entropy

MDPI

Article

Design and Analysis of Joint Group Shuffled Scheduling Decoding Algorithm for Double LDPC Codes System

Qiwang Chen, Yanzhao Ren, Lin Zhou *, Chen Chen and Sanya Liu

Xiamen Key Laboratory of Mobile Multimedia Communications, College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China
* Correspondence: linzhou@hqu.edu.cn

Abstract: In this paper, a joint group shuffled scheduling decoding (JGSSD) algorithm for a joint source-channel coding (JSCC) scheme based on double low-density parity-check (D-LDPC) codes is presented. The proposed algorithm considers the D-LDPC coding structure as a whole and applies shuffled scheduling to each group; the grouping relies on the types or the length of the variable nodes (VNs). By comparison, the conventional shuffled scheduling decoding algorithm can be regarded as a special case of this proposed algorithm. A novel joint extrinsic information transfer (JEXIT) algorithm for the D-LDPC codes system with the JGSSD algorithm is proposed, by which the source and channel decoding are calculated with different grouping strategies to analyze the effects of the grouping strategy. Simulation results and comparisons verify the superiority of the JGSSD algorithm, which can adaptively trade off the decoding performance, complexity and latency.

Keywords: joint source-channel coding; shuffled scheduling decoding; belief propagation; EXIT; LDPC code

Citation: Chen, Q.; Ren, Y.; Zhou, L; Chen, C; Liu, S. Design and Analysis of Joint Group Shuffled Scheduling Decoding Algorithm for Double LDPC Codes System. *Entropy* **2023**, *25*, 357. https://doi.org/10.3390/e25020357

Academic Editors: Pingyi Fan, Qi Chen, Suihua Cai and Gangtao Xin

Received: 18 January 2023
Revised: 14 February 2023
Accepted: 14 February 2023
Published: 15 February 2023

1. Introduction

In an integrated communication system [1], Shannon's separation principle indicates that arbitrarily high reliability can be attained for infinite source and channel code block lengths. In a nonasymptotic regime, a joint source-channel coding (JSCC) [2] design can be more attractive, allowing source redundancy and channel information to be exchanged iteratively to improve decoding performance. In addition, the JSCC system can also reduce decoding complexity and transmission delay, which results in its successful application in image and video transmission [3,4].

A JSCC scheme based on double low-density parity-check (D-LDPC) codes was proposed [2]; in this scheme, one LDPC is for source compression and another is for channel error-control. As the LDPC code can be represented by a Tanner graph, a belief propagation (BP) decoding algorithm can be applied. The source and channel coding structures both perform BP decoding, pass decoding information to each other and accomplish information exchange. However, this is a parallel decoding method and needs multiple pieces of calculation hardware working simultaneously. This high decoding complexity is not suitable for low-complexity green communication systems, e.g., Internet of Things (IoT) [5].

1.1. Related Work and Motivation of Shuffled Scheduling Decoding

Recently, the concept of shuffled scheduling decoding [6], a kind of serial decoding, was introduced into the D-LDPC codes system [7]. This can lower the decoding hardware complexity and reduce the number of decoding iterations (an iteration indicates all of the VNs and CNs being updated one time). In order to take full advantage of the linking relationship, a more generalized shuffled scheduling decoding algorithm [8] was proposed. However, this algorithm respectively applied the shuffled scheduling strategy to source decoding and channel decoding, which in fact is a separated decoding method, denoted

as the separated shuffled scheduling decoding (SSSD) algorithm. Compared with the conventional BP decoding algorithm, the SSSD algorithm has a higher decoding latency.

In this paper, a joint shuffled scheduling decoding algorithm will be proposed for the D-LDPC codes system from a global viewpoint. The proposed algorithm is the generalization version of the SSSD algorithm. It considers the Tanner graph of source and channel coding structure as a whole and applies shuffled decoding. Moreover, the joint Tanner graph can be divided into a number of groups to trade off the decoding performance and decoding latency. The grouping relies on the types or on the code-length. We describe the decoding algorithm as a joint group shuffled scheduling decoding (JGSSD) algorithm. Specifically, if the grouping divides the Tanner graph into two parts, i.e., the source and channel parts, then the JGSSD algorithm changes to the SSSD algorithm.

1.2. Related Work on D-LDPC Codes Systems

In recent years, a large amount of research has focused on the investigation of the D-LDPC codes system [9–11]. The main difference between the D-LDPC system and single channel LDPC code systems is the consideration of nonuniform sources and source coding. For the D-LDPC codes system, the source coding may cause the loss of original information and trigger the phenomenon of error floor. In order to improve the performance of the error floor region, a linking matrix is set up between the variable nodes (VNs) of the source code and the check nodes (CNs) of the channel codes [12], and more original information participates in coding. The improvement of the error floor can be evaluated by the source decoding threshold and analyzed by the source protograph EXIT (SPEXIT) algorithm [13]. The linking matrix is further optimized for high-entropy sources [14]. The source LDPC coding matrix can be also optimized to match the source statistic [15] as well as the joint optimization of the source coding matrix and linking matrix [16].

On the other hand, the source redundancy left in the source coding affects the specific structure of the D-LDPC codes. Firstly, the effect of the source statistic is analyzed over the Rayleigh fading channel compared with reception diversity [17]. The channel decoding threshold can be evaluated using the joint protograph EXIT (JPEXIT) algorithm, by which the channel P-LDPC codes [18] and the allocation [19,20] of an important structure, i.e., degree-2 VNs [21], are redesigned. Several works are also performed for the joint component design, including the optimized source and channel pairs [22] and the joint coding matrix [23,24].

In addition, an information shortening strategy was conducted to reduce the effects of the short cycles in the Tanner graph [25]. The D-LDPC codes system was also considered in some nonstandard coding channels [26–28]. Spatially coupled LDPC codes were introduced into the D-LDPC codes system [29]; these can perform sliding window decoding (SWD) for significantly reduced latency and complexity requirements. A proposed SWD algorithm [30] with variable window size was optimized for balancing performance and complexity. The D-LDPC codes system has been applied to image transmission [31,32].

1.3. Main Contribution

The aforementioned D-LDPC codes systems mostly perform BP decoding algorithms. In this paper, a joint decoding viewpoint is introduced, and shuffled scheduling decoding for the D-LDPC codes system is generalized.

The novelty and contributions of this paper can be summarized as follows:

(1) From a global viewpoint, the D-LDPC codes structure is considered as a whole, and a joint shuffled scheduling decoding strategy is introduced to the D-LDPC codes system.
(2) A grouping method for the joint shuffled scheduling decoding strategy, which relies on the types or the length of the VNs, is introduced.
(3) A novel EXIT algorithm to calculate the channel and source decoding thresholds for the general D-LDPC coding structure with the JGSSD algorithm is proposed.
(4) A comparison between the SSSD algorithm and the JGSSD algorithm is conducted, including decoding performance, decoding complexity and decoding latency.

The main differences between the present work and previous work are shown in Table 1.

Table 1. The main differences between the present work (JGSSD) and previous work (IBP and SSSD).

	IBP (e.g., [2])	**SSSD [7,8]**	**JGSSD**
Decoding order	Parallel	Serial	Parallel and Sequential
EXIT algorithm	\	only for algorithm SSSD with the case $\boldsymbol{B}_{l2} = 0$	both for the SSSD and JGSSD with the case $\boldsymbol{B}_{l2} = 0$ and $\boldsymbol{B}_{l2} \neq 0$
Complexity	High	Low	Adaptive
Latency	Low	High	Adaptive
Adaptive judgement	No	No	Yes

1.4. Paper Organization

The remainder of the paper is organized as follows. In Section 2, we describe the preliminaries of the D-LDPC codes. The joint shuffled scheduling decoding algorithm with grouping strategy is proposed in Section 3. An EXIT algorithm for analyzing the D-LDPC codes system with JGSSD algorithm is presented in Section 4. A simulation and comparisons are conducted in Section 5. Finally, Section 6 draws the conclusions of this paper.

