  is less than or equal to  $\frac{m(m-1)}{W(W-1)}$  (the upper bound of row weight of the additional submatrix is  $\frac{m-1}{W-1}$ ), the number of columns of the systematic channel LDPC code is less than or equal to  $\frac{m \cdot (W^2 - W + m - 1)}{W \cdot (W - 1)}$ . (Here, we remove the subscripts for  $m$  and  $W$  to make it more general, since this statement can be applied to both source code and channel code.) The condition that a larger-length ( $>4$ ) cycle does not exist is pre-conditioned on the fact that there does not exist any length-4 circles. Therefore, when a larger-length cycle does not exist, the number of columns of the constructed parity-check matrix has to be reduced to satisfy the stricter set constraint in Theorem 1.  $\square$

### 3.2. Algorithm Description

According to Theorem 1, the corresponding pseudo-code for the proposed construction algorithm of the LDPC code pair is given in Algorithm 1, described as follows:

**Step 1:** Set up the column weight  $W_s$  of the parity-check matrix for the source code and the column weight  $W_c$  of the parity-check matrix for the channel code, and calculate their dimensions according to Theorem 2 and (6) (i.e., first calculate the number of columns  $n_s$  needed in the parity-check matrix of the source code  $\mathbf{H}_s$  based on the corresponding source parameters (e.g., message length and desired code rate). Then, use the desired column weights  $W_s$  and  $W_c$ , along with  $n_s$ , to calculate the number of rows  $m_c$  required for the parity-check matrix of the channel code  $\mathbf{H}_c$ .

**Step 2:** List all the subsets (of set  $\{1, 2, \dots, m_c\}$ ) containing  $W_c$  different elements. The number of such subsets is  $\binom{m_c}{W_c}$ .

**Step 3:** Select the desired subsets (from the previous step) satisfying Theorem 1 for constructing the column vector of the parity-check matrix for the channel code  $\mathbf{H}_c$  (the elements in each subset indicate the row indices of '1' in the associated parity-check matrix), which specifies the constraints that the parity-check matrix must satisfy to ensure good error-correction performance.

**Step 4:** Calculate  $m_s$  according to the relationship in (6) and the constructed channel code from the previous step, and list all the subsets (with  $W_s$  elements) of set  $\{1, 2, \dots, m_s\}$  intended for the construction of the parity-check matrix for the source code  $\mathbf{H}_s$ .

**Step 5:** According to Theorem 1, select the desired subsets for constructing the column vector of the parity-check matrix for source code  $\mathbf{H}_s$ . This matrix must satisfy certain constraints to ensure effective error-correction performance.

**Algorithm 1** Construction of LDPC code pair**Require:**

- $m_c$ : row number of source LDPC code
- $W_s$ : column weight of source LDPC code
- $W_c$ : column weight of channel LDPC code
- $s$ : girth of the constructed source code and channel code.

