Chapter

Generalized Low-Density Parity-Check Codes: Construction and Decoding Algorithms

Sherif Elsanadily

Abstract

Scientists have competed to find codes that can be decoded with optimal decoding algorithms. Generalized LDPC codes were found to compare well with such codes. LDPC codes are well treated with both types of decoding; HDD and SDD. On the other hand GLDPC codes iterative decoding, on both AWGN and BSC channels, was not sufficiently investigated in the literature. This chapter first describes its construction then discusses its iterative decoding algorithms on both channels so far. The SISO decoders, of GLDPC component codes, show excellent error performance with moderate and high code rate. However, the complexities of such decoding algorithms are very high. When the HDD BF algorithm presented to LDPC for its simplicity and speed, it was far from the BSC capacity. Therefore involving LDPC codes in optical systems using such algorithms is a wrong choice. GLDPC codes can be introduced as a good alternative of LDPC codes as their performance under BF algorithm can be improved and they would then be a competitive choice for optical communications. This chapter will discuss the iterative HDD algorithms that improve decoding error performance of GLDPC codes. SDD algorithms that maintain the performance but lowering decoding simplicity are also described.

Keywords: channel coding, generalized LDPC codes, iterative decoding, bit-flipping, chase algorithm

1. Introduction

Generalized LDPC (GLDPC) block codes were first proposed by Tanner [1] as they internally contain block codes (which are called component codes) and not just single parity check (SPC) as the case in LDPC codes. From this definition, we know that LDPC codes can be regarded as a special class of GLDPC codes. GLDPC block codes possess many desirable features, such as large minimum distance [2], good iterative decoding performance, and low error floor [3]. At the same time, the complexity of the processed operations increases due to the inserted complicated constraints. Therefore, scientists are stirred to find a good GLDPC code with suitable subcodes achieving the desired error performance at fair complication. The methods in [4] by Fossorier and the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm

[5], APP decoding algorithm, are vastly investigated for GLDPC decoding. Both algorithms perform the task with various subcodes, such as Hamming codes [2, 6], BCH codes [6, 7], RS codes [7], and GLDPC code with hybrid subcodes [8, 9]. However, these decoders can be considered highly complicated. In [10], a kind of Hadamard-based GLDPC codes was suggested. The complicated decoding processes were avoided due to the easiness and fastness of the used subcode transform (FHT). On the other hand, this code is not convenient in many communication systems as it is described as low code rate ($R \leq 0.1$). The soft-input/soft-output (SISO) decoders for decoding the component codes in GLDPC codes were considered in [2, 6] and [11] showing very good error performance with moderate and high code rate. Two interesting approaches at GLDPC codes have been presented in [4, 12]. In [12] the proposed construction concentrates on maximizing the girth and the minimum distance of the global code. The generated codes achieved coding gain up to 11 dB over 40 GB/s optical channel. In [4] the authors propose doubly generalized LDPC codes. These codes employ local codes at both variable and check nodes. Back to LDPC codes and since their resurrection, most research efforts were directed toward the implementation of these codes over the additive white Gaussian noise (AWGN) channel. The LDPC codes were proven to perform very close to the Shannon limit of AWGN channel, and much work has been carried out to design optimal codes and improve and simplify iterative decoding of these codes over the AWGN channel. The major drawback is that it exhibits considerable computational complexity. Gallager accurately, on the other hand, analyzed the performance of LDPC codes over the binary symmetric channel (BSC) in his original paper and proposed two HDD bit-flipping (BF) algorithms, for which he provided theoretical limits under iterative decoding. However, BF algorithms did not gain much attention. Most of the work that considered BF decoding used it in conjunction with soft information obtained from AWGN channel to improve decoding performance. However, in some applications, such as optical communications, soft values are not available at the receiver. Therefore, optical channels are an excellent example of the BSC, and only hard decision decoding is possible. Currently, BCH and RS codes are exclusively used in optical communication for error control, since there are simple and efficient algorithms for decoding. With the recent introduction of wavelength division multiple access (WDMA), transmission rates in optical communications reach 40 GB/s per channel/fiber, a standard in modem optical networks, SONET/SDH. Moreover, the concept of signal regeneration was abandoned in optical communication with the advancement in lasers so that optical signal is transmitted over larger distance than before, reaching the receiver very attenuated. These developments in optical communication call for an error control code which is very powerful yet has a simple and fast decoding. Furthermore, very low error rates are needed, say BER of 10^{-15}.