2. Preliminaries of D-LDPC Codes Systems

2.1. The D-LDPC Coding Structure

An LDPC code can be represented by a protograph, a small protomatrix $\boldsymbol{B} = [b_{ij}]$, where b_{ij} indicates the number of edges connecting a VN v_j to a CN c_i. Then, a large parity-check matrix can be obtained by a "copy-and-permute" operation, such as the progressive edge growth (PEG) algorithm [33] with a lifting factor.

A D-LDPC code can be represented by a joint protograph $\boldsymbol{B}_J$ as follows:

$$\boldsymbol{B}_J = \begin{bmatrix} \boldsymbol{B}_s & \boldsymbol{B}_{l1} \\ \boldsymbol{B}_{l2} & \boldsymbol{B}_c \end{bmatrix}, \tag{1}$$

where $\boldsymbol{B}_s$ is the source coding protomatrix of size $m_s \times n_s$, $\boldsymbol{B}_c$ is the channel coding protomatrix of size $m_c \times n_c$, $\boldsymbol{B}_{l1} = [\boldsymbol{I}\ \boldsymbol{0}]$ ($\boldsymbol{I}$ is an identity matrix) is the source-check-channel-variable linking protomatrix of size $m_s \times n_c$ and $\boldsymbol{B}_{l2}$ is the source-variable-channel-check linking protomatrix of size $m_c \times n_s$. Then, a joint parity-check matrix $\boldsymbol{H}_J$ can be derived:

$$\boldsymbol{H}_J = \begin{bmatrix} \boldsymbol{H}_s & \boldsymbol{H}_{l1} \\ \boldsymbol{H}_{l2} & \boldsymbol{H}_c \end{bmatrix}, \tag{2}$$

where $\boldsymbol{H}_s$ is the source coding matrix of size $M_s \times N_s$, $\boldsymbol{H}_c$ is the channel coding matrix of size $M_c \times N_c$, $\boldsymbol{H}_{l1} = [\boldsymbol{I}\ \boldsymbol{0}]$ is the source-check-channel-variable linking matrix of size $M_s \times N_c$ and $\boldsymbol{H}_{l2}$ is the source-variable-channel-check linking matrix of size $M_c \times N_s$. The overall code rate of the D-LDPC codes is given by $R_{overall} = \frac{N_s}{M_s} \times \frac{N_c - M_c}{N_c}$.

2.2. Transmission System Model

Assume that original source bits $\boldsymbol{s} \in \{0,1\}^{(1 \times N_s)}$ are generated from a binary independent and identically distributed (i.i.d) Bernoulli source, where the probability of "1" is η. The encoding procedures with a nonzero $\boldsymbol{H}_{l2}$ are given as follows. Firstly, the compressed source bits $\boldsymbol{c}$ can be obtained by

$$\boldsymbol{c} = \boldsymbol{s}(\boldsymbol{H}_s)^T, \tag{3}$$

where $(\cdot)^T$ represents the matrix transposition in math. Then, a codeword $\boldsymbol{u}$ can be obtained by

$$\boldsymbol{u} = [\boldsymbol{s}, \boldsymbol{c}]\boldsymbol{G}_p, \tag{4}$$

where $\boldsymbol{G}_p$ is a systematic generator matrix obtained from $[\boldsymbol{H}_{l2}\ \boldsymbol{H}_c]$. Thus, the $\boldsymbol{u}$ consists of three parts, i.e., $\boldsymbol{u} = [\boldsymbol{s}, \boldsymbol{c}, \boldsymbol{p}]$, where $\boldsymbol{p}$ is the parity bits. Finally, the channel codeword $\boldsymbol{d} = [\boldsymbol{c}, \boldsymbol{p}]$ is sent to the channel, after puncturing if punctured LDPC code is considered. If $\boldsymbol{H}_{l2} = 0$, the encoding procedure can be simplified by $\boldsymbol{d} = [\boldsymbol{c}, \boldsymbol{p}] = \boldsymbol{s}(\boldsymbol{H}_s)^T\boldsymbol{G}_c$, where $\boldsymbol{G}_c$ is obtained from $\boldsymbol{H}_c$.

In the decoding process, a binary-phase-shift keying (BPSK) and AWGN channel model are assumed and the log-likelihood-ratio (LLR) values of all VNs are first initialized. Then, as shown in Figure 1, the iterative BP (IBP) algorithm is performed as follows:

(1) Update all the C2V messages for each of the M_c CNs in the channel part;
(2) Update all the V2C messages for each of the N_s VNs in the source part;
(3) Update all the C2V messages for each of the M_s CNs in the source part;
(4) Update all the V2C messages for each of the N_c VNs in the channel part;
(5) The source part and channel part exchange decoding information through $\boldsymbol{H}_{l1}$ and $\boldsymbol{H}_{l2}$ (i.e., the dashed blue and red lines in Figure 1);
(6) Estimate the codeword $\hat{\boldsymbol{u}}$ based on the posterior LLRs at the VNs;
(7) Repeat Steps (1) to (6), unless (i) the estimated codeword $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{d}}$ satisfy $\hat{\boldsymbol{s}}(\boldsymbol{H}_s)^T = 0$ and $\hat{\boldsymbol{d}}(\boldsymbol{H}_c)^T = 0$ (ii) the maximum iteration number K is reached.

For the SSSD algorithm in [7,8], the updates for C2V in the channel part and source part, i.e., Step (1) and (3), are performed using shuffled scheduling. For more details about the decoding procedure, the reader can refer to [8].

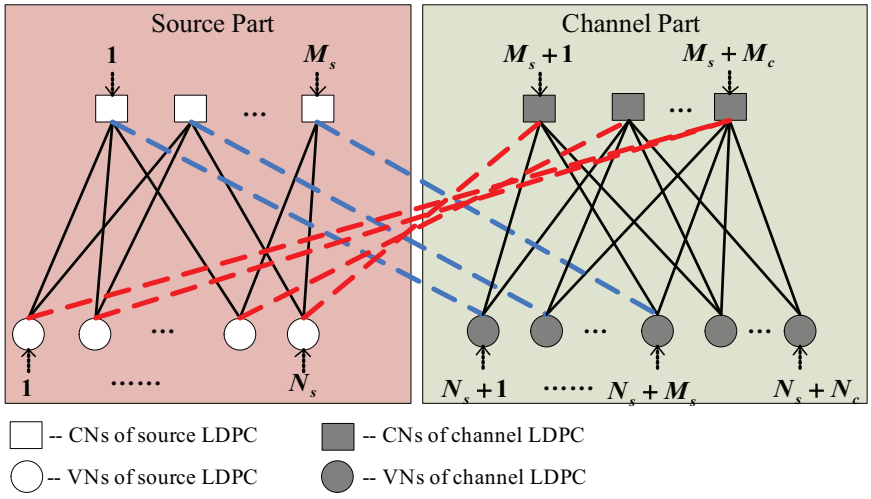

Figure 1. The joint Tanner graph of the D-LDPC codes system.

3. Joint Group Shuffled Scheduling Decoding Algorithm

It can be observed that the BP and SSSD algorithms mentioned above are both types of iterative decoding method between the source decoder and channel decoder. The shuffled scheduling decoding is only respectively applied in source decoding and channel decoding, and not applied to the update of $\boldsymbol{H}_{l1}$ and $\boldsymbol{H}_{l2}$. Thus, the SSSD algorithm is in fact a separated decoding method.