**Ensure:**

- Initialize:** parity-check matrix for channel code  $\mathbf{H}_C = [\mathbf{I}_{m_c}]$ ,  $n_c = m_c$ .
- 1: **Begin**
  - 2:  $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_i\}$ ;  $\triangleright$  All elements in  $\mathcal{T}$  are a subset of  $\{1, \dots, m_c\}$  containing  $W_c$  elements, i.e.,  $\mathcal{T}_i = \{e_{i,1}, \dots, e_{i,W_c}\}$ .
  - 3:  $\mathcal{T}_1 = \{e_{1,1}, \dots, e_{1,W_c}\}$ ;
  - 4:  $\mathbf{H}_C = [\mathbf{H}_C | \mathbf{h}_{\mathcal{T}_1}]$ ;  $\triangleright$  The elements in  $\mathcal{T}_1$  indicate the row indices of '1' in the associated column vector  $\mathbf{h}_{\mathcal{T}_1}$ .
  - 5:  $n_c = n_c + 1$ ;
  - 6: **for**  $i = 2 : \binom{m_c}{W_c}$  **do**
  - 7:    $t_s = s$ ;  $\triangleright$  If the number of columns in the parity-check matrix is less than  $m_c + s$ , the inter-set constraint conditions will change.
  - 8:   **if**  $n_c - m_c + 1 < s$  **then**
  - 9:      $\triangleright$  The variable  $n_c$  represents the number of columns in the parity check matrix  $H_C$ .
  - 10:      $t_s = n_c - m_c + 1$ ;
  - 11:   **end if**
  - 12:    $l = 0$ ;
  - 13:   **for**  $t = 2 : t_s$  **do**
  - 14:     **if**  $t = 2$  **then**
  - 15:        $I = \max(|(\mathcal{T}_j \cap \mathcal{T}_k)|, \forall \mathcal{T}_j, \mathcal{T}_k \subseteq \mathcal{E}(n), j \neq k), j, k \in [1, n_c - m_c + 1]$ ;  $\triangleright$  Calculate the maximum number of common elements between any two subsets when  $s = 2$ .
  - 16:     **else**
  - 17:        $I = \max(|\cup_{\mathcal{T}_j, \mathcal{T}_k \subseteq \mathcal{E}(n), j \neq k}^t (\mathcal{T}_j \cap \mathcal{T}_k)|), j, k \in [1, n_c - m_c + 1]$ ;  $\triangleright$  Calculate the maximum number of common elements between subsets when  $s > 2$ .
  - 18:     **end if**
  - 19:     **if**  $I < t$  **then**
  - 20:        $l = l + 1$ ;
  - 21:     **else**
  - 22:       **break**;
  - 23:     **end if**
  - 24:   **end for**
  - 25:   **if**  $l = t_s - 1$  **then**
  - 26:      $\triangleright$  Determine whether the subset  $\mathcal{T}_i$  satisfies **Theorem 1**.
  - 27:      $\mathbf{H}_C = [\mathbf{H}_C | \mathbf{h}_{\mathcal{T}_i}]$ ;
  - 28:      $n_c = n_c + 1$ ;
  - 29:   **else**
  - 30:      $\mathbf{H}_C = \mathbf{H}_C$ ;
  - 31:   **end if**
  - 32: **end for**
  - 33: Construct the parity-check matrix of the source code  $\mathbf{H}_S = [\mathbf{H}_{\mathcal{T}_i}]$  similarly according to lines 2 to 32.
  - 34: **End**

**4. Simulation Results**

We evaluate the performance of the proposed JSCC system over AWGN channels in terms of bit error rate (BER) in this section. The simulation results are obtained by considering binary phase-shift keying modulation and employing joint belief propagation iterative decoding. The benchmark code pairs are generated using the progressive edge-growth algorithm with 'copy-and-permute' [23]. In contrast to the benchmark code pair,



the channel code of the proposed code pair does not include punctured VNs. The complexity of proposed code pairs is primarily determined via the column weight of the proposed code, resulting in a complexity of  $O(n^W)$ , where  $W$  is the larger value between the source code column weight  $W_s$  and the channel code column weight  $W_c$ . The maximum number of iterations is set as 50, and that of erroneously decoded frames is set as 100 for all the signal-to-noise ratios (SNRs), denoted as  $E_b/N_0$  in Figures 2 and 3.

Figure 2 presents the simulated BER values for the benchmark code pair (repeat-by-4-jagged-accumulate (R4JA) [24], Accumulate-repeat-by-4-jagged-accumulate (AR4JA) [25]), and the proposed code pair according to Algorithm 1. The source and channel codes of the benchmark code pair are rate-1/5 R4JA and rate-2/3 AR4JA. The girth of the proposed code pair is 8, i.e.,  $s = 3$ . The column weights of the source and channel codes of the proposed code pair are  $W_s = 7$  and  $W_c = 8$ , respectively. The frame length of the proposed code pair is 1693, and the corresponding frame length of the benchmark code pair (R4JA, AR4JA) is 1700. The code rates of the source and channel codes of the proposed code pair are 0.2 and 0.66. The code rate of the proposed code is slightly lower than that of (R4JA, AR4JA). Nevertheless, for  $p = 0.01$ , the proposed code pair brings a 0.5 dB improvement over the benchmark code pair (R4JA, AR4JA) when the BER is at the level of  $10^{-5}$ .