The BF algorithm proposed by Gallager is a HDD algorithm and is implemented using modulo-2 logic. Therefore it satisfies the requirement for simplicity and speed. However, the performance of the decoding algorithm is far from the capacity of the BSC, and, more importantly, an error floor is generally observed, which seriously constrains implementation of LDPC codes in optical systems. GLDPC codes can be introduced as a good alternative of LDPC codes as their performance under the BF algorithm can be improved and the observed error floor can be lowered or even removed. GLDPC codes would then be a competitive choice for optical communications.

In this chapter, iterative decoding of GLDPC codes over AWGN and BSC is studied. HDD algorithms that improve decoding performance and error floor behavior of GLDPC codes over BSC channels are discussed. They make GLDPC codes very competitive for high-rate optical communications. Soft decision decoding (SDD) algorithms that maintain the performance but lower decoding simplicity are also presented.

2. GLDPC code construction

The check node in the bipartite graph of the LDPC code, as said before, is connected to a number of variable nodes which satisfy a single parity check. The GLDPC is a more generalized form of LDPC as the bits of the VNs, connected to the same CN, constitute a valid codeword of a (n,k) linear block code (other than the simple SPC code). Therefore this (n,k) code is called a constituent code, component code, or simply subcode. The CN which associates with this generalized code is called a generalized CN (GCN). The GLDPC code is referred to as a (N, J, n) regular and "strict-sense" code, as depicted in **Figure 1**, if:

- The VN degree (denoted as q_w) is constant for all VNs ($q_w = J$).
- The GCN degree (denoted as q_c) is constant for all GCNs ($q_c = n$).
- The same constituent code (other than the simple SPC code) stands for all GCNs.

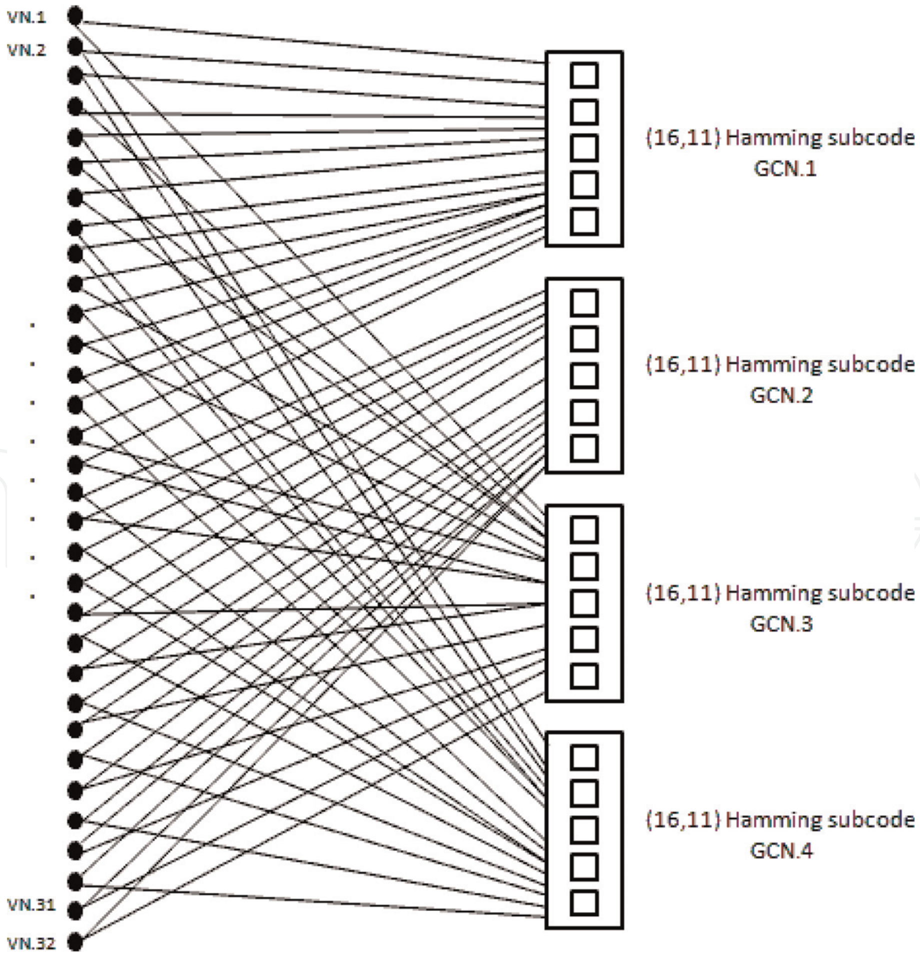