Here, considering that the vector $\boldsymbol{u}$ satisfies

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{H}_J)^T &= [\boldsymbol{s},\boldsymbol{c},\boldsymbol{p}](\boldsymbol{H}_J)^T \\ &= [\boldsymbol{s},\boldsymbol{c},\boldsymbol{p}]\left[\begin{array}{c:c} \boldsymbol{H}_s & \boldsymbol{I}\ \boldsymbol{0} \\ \hdashline \boldsymbol{H}_{l2} & \boldsymbol{H}_c \end{array}\right]^T \\ &= [\boldsymbol{s},\boldsymbol{c},\boldsymbol{p}]\left[\begin{array}{c:c} (\boldsymbol{H}_s)^T & \boldsymbol{0} \\ \hdashline (\boldsymbol{I})^T & \\ \boldsymbol{0} & (\boldsymbol{H}_c)^T \end{array}\right] \\ &= \left[[\boldsymbol{s},\boldsymbol{c}]\left[\begin{array}{c} \boldsymbol{H}_s \\ \boldsymbol{I} \end{array}\right]^T, [\boldsymbol{c},\boldsymbol{p}](\boldsymbol{H}_c)^T\right], \end{aligned} \tag{5}$$

where $[\boldsymbol{s},\boldsymbol{c}]\left[\begin{array}{c} \boldsymbol{H}_s \\ \boldsymbol{I} \end{array}\right]^T = \boldsymbol{s}(\boldsymbol{H}_s)^T + \boldsymbol{c} = \boldsymbol{c} + \boldsymbol{c} = \boldsymbol{0}$ and $[\boldsymbol{c},\boldsymbol{p}](\boldsymbol{H}_c)^T = 0$, so that $\boldsymbol{u}(\boldsymbol{H}_J)^T = \boldsymbol{0}$.

Thus, the combined Tanner graph of the source part and the channel part can be considered a joint Tanner graph. We apply the BP decoding to the D-LDPC codes system, and this is denoted as a joint BP decoding (JBP) algorithm. For ease of description of the JBP algorithm, several types of LLRs are defined, and k-th iteration is assumed.

- z_n^s represents the LLR of the n-th bit of original source $\boldsymbol{s}$.
- z_n^d represents the LLR of the n-th bit of codeword $\boldsymbol{d}$.
- ε_{mn}^k represents the LLR from the m-th CN to the n-th VN at k-th iteration.
- ϕ_{mn}^k represents the LLR from the n-th VN to the m-th CN at k-th iteration.
- Φ_n^k represents the LLRs of the n-th bit at k-th iteration.

Based on the above definitions, the decoding procedure is described as follows.

Initialization: The initial LLR of VNs can be calculated by

$$z_n^s = \ln((1-\eta)/\eta), (n = 1, 2, \cdots, N_s) \tag{6}$$

and

$$z_n^d = (2r_n)/\sigma^2, (n = 1, 2, \cdots, N_c), \tag{7}$$

where r_n is the n-th received signal and σ^2 is the noise variance, and the LLRs are 0 for the punctured bits. Furthermore, the σ^2 can be calculated by $\sigma^2 = 1/(2 \times R_{overall} \times (E_s/N_0))$, where E_s is the average transmitted energy per source information bit, and N_0 is the noise power spectral density.

Step 1: Update C2V messages, for $1 \le n \le N_s + N_c$ and $1 \le m \le M_s + M_c$,

$$\varepsilon_{mn}^k = 2\tanh^{-1}\left(\prod_{n' \ne n} \tanh\left(\frac{\phi_{mn'}^{k-1}}{2}\right)\right). \tag{8}$$

Step 2: Update V2C messages, for $1 \le n \le N_s + N_c$ and $1 \le m \le M_s + M_c$,

$$\phi_{mn}^k = \begin{cases} z_n^s + \sum\limits_{m' \in \Theta(n)\backslash m} \varepsilon_{m'n}^k, & \text{if} \quad 1 \le n \le N_s \\ z_n^d + \sum\limits_{m' \in \Theta(n)\backslash m} \varepsilon_{m'n}^k, & \text{if} \quad N_s + 1 \le n \le N_s + N_c \end{cases} \tag{9}$$

and

$$\Phi_n^k = \begin{cases} z_n^s + \sum\limits_{m \in \Theta(n)} \varepsilon_{mn}^k, & \text{if} \quad 1 \le n \le N_s \\ z_n^d + \sum\limits_{m \in \Theta(n)} \varepsilon_{mn}^k, & \text{if} \quad N_s + 1 \le n \le N_s + N_c \end{cases}. \tag{10}$$

where $\Theta(n)$ denotes all CNs connected to the n-th VN. For $1 \leq n \leq N_s + N_c$, if $\Phi_n^k \geq 0$, $\hat{u}_n = 0$; otherwise, $\hat{u}_n = 1$, where $\hat{\boldsymbol{u}} = [\hat{u}_n](n = 1, \cdots, N_s + N_c)$ is the estimated bit codeword.

Step 3: Stopping condition: if $\hat{\boldsymbol{u}} \cdot \boldsymbol{H}_J = 0$ or $k = K_{max}$, the iteration will stop; otherwise, set $k = k + 1$ and go to Step 1, where K_{max} is the maximum iteration of decoding.

For the joint shuffled scheduling BP algorithm, the initialization and stopping conditions remain the same as in the JBP algorithm. The only difference between the two algorithms lies in the updating procedure. For the updated C2V message, certain $\phi_{mn'}^k$ have been updated in Step 2 and can be used instead of $\phi_{mn'}^{k-1}$ in Step 1 to calculate the remaining values ε_{mn}^k. Thus, the updated C2V message can be modified as follows:

$$\varepsilon_{mn}^k = 2\tanh^{-1}\left(\prod_{n'<n}\tanh\left(\frac{\phi_{mn'}^k}{2}\right) \times \prod_{n'>n}\tanh\left(\frac{\phi_{mn'}^{k-1}}{2}\right)\right). \tag{11}$$

However, it is observed that one iteration of the standard JBP algorithm can be fully processed in parallel, while that of a shuffled JBP algorithm becomes totally serial, and this will bring about a large decoding latency. To decrease decoding latency and keep the parallelism advantages of the standard JBP, the concept of grouping is introduced, and the decoding algorithm is developed into a joint group shuffled scheduling decoding (JGSSD) algorithm. Before performing BP decoding, the decoding information is first divided into a certain number of groups. The updating of information in each group is processed in parallel, but the processing of groups remains serial. In detail, the VNs are divided into a number of groups according to certain criteria, i.e.,

$$\boldsymbol{V} = \{\boldsymbol{V}^1, \boldsymbol{V}^2, \cdots, \boldsymbol{V}^G\}, \tag{12}$$

where $\boldsymbol{V}^g = [V_i^g](g = 1, 2, \cdots, G^H, i = 1, 2, \cdots, n_g)$, G^H is the number of groups and n_g is the size of $\boldsymbol{V}^g$. Thus, the updated C2V message can be modified as follows:

For $n \in \boldsymbol{V}^g$, i.e., $N_{g-1} < n < N_{g-1} + n_g$ and $m \in \Theta(n)$,

$$\varepsilon_{mn}^k = 2\tanh^{-1}\left(\prod_{\substack{n' \in \Theta(m)\backslash n \\ n' \leqslant N_{g-1}}}\tanh\left(\frac{\phi_{mn'}^k}{2}\right) \times \prod_{\substack{n' \in \Theta(m)\backslash n \\ n' > N_{g-1}}}\tanh\left(\frac{\phi_{mn'}^{k-1}}{2}\right)\right), \tag{13}$$

where $\Theta(m)$ denotes all VNs connected to the m-th CN, $\Theta(n)$ denotes all CNs connected to the n-th VN, $\Theta(m) \setminus n$ denotes all VNs in $\Theta(m)$ excluding the n-th VN, N_{g-1} is the size of the sum of the former sets $\{\boldsymbol{V}^1, \boldsymbol{V}^2, \cdots, \boldsymbol{V}^{g-1}\}$.