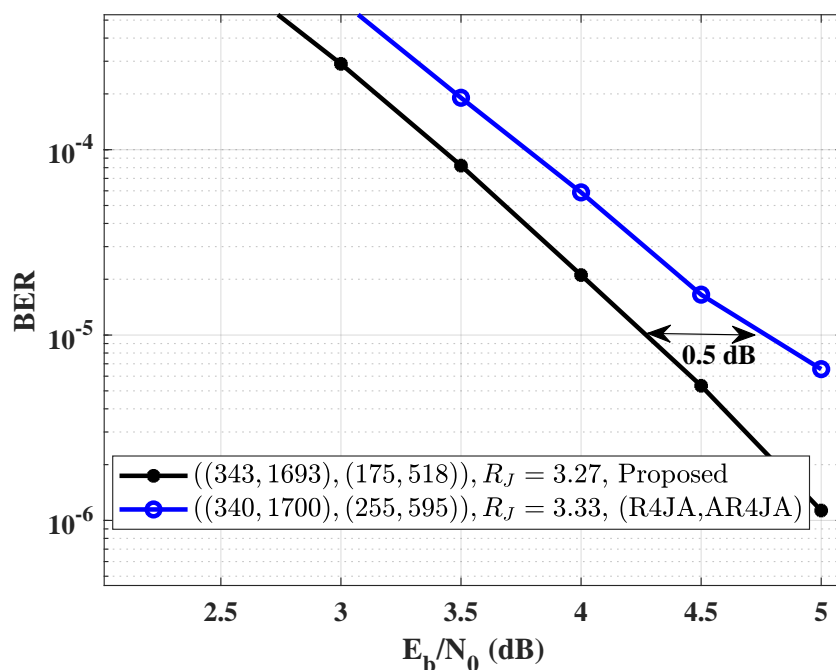


Figure 2. Comparison of BER performance between the proposed code pair and the code pair (R4JA, AR4JA) for  $p = 0.01$ .

Figure 3 shows the simulated BER performance of the code pair (R4JA, AR4JA) and the proposed code pair. The source and channel codes of the benchmark code pair are rate-1/4 R4JA and rate-2/3 AR4JA. The proposed code pair successfully avoids both length-4 and length-6 cycles. The column weights of the source and channel codes are  $W_s = 7$  and  $W_c = 7$ . The code rates of the source and channel codes are 0.25 and 0.67. The girth of the proposed code pair is 8. In this study, the frame length of the proposed code pair is 1088, and the corresponding frame length of the benchmark code pair is 1088. Nevertheless, for  $p = 0.015$ , the proposed code pair brings 0.7 dB gain over the benchmark code pair when the BER is at the level of  $10^{-5}$ .

Figure 4 presents the simulated BER performance of the proposed code pair alongside [15,16]. The code rates of the source and channel codes in [15,16] are 0.25 and 0.5, respectively. The proposed code pair effectively eliminates cycle of length-4 and length-6. The column weights of the source and channel codes are  $W_s = 8$  and  $W_c = 12$ , respectively. The code rates of the source and channel codes are 0.24 and 0.5, respectively. The frame

length of the proposed code pair is 1404, whereas the code pairs in [15,16] correspond to a frame length of 1400. When the probability  $p$  is set to 0.025, the proposed code pair outperforms the code pair in [15] by yielding a gain of 1 dB at the bit error rate of  $10^{-4}$ . Similarly, compared with the code pair in [16], the proposed code pair achieves a gain of 0.6 dB at the same bit error rate.

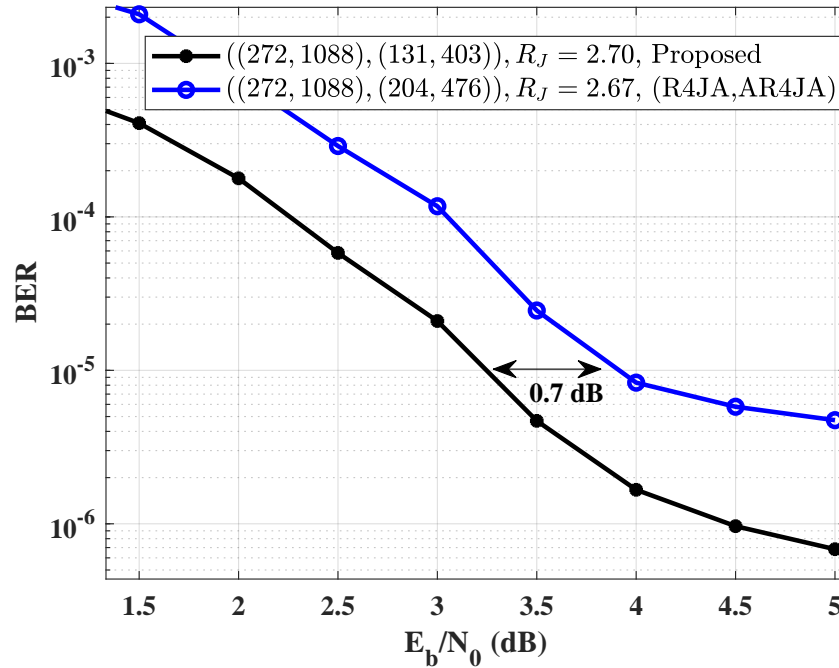


Figure 3. Comparison of BER performance between the proposed code pair and the code pair (R4JA, AR4JA) for  $p = 0.015$ .