For the JGSSD algorithm, we can group the $\boldsymbol{V}$ based on code length. For example, we assumed that the $N_s + N_c$ VNs are divided into G^H groups on average, and that each group contains $(N_s + N_c)/G^H$ VNs (assuming $(N_s + N_c)$ mod $G^H = 0$ for simplicity). We can also group the $\boldsymbol{V}$ according to the types of VNs. For example, $\boldsymbol{V} = \{\boldsymbol{V}^1, \boldsymbol{V}^2\}$ and $\boldsymbol{V}^1 = \{\text{source VNs}\}$ of length N_s and $\boldsymbol{V}^2 = \{\text{channel VNs}\}$ of length N_c. Now, the JGSSD becomes the IBP algorithm. If the $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$ are further divided into N_s and N_c groups, respectively, this grouping strategy makes the algorithm change to be the SSSD version.

4. Analysis of the D-LDPC Codes System with JGSSD Algorithm

4.1. Joint Shuffled Extrinsic Information Algorithm

EXIT analysis can reflect the ultimate performance of a D-LDPC codes system by calculating the decoding threshold of its corresponding protograph. Although an EXIT algorithm for analyzing the decoding threshold of the D-LDPC codes system with shuffled scheduling has been proposed in [34], it only aimed at the $\boldsymbol{B}_J$ with $\boldsymbol{B}_{J2} = 0$. In addition, the EXIT algorithm is comprised of source and channel parts, just like the decoding procedure. Thus, it is not suitable for the D-LDPC codes system with general structure and JGSSD

algorithm. In this section, a joint shuffled extrinsic information transfer (JSEXIT) algorithm is proposed.

Firstly, five types of mutual information (MI) are defined as:

- $I^{Ev}_{ij,(k)}$: the extrinsic MI from j-th VN to i-th CN at k-th iteration;
- $I^{Ec}_{ij,(k)}$: the extrinsic MI from i-th CN to j-th VN at k-th iteration;
- $I^{Av}_{ij,(k)}$: the a priori MI from j-th VN to i-th CN at k-th iteration;
- $I^{Ac}_{ij,(k)}$: the a priori MI from i-th CN to j-th VN at k-th iteration;
- $I^{APP}_{j,(k)}$: the MI between a posteriori LLR evaluated by j-th VN and the corresponding source bit s_j at k-th iteration.

In addition, an indicator function is defined as follows:

$$\Omega(b_{ij}) = \begin{cases} 1 & \text{if } b_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

and if a VN is punctured, its initial LLR value is 0. Moreover, $\mathbb{J}(\sigma_{ch})$ represents the MI between a binary bit and its corresponding LLR value $L_{ch} \sim \mathbb{N}(\sigma^2_{ch}/2, \sigma^2_{ch})$ $\mathbb{N}(\theta, \sigma^2)$ represent the Gaussian distribution with expectation θ and variance σ^2, given by [2]

$$\mathbb{J}(\sigma_{ch}) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(\xi - \sigma^2_{ch}/2)^2/2\sigma^2_{ch}}}{\sqrt{2\pi\sigma^2_{ch}}} \cdot \log_2(1 + e^{-\xi}) d\xi. \tag{15}$$

Then, the VNs of the joint protograph are divided into a number of groups according to certain criteria, i.e.,

$$\boldsymbol{v} = \{\boldsymbol{v}^1, \boldsymbol{v}^2, \cdots, \boldsymbol{v}^{G^B}\}, \tag{16}$$

where $\boldsymbol{v}^g = [v^g_i](g = 1, 2, \cdots, G^B, i = 1, 2, \cdots, t_g)$, G^B is the number of groups and t_g is the size of $\boldsymbol{v}^g$.

Finally, the proposed JSEXIT algorithm for the D-LDPC codes system over AWGN channel is described as follows.

Step 1: The MI update from VNs to CNs

For $1 \leq j \leq n_s$ and $1 \leq i \leq m_s + m_c$

$$I^{Ev}_{ij,(k)} = \Omega(b_{ij}) \mathbb{J}_{BSC}\left(\sum_{s \neq i} b_{is} [\mathbb{J}^{-1}(I^{Av}_{sj,(k)})]^2 + (b_{ij} - 1)[\mathbb{J}^{-1}(I^{Av}_{ij,(k)})]^2, \eta\right). \tag{17}$$

The function $\mathbb{J}_{BSC}$ is defined as [2]

$$\mathbb{J}_{BSC}(\mu, \eta) = (1 - \eta)\mathbb{I}(V; \chi^{(1-\eta)}) + \eta \mathbb{I}(V; \chi^{\eta}),$$

where $\mathbb{I}(V; \chi)$ is an MI calculation between the VN of the source and the distribution χ.

Further, $\chi^{(1-\eta)} \sim \mathbb{N}(\mu + Z^{sc}_v, 2\mu)$ and $\chi^{\eta} \sim \mathbb{N}(\mu - Z^{sc}_v, 2\mu)$ with $Z^{sc}_v = \ln((1 - \eta)/\eta)$.

For $(n_s + 1) \leq j \leq (n_s + n_c)$ and $1 \leq i \leq (m_s + m_c)$

$$I^{Ev}_{ij,(k)} = \Omega(b_{ij}) \mathbb{J}\left(\sqrt{\sum_{s \neq i} b_{sj} [\mathbb{J}^{-1}(I^{Av}_{sj,(k)})]^2 + (b_{ij} - 1)[\mathbb{J}^{-1}(I^{Av}_{ij,(k)})]^2 + \sigma^2_{ch,j}}\right). \tag{18}$$

For $1 \leq j \leq (n_s + n_c)$ and $1 \leq i \leq (m_s + m_c)$

$$I^{Ac}_{ij,(k)} = I^{Ev}_{ij,(k)}. \tag{19}$$

Step 2: The MI update from CNs to VNs

For $1 \leq g \leq G^B$, $T_{g-1} \leq j \leq T_{g-1} + t_g$ and $1 \leq i \leq (m_s + m_c)$

$$I_{ij,(k)}^{Ec} = \Omega(b_{ij})\left(1 - \mathbb{J}\left(\sqrt{\alpha_{(k)} + \alpha_{(k-1)}}\right)\right) \tag{20}$$

where

$$\alpha_{(k)} = (\sum_{s \in \theta(i) \setminus j,\ s \leqslant T_{g-1}} b_{is}[\mathbb{J}^{-1}(1 - I_{is,(k)}^{Ac})]^2), \tag{21}$$

$$\alpha_{(k-1)} = (\sum_{s \in \theta(i) \setminus j,\ s > T_{g-1}} b_{is}[\mathbb{J}^{-1}(1 - I_{is,(k-1)}^{Ac})]^2) + (b_{ij} - 1)[\mathbb{J}^{-1}(1 - I_{ij,(k-1)}^{Ac})]^2, \tag{22}$$

and set $I_{ij,(k)}^{Av} = I_{ij,(k)}^{Ec}$. Further, $\theta(i) \setminus j$ denotes all VNs connected to the i-th CN excluding the j-th VN, T_{g-1} is the size of the sum of the former sets $\{\boldsymbol{v}^1, \boldsymbol{v}^2, \cdots, \boldsymbol{v}^{g-1}\}$. It is noted that in the calculation of $I_{ij,(k)}^{Ec}$, partial $I_{is,(k)}^{Ac}$ has been updated to replace the $I_{is,(k-1)}^{Ac}$, which is reflected in the different calculations of $\alpha_{(k-1)}$ and $\alpha_{(k)}$.

Step 3: The APP-LLR MI evaluation

For $1 \leq j \leq n_s$ and $1 \leq i \leq m_s$

$$I_{j,(k)}^{APP} = \mathbb{J}_{BSC}(\mu(j), \eta) \tag{23}$$

where $\mu(j) = \sum_i b_{ij}[\mathbb{J}^{-1}(I_{ij}^{Av})]^2$.

The procedure of Steps 1 to 3 is performed iteratively until $I_j^{APP} = 1$ or the maximum iteration is reached.

Remarks: If the maximum iteration is set to a large value, like the conventional EXIT algorithm, then the JSEXIT algorithm cannot reflect its advantage of shuffled scheduling, as with the larger iteration number in shuffled scheduling decoding, which has similar performance to that of the standard BP algorithm. We set the maximum iteration to 20 here for this reason. Therefore, the decoding threshold has a gap compared with that of the conventional EXIT algorithm, but it can provide comparable results.

4.2. Decoding Threshold Calculation

The MI I_j^{APP} can be viewed as a function of independent variables $\boldsymbol{B}_J$, η, σ_{ch} and G^B, i.e.,

$$I_j^{APP} = \Upsilon(\boldsymbol{B}_J, \eta, \sigma_{ch}, G^B), \tag{24}$$

where σ_{ch} can be calculated from E_s/N_0. The channel decoding threshold $(E_s/N_0)_{th}$ indicates the performance of the water-fall region, which is the minimum value to make all I_j^{APP} 1 for a given η. The η_{th} indicates the performance of the error floor level, which is the maximum value to make all I_j^{APP} 1 when $E_s/N_0 \to \infty$. The $(E_s/N_0)_{th}$ and η_{th} will also be calculated when a different G^B is set. Without loss of generality, two examples using regular LDPC codes as source and channel code are presented as follows, where $\boldsymbol{B}_{J1}$ is with $\boldsymbol{B}_{l2} = 0$ and another $\boldsymbol{B}_{J2}$ is with $\boldsymbol{B}_{l2} \neq 0$. The regular source and channel protographs with degree-3 VNs are given by

$$\boldsymbol{B}_s^{reg} = \boldsymbol{B}_c^{reg} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{25}$$

The $\boldsymbol{B}_{l1}$ is given by

$$\boldsymbol{B}_{l1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{26}$$

The nonzero $\boldsymbol{B}_{l2}^{nz}$ is represented by

$$\boldsymbol{B}_{l2}^{nz} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{27}$$

Thus, the $\boldsymbol{B}_{J1}$ and $\boldsymbol{B}_{J2}$ are respectively represented by

$$\boldsymbol{B}_{J1} = \begin{bmatrix} \boldsymbol{B}_s^{reg} & \boldsymbol{B}_{l1} \\ \boldsymbol{0} & \boldsymbol{B}_c^{reg} \end{bmatrix}, \quad \boldsymbol{B}_{J2} = \begin{bmatrix} \boldsymbol{B}_s^{reg} & \boldsymbol{B}_{l1} \\ \boldsymbol{B}_{l2}^{nz} & \boldsymbol{B}_c^{reg} \end{bmatrix}. \tag{28}$$

The source decoding thresholds η_{th} and channel decoding thresholds $(E_s/N_0)_{th}$ for different grouping strategies are calculated and shown in Tables 2 and 3. It can be seen that 1) the JSEXIT algorithm can calculate the decoding threshold regardless of the $\boldsymbol{B}_{l2} = 0$ or $\boldsymbol{B}_{l2} \neq 0$; 2) with the increase of G^B, the source coding threshold becomes large and the channel threshold becomes small. For example, with $\boldsymbol{B}_{J2}$ at source statistic $\eta = 0.11$, the case of $G^B = 16$ outperforms the cases of $G^B = 8$, $G^B = 4$, $G^B = 2$ and $G^B = 1$ by 0.12 dB, 0.35 dB, 1.17 dB and 1.27 dB, respectively.

Table 2. Source decoding thresholds η_{th} and channel decoding thresholds for $(E_s/N_0)_{th}$ at $\eta = 0.04$ and 0.07 for $\boldsymbol{B}_{J1}$ with different grouping strategies.

$\boldsymbol{B}_{J1}$	η_{th}	$(E_s/N_0)_{th}$	
		0.04	**0.07**
$G^B = 1$	0.072	−3.13 dB	−1.39 dB
$G^B = 2$	0.073	−3.27 dB	−1.41 dB
$G^B = 4$	0.077	−3.46 dB	−2.00 dB
$G^B = 8$	0.078	−3.54 dB	−2.19 dB
$G^B = 16$	0.079	−3.58 dB	−2.25 dB

Table 3. Source decoding thresholds η_{th} and channel decoding thresholds for $(E_s/N_0)_{th}$ at $\eta = 0.07$ and 0.11 for $\boldsymbol{B}_{J2}$ with different grouping strategies.

$\boldsymbol{B}_{J2}$	η_{th}	$(E_s/N_0)_{th}$	
		0.07	**0.11**
$G^B = 1$	0.118	−1.67 dB	0.65 dB
$G^B = 2$	0.119	−1.77 dB	0.55 dB
$G^B = 4$	0.125	−1.98 dB	−0.27 dB
$G^B = 8$	0.128	−2.07 dB	−0.50 dB
$G^B = 16$	0.129	−2.10 dB	−0.62 dB

5. Simulation and Comparison

In this section, we will illustrate the advantages of the JGSSD algorithm through Monte Carlo simulations and analyses of iteration number and decoding latency. For all

simulations, the length of the source sequence is 3200, so the lifting factor of the PEG algorithm for $\boldsymbol{B}_{J1}$ and $\boldsymbol{B}_{J2}$ is 800. The maximum number of decoding iterations K is set as 30.

5.1. BER Performance

The BER performance of $\boldsymbol{B}_{J1}$ and $\boldsymbol{B}_{J2}$ with different grouping strategies are shown in Figures 2–5. Firstly, it should be noted that the JGSSD algorithm is suitable for both the cases of $\boldsymbol{B}_{l2} = \mathbf{0}$ and $\boldsymbol{B}_{l2} \neq \mathbf{0}$. Secondly, the BER performance in the water-fall region becomes better as the G^H increases; this is in line with the EXIT analysis. It should be noted that the case of $G^H = 3200$ is equivalent to the SSSD algorithm in [7,8]. For the case of $\boldsymbol{B}_{J1}$ at source statistic at $\eta = 0.04$, the case of $G^H = 400$ has a coding gain of 0.2 dB compared with the case of $G^H = 1$ and has no significant difference compared with $G^H = 3200$ and $G^H = 6400$ at the BER of 2×10^{-7}. Other comparisons can be seen in Table 4. Thirdly, the error floor level also becomes lower as the G^H increases, and this is in line with the EXIT analysis. For the case of $\boldsymbol{B}_{J2}$ at source statistic at $\eta = 0.11$, the case of $G^H = 8$ is better than that of $G^H = 1$, but worse than that of $G^H = 6400$. It should be explained that the case $G^H = 400$ and $G^H = 3200$ have almost the same error floor level as that of $G^H = 6400$.

Table 4. E_b/N_0 Gain at BER = 1×10^{-6} and error floor levels for different grouping strategies.