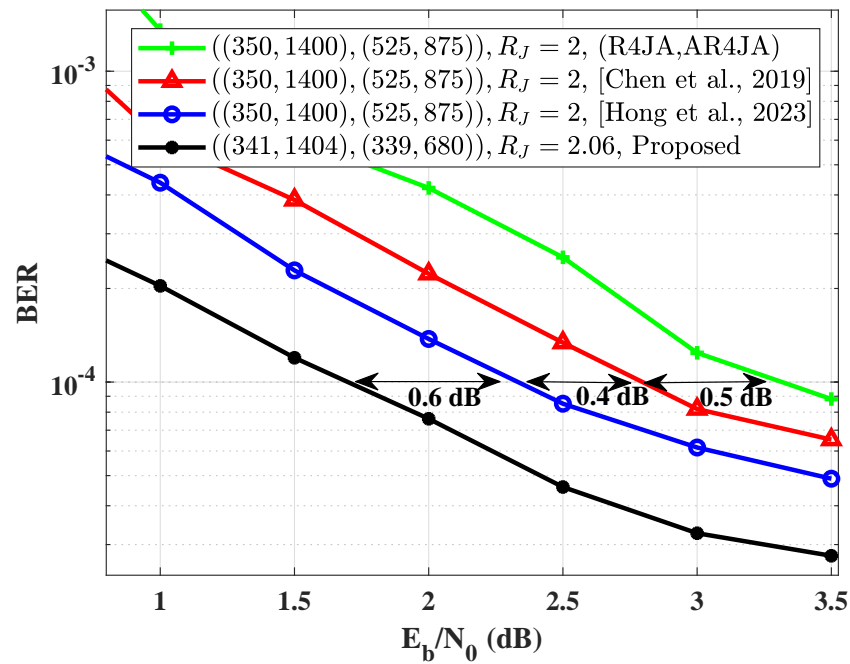


Figure 4. Comparison of BER performance involves the proposed code pair, the code pairs from [15,16], and the code pair (R4JA, AR4JA) for  $p = 0.025$ .



## 5. Conclusions

In this article, an algebraic construction method for LDPC code pairs of JSCC systems has been proposed based on the graph-theoretic approach. The constructed code pairs could avoid short-length cycles by following set constraints. The simulation results have shown that the proposed code pairs achieve significant performance improvement compared with the benchmark code pair (R4JA, AR4JA), [15,16]. The performance enhancement has been observed in both the waterfall region and the error-floor region. In addition, the proposed code pairs customized for the JSCC system have flexible frame lengths (i.e., enhanced adaptability over various channel conditions) unlike the protograph code pairs.