Grouping Strategy	E_b/N_0 Gain at BER = 1×10^{-6}		Error Floor Level	
	$\eta = 0.04, \boldsymbol{B}_{J1}$	$\eta = 0.07, \boldsymbol{B}_{J2}$	$\eta = 0.07, \boldsymbol{B}_{J1}$	$\eta = 0.11, \boldsymbol{B}_{J2}$
$G^H = 1$	0	0	7×10^{-4}	2×10^{-4}
$G^H = 2$	0.05 dB	0.01 dB	-	-
$G^H = 8$	0.08 dB	0.12 dB	5×10^{-4}	1×10^{-4}
$G^H = 400$	0.18 dB	0.18 dB	-	-
$G^H = 3200$	0.19 dB	0.22 dB	-	-
$G^H = 6400$	0.19 dB	0.22 dB	3×10^{-4}	9×10^{-5}

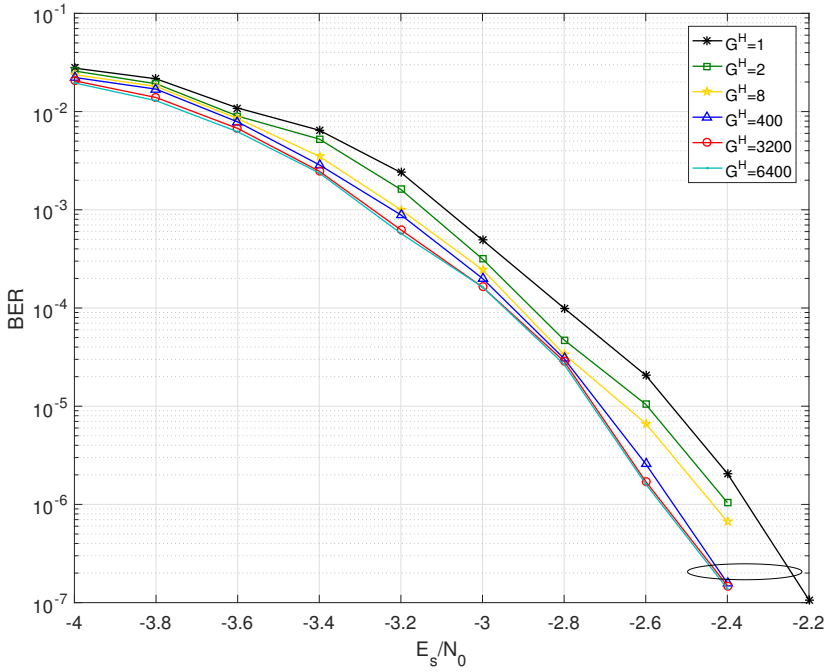

Figure 2. BER performance of $\boldsymbol{B}_{J1}$ with $G^H = 1, 2, 8, 400, 3200, 6400$ at source statistic $\eta = 0.04$.

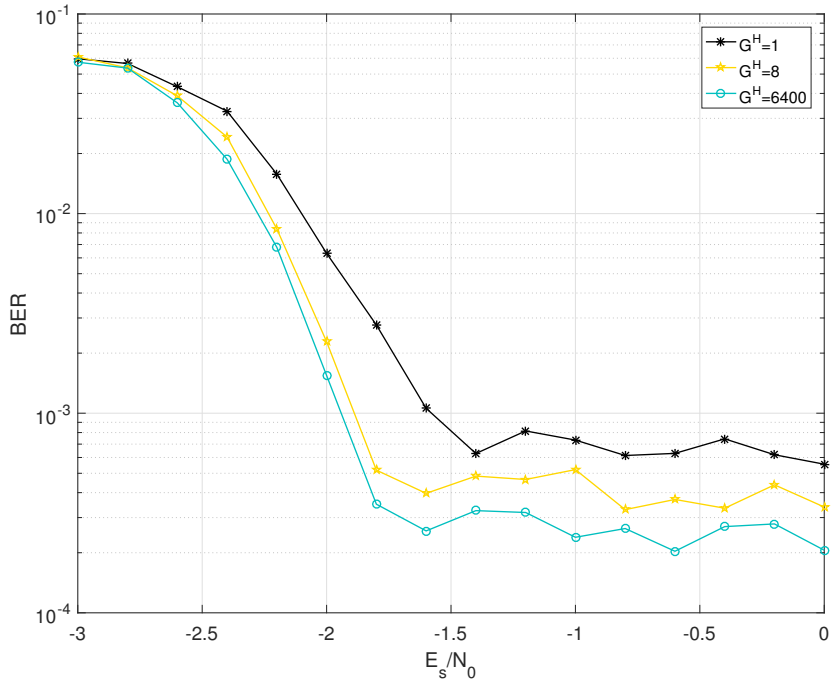

Figure 3. BER performance of $\boldsymbol{B}_{J1}$ with $G^H = 1, 8, 6400$ at source statistic $\eta = 0.07$.

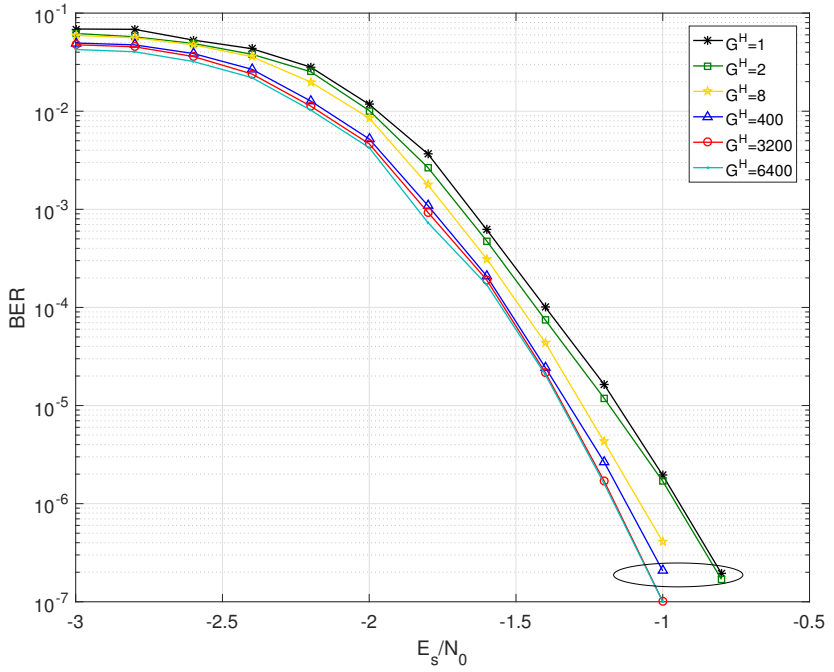

Figure 4. BER performance of $\boldsymbol{B}_{J2}$ with $G^H = 1, 2, 8, 400, 3200, 6400$ at source statistic $\eta = 0.07$.

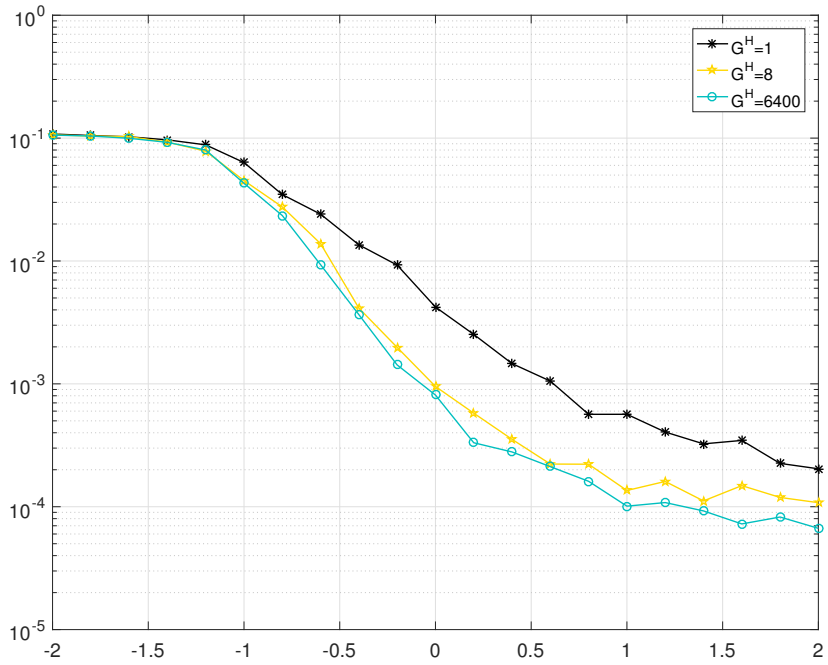

Figure 5. BER performance of $\boldsymbol{B}_{J2}$ with $G^H = 1, 8, 6400$ at source statistic $\eta = 0.11$.