**Author Contributions:** Conceptualization, Y.L., J.H., W.X. and L.W.; methodology, Y.L. and J.H.; software, Y.L.; validation, Y.L.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, J.H., W.X. and L.W.; supervision, J.H., W.X. and L.W.; funding acquisition, W.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant 61871337 and 61671395. This work was supported in part by the Key Laboratory of Southeast Coast Marine Information Intelligent Perception and Application, Ministry of Natural Resources (MNR), No. KFJJ20220201.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [[CrossRef](#)]
2. Guyader, A.; Fabre, E.; Guillemot, C.; Robert, M. Joint source-channel turbo decoding of entropy-coded sources. *IEEE J. Sel. Areas Commun.* **2001**, *19*, 1680–1696. [[CrossRef](#)]
3. Hedayat, A.; Nosratinia, A. Performance analysis and design criteria for finite-alphabet source-channel codes. *IEEE Trans. Commun.* **2004**, *52*, 1872–1879. [[CrossRef](#)]
4. Lakovic, K.; Villasenor, J. Combining variable length codes and turbo codes. In Proceedings of the IEEE 55th Vehicular Technology Conference, Birmingham, AL, USA, 6–9 May 2002; pp. 1719–1723.
5. Lakovic, K.; Villasenor, J. On reversible variable length codes with turbo codes, and iterative source-channel decoding. In Proceedings of the IEEE International Symposium on Information Theory, Lausanne, Switzerland, 30 Jun–5 July 2002; p. 170.
6. Jaspar, X.; Vandendorpe, L. Design and performance analysis of joint source-channel turbo schemes with variable length codes. In Proceedings of the IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2005; pp. 526–530.
7. Poulliat, C.; Declercq, D.; Lamy-Bergot, C.; Fijalkow, I. Analysis and optimization of irregular LDPC codes for joint source-channel decoding. *IEEE Commun. Lett.* **2005**, *9*, 1064–1066. [[CrossRef](#)]
8. Zribi, A.; Pyndiah, R.; Zaibi, S. Low-complexity soft decoding of Huffman codes and iterative joint source channel decoding. *IEEE Trans. Commun.* **2012**, *60*, 1669–1679. [[CrossRef](#)]
9. Fresia, M.; Pérez-Cruz, F.; Poor, H.V. Optimized concatenated LDPC codes for joint source-channel coding. In Proceedings of the IEEE International Symposium on Information Theory, Seoul, Republic of Korea, 28 June–3 July 2009; pp. 2131–2135.
10. Fresia, M.; Pérez-Cruz, F.; Poor, H.V.; Verdu, S. Joint source and channel coding. *IEEE Signal Process. Mag.* **2010**, *27*, 104–113. [[CrossRef](#)]
11. He, J.; Wang, L.; Chen, P. A joint source and channel coding scheme based on simple protograph structured codes. In Proceedings of the International Symposium on Communications and Information Technologies, Gold Coast, Australia, 2–5 October 2012; pp. 65–69.
12. Chen, C.; Wang, L.; Liu, S. The design of protograph LDPC codes as source codes in a JSCC system. *IEEE Commun. Lett.* **2018**, *22*, 672–675. [[CrossRef](#)]
13. Chen, Q.; Wang, L.; Hong, S.; Xiong, Z. Performance improvement of JSCC scheme through redesigning channel code. *IEEE Commun. Lett.* **2016**, *20*, 1088–1091. [[CrossRef](#)]
14. Xu, Z.; Wang, L.; Hong, S.; Chen, G. Design of Code Pair for Protograph LDPC Codes-Based JSCC System with Joint Shuffled Scheduling Decoding Algorithm. *IEEE Commun. Lett.* **2021**, *25*, 3770–3774. [[CrossRef](#)]
15. Chen, Q.; Wang, L.; Hong, S.; Chen, Y. Integrated design of JSCC scheme based on double protograph LDPC codes system. *IEEE Commun. Lett.* **2019**, *23*, 218–221. [[CrossRef](#)]

16. Hong, S.; Ke, J.; Wang, L. Global Design of Double Protograph LDPC Codes for Joint Source-Channel Coding. *IEEE Commun. Lett.* **2023**, *14*, 424–427. [[CrossRef](#)]
17. Golmohammadi, A.; Mitchell, D.G. Concatenated Spatially Coupled LDPC Codes with Sliding Window Decoding for Joint Source-Channel Coding. *IEEE Trans. Commun.* **2021**, *70*, 851–864. [[CrossRef](#)]
18. Djurdjevic, I.; Lin, S.; Abdel-Ghaffar, K. Graph-theoretic construction of low-density parity-check codes. *IEEE Commun. Lett.* **2003**, *7*, 171–173. [[CrossRef](#)]
19. Ammar, B.; Honary, B.; Kou, Y.; Xu, J.; Lin, S. Construction of low-density parity-check codes based on balanced incomplete block designs. *IEEE Trans. Inf. Theory* **2004**, *50*, 1257–1269. [[CrossRef](#)]
20. Kou, Y.; Lin, S.; Fossorier, M.P. Low-density parity-check codes based on finite geometries: A rediscovery and new results. *IEEE Trans. Inf. Theory* **2001**, *47*, 2711–2736. [[CrossRef](#)]
21. Tang, H.; Xu, J.; Kou, Y.; Lin, S.; Abdel-Ghaffar, K. On algebraic construction of Gallager and circulant low density parity-check codes. *IEEE Trans. Inf. Theory* **2004**, *50*, 1269–1279. [[CrossRef](#)]
22. Wu, G.; Lv, Y.; He, J. Design of high-rate LDPC codes based on matroid theory. *IEEE Commun. Lett.* **2019**, *23*, 2146–2149. [[CrossRef](#)]
23. Hu, X.Y.; Eleftheriou, E.; Arnold, D.M. Regular and irregular progressive edge-growth tanner graphs. *IEEE Trans. Inf. Theory* **2005**, *51*, 386–398. [[CrossRef](#)]
24. Divsalar, D.; Dolinar, S.; Jones, C.R.; Andrews, K. Capacity-approaching protograph codes. *IEEE J. Sel. Areas Commun.* **2009**, *2009*, 876–888. [[CrossRef](#)]
25. Divsalar, D.; Dolinar, S.; Jones, C. Construction of protograph LDPC codes with linear minimum distance. In Proceedings of the IEEE International Symposium on Information Theory, Seattle, DC, USA, 9–14 July 2006; pp. 664–668.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.