5.2. Decoding Complexity and Latency

Without loss of generality, the case of $\boldsymbol{B}_{J1}$ at source statistic $\eta = 0.04$ is taken to compare the decoding complexity and latency. The decoding complexity can be evaluated by the decoding iteration number. Thus, the average iteration number K_{avg} for different grouping strategies is shown in Figure 6. With increasing E_s/N_0, the K_{avg} decreases, but the trend is getting smaller. The larger G^H has a smaller K_{avg} than that of the smaller G^H. For example, the cases of $G^H = 400$ and $G^H = 8$ respectively decrease by 40% and 34% compared with the case of $G^H = 1$ at $E_s/N_0 = -2.4$ dB, as shown in Table 5.

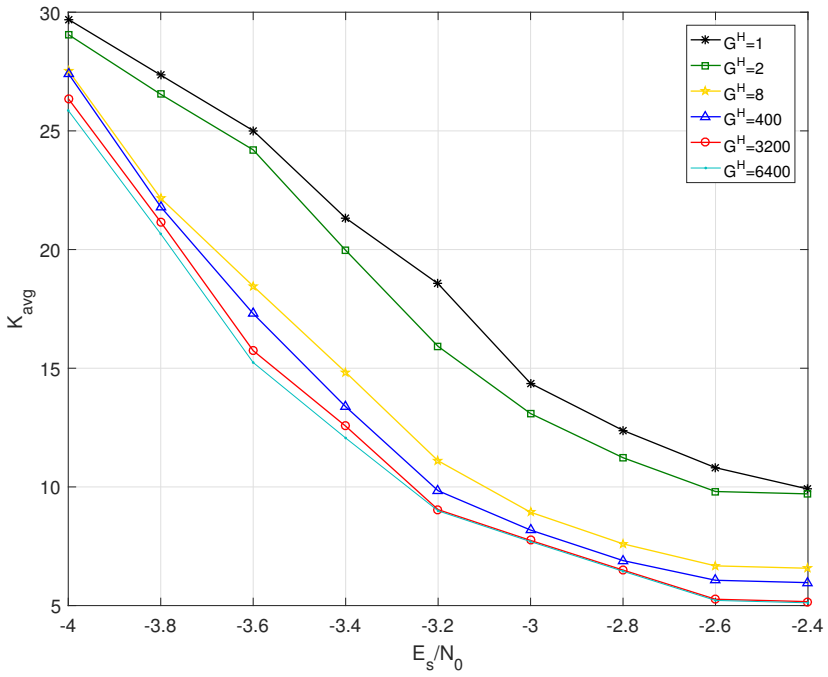

Figure 6. Average iteration number K_{avg} of $\boldsymbol{B}_{J1}$ with $G^H = 1, 2, 8, 400, 3200, 6400$ at source statistic $\eta = 0.04$.

The decoding latency indicates the average time taken for information bits to be decoded. The decoding procedure consists of the C2V and V2C, and the total number of C2V and V2C depends on the average degree of CNs and VNs. Considering the same $\boldsymbol{B}_{J1}$ used here, the decoding time of one group of C2V and V2C is the same, denoted as t_{uni}. Because the shuffled scheduling decoding algorithm is serial but the BP decoding algorithm is parallel, different grouping strategies imply the combination of serial and parallel methods. Take the case $G^H = 400$ as an example: 400 groups perform the decoding procedure one by one, and every group of these 400 groups performs parallel decoding. Thus, the process will consume 400 t_{uni} in an iteration. If the average iteration is K_{avg}, the average time of decoding information bits can be calculated by

$$T_{avg} = t_{uni} \times G^H \times K_{avg}. \tag{29}$$

As shown in Table 5, the decoding latency increases with the increase of G_H.

Table 5. The decoding complexity (average iteration number K_{avg}) and the decoding latency (average decoding time T_{avg}) for different grouping strategies of $\boldsymbol{B}_{J1}$ at $\eta = 0.04$ and $E_s/N_0 = -2.4$ dB.

Grouping Strategy	K_{avg}	T_{avg}
$G^H = 1$	9.9	$9.9t_{uni}$
$G^H = 2$	9.7	$19.4t_{uni}$
$G^H = 8$	6.5	$52t_{uni}$
$G^H = 400$	5.9	$2360t_{uni}$
$G^H = 3200$	5.1	$16{,}320t_{uni}$
$G^H = 6400$	5.1	$32{,}640t_{uni}$

A reasonable decoding solution, i.e., the grouping strategy, should take BER performance, decoding complexity and decoding latency into consideration.

6. Conclusions

In this paper, a JGSSD algorithm for the D-LDPC codes system is proposed. The proposed algorithm considers the D-LDPC coding structure as a whole. In order to analyze the performance of different grouping strategies, a JSEXIT algorithm is proposed for the general D-LDPC coding structure, i.e., both the cases of $\boldsymbol{B}_{l2} = 0$ and $\boldsymbol{B}_{l2} \neq 0$. It can be seen that both the source decoding threshold and the channel decoding threshold improve as G^B increases. The BER simulation is in line with the EXIT analysis, i.e., the BER performance has a better coding gain or lower error floor level when the G^H has a higher value. In addition, the decoding complexity and decoding latency are also compared, and it is shown that the larger G^H gives a lower decoding complexity but a higher decoding latency. Thus, a suitable shuffled scheduling decoding algorithm should give overall consideration to factors including performance, complexity and latency, and the JGSSD algorithm provides an intelligent choice. In future, the performance of the JGSSD algorithm can be studied for specific applications, such as multiple fading channels, and the optimization of the D-LDPC coding structure with the aid of the JSEXIT algorithm for different grouping strategies can be studied.

Author Contributions: Conceptualization, Q.C.; methodology, Q.C., Y.R. and C.C.; software, Q.C.; validation, L.Z. and Q.C.; formal analysis, Q.C. and Y.R.; investigation, Q.C. and Y.R.; resources, Q.C.; data curation, Q.C.; writing—original draft preparation, Q.C.; writing—review and editing, Y.R., S.L. and C.C.; visualization, L.Z.; supervision, L.Z.; project administration, Q.C. and C.C.; funding acquisition, Q.C. and C.C. 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 62101195 and 61901182, the Science Foundation of the Fujian Province, China (Grant No. 2020J05056), the Fundamental Research Funds for the Central Universities (ZQN-1008, ZQN-1005) and the Scientific Research Funds of Huaqiao University (20BS105, 19BS206, 21BS118).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AWGN	additive white Gaussian noise
APP	a posteriori probability
BER	bit error rate
BP	belief propagation
CNs	check nodes
C2V	check-to-variable
D-LDPC	double low-density parity-check
EXIT	extrinsic information transfer
IBP	iterative belief propagation
IoT	Internet of things
JBP	joint belief propagation
JEXIT	joint extrinsic information transfer
JPEXIT	joint protograph EXIT
JGSSD	joint group shuffled scheduling decoding
JSCC	joint source-channel coding
JSEXIT	joint shuffled EXIT
LLR	log-likelihood-ratio
MI	mutual information
PEG	progressive edge growth
SPEXIT	source protograph EXIT
SSSD	separated shuffled scheduling decoding
SWD	sliding window decoding
V2C	variable-to-check
VNs	variable nodes

References

1. Chen, Q.; Wang, L.; Chen, P.; Chen, G. Optimization of component elements in integrated communication systems: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *3*, 2977–2999. [CrossRef]
2. Fresia, M.; Perez-cruz, F.; Poor, H.V.; Verdu, S. Joint source and channel coding. *IEEE Signal Process. Mag.* **2010**, *6*, 104–113. [CrossRef]
3. Bi, C.; Liang, J. Joint source-channel coding of jpeg 2000 image transmission over two-way multi-relay networks. *IEEE Trans. Image Process.* **2017**, *7*, 3594–3608. [CrossRef] [PubMed]
4. Zhou, C.; Lin, C.; Zhang, X.; Guo, Z. A novel JSCC scheme for UEP-based scalable video transmission over MIMO systems. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *6*, 1002–1015. [CrossRef]
5. Wang, K.; Wang, Y.; Sun, Y.; Guo, S.; Wu, J. Green industrial Internet of Things architecture: An energy-efficient perspective. *IEEE Commun. Mag.* **2016**, *12*, 48–54. [CrossRef]
6. Zhang, J.; Fossorier, M.P.C. Shuffled iterative decoding. *IEEE Trans. Commun.* **2005**, *2*, 209–213. [CrossRef]
7. Xu, Z.; Wang, L.; Hong, S.; Lau, F.; Sham, C. Joint shuffled scheduling decoding algorithm for DP-LDPC codes-based JSCC systems. *IEEE Wirel. Commun. Lett.* **2019**, *6*, 1696–1699. [CrossRef]
8. Xu, Z.; Wang, L.; Hong, S.; Chen, G. Generalized joint shuffled scheduling decoding algorithm for the JSCC system based on protograph-LDPC codes. *IEEE Access* **2021**, *9*, 128372–128380. [CrossRef]
9. He, J.; Wang, L.; Chen, P. A joint source and channel coding scheme based on simple protograph structured codes. In Proceedings of the Symposium of on Communications and Information Technologies (ISCIT), Sydney, Australia, 2–5 October 2012; pp. 65–69.
10. Lin,H.; Lin, S.; Abdel-Ghaffar, K. Integrated code design for a joint source and channel LDPC coding scheme. In Proceedings of the IEEE Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 12–17 February 2017; pp. 1–9.
11. He, J.; Li, Y.; Wu, G.; Qian, S.; Xue, Q.; Matsumoto, T. Performance improvement of joint source-channel coding with unequal power allocation. *IEEE Wirel. Commun. Lett.* **2017**, *5*, 582–585. [CrossRef]
12. Neto, H.; Henkel, W. Multi-edge optimization of low-density parity-check codes of joint source-channel coding. In Proceedings of the IEEE Systems, Communication and Coding (SCC), Munich, Germany, 21–24 January 2013.

13. Chen, C.; Wang, L.; Xiong, Z. Matching criterion between source statistics and source coding rate. *IEEE Commun. Lett.* **2015**, *9*, 1504–1507. [CrossRef]
14. Chen, Q.; Hong, S.; Chen, Y. Design of linking matrix in JSCC scheme based on double protograph LDPC codes. *IEEE Access* **2019**, *7*, 92176–92183. [CrossRef]
15. Chen, C.; Wang, L.; Liu, S. The design of protograph LDPC codes as source code in a JSCC system. *IEEE Commun. Lett.* **2018**, *4*, 672–675. [CrossRef]
16. Chen, Q.; Lau, F.C.M.; Wu, H.; Chen, C. Analysis and improvement of error-floor performance for JSCC Scheme based on double protograph LDPC codes. *IEEE Trans. Veh. Technol.* **2020**, *12*, 14316–14329. [CrossRef]
17. Wu, H.; Wang, L.; Hong, S.; He, J. Performance of joint source channel coding based on protograph LDPC codes over Rayleigh fading channels. *IEEE Commun. Lett.* **2014**, *4*, 652–655. [CrossRef]
18. Chen, Q.; Wang, L.; Hong, S.; Xiong, Z. Performance improvement of JSCC scheme through redesigning channel codes. *IEEE Commun. Lett.* **2016**, *6*, 1088–1091. [CrossRef]
19. Hong, S.; Ke, J.; Wang, L. Global design of double protograph LDPC codes for joint source-channel coding. *IEEE Commun. Lett.* **2023**, *2*, 424–427. [CrossRef]
20. Chen, Q.; Wang, L.; Hong, S.; Chen, Y. Integrated design of JSCC scheme based on double protograph LDPC codes system. *IEEE Commun. Lett.* **2019**, *2*, 218–221. [CrossRef]
21. Nguyen, T.; Nosratinia, A.; Divsalar, D. The design of rate-compatible protograph LDPC codes. *IEEE Trans. Commun.* **2012**, *10*, 2841–2850. [CrossRef]
22. Chen, C.; Wang, L.; Lau, F. Joint optimization of protograph LDPC code pair for joint source and channel coding. *IEEE Trans. Commum.* **2018**, *8*, 3255–3267. [CrossRef]
23. Liu, S.; Chen, C.; Wang, L.; Hong, S. Edge connection optimization for JSCC system based on DP-LDPC codes. *IEEE Wirel. Commun. Lett.* **2019**, *4*, 996–999. [CrossRef]
24. Liu, S.; Wang, L.; Chen, J.; Hong, S. Joint component design for the JSCC system based on DP-LDPC codes. *IEEE Trans. Commum.* **2020**, *9*, 5808–5818. [CrossRef]
25. Neto, H.V.B.; Henkel, W. Information shortening for joint source-channel coding schemes based on low-density parity-check codes. In Proceedings of the 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Bremen, Germany, 18–22 August 2014; pp. 1–5.
26. Chen, Q.; Wang, L. Design and analysis of joint source channel coding schemes over non-standard coding channels. *IEEE Trans. Veh. Technol.* **2020**, *8*, 5369–5380. [CrossRef]
27. Xu, Z.; Wang, L.; Chen, G. Joint coding/decoding optimization for DC-BICM system: Collaborative design. *IEEE Commun. Lett.* **2021**, *8*, 2487–2491. [CrossRef]
28. Song, D.; Ren, J.; Wang, L.; Chen, G. Designing a common DP-LDPC codes pair for variable on-body channels. *IEEE Trans. Wirel. Commun.* **2022**, *11*, 9596–9609. [CrossRef]
29. Golmohammadi, A.; Mitchell, D.G.M. Concatenated spatially coupled LDPC codes with sliding window decoding for joint source—Channel coding. *IEEE Trans. Commun.* **2022**, *2*, 851–864. [CrossRef]
30. Lian, Q.; Chen, Q.; Zhou, L.; He, Y.; Xie, X. Adaptive decoding algorithm with variable sliding window for double SC-LDPC coding system. *IEEE Commun. Lett.* **2023**, *2*, 404–408. [CrossRef]
31. Xu, Z.; Wang, L.; Hong, S. Joint early stopping criterions for protograph LDPC codes-based JSCC system in images transmission. *Entropy* **2021**, *23*, 1392. [CrossRef]
32. Deng, L.; Shi, Z.; Li, O.; Ji, J. Joint coding and adaptive image transmission scheme based on DP-LDPC codes for IoT scenarios. *IEEE Access* **2019**, *7*, 18437–18449. [CrossRef]
33. Hu, X.-Y.; Eleftheriou, E.; Arnold, D.M. Regular and irregular progressive edge—Growth Tanner graphs. *IEEE Trans. Inf. Theory* **2005**, *1*, 386–398. [CrossRef]
34. 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**, *12*, 3770–3774. [CrossRef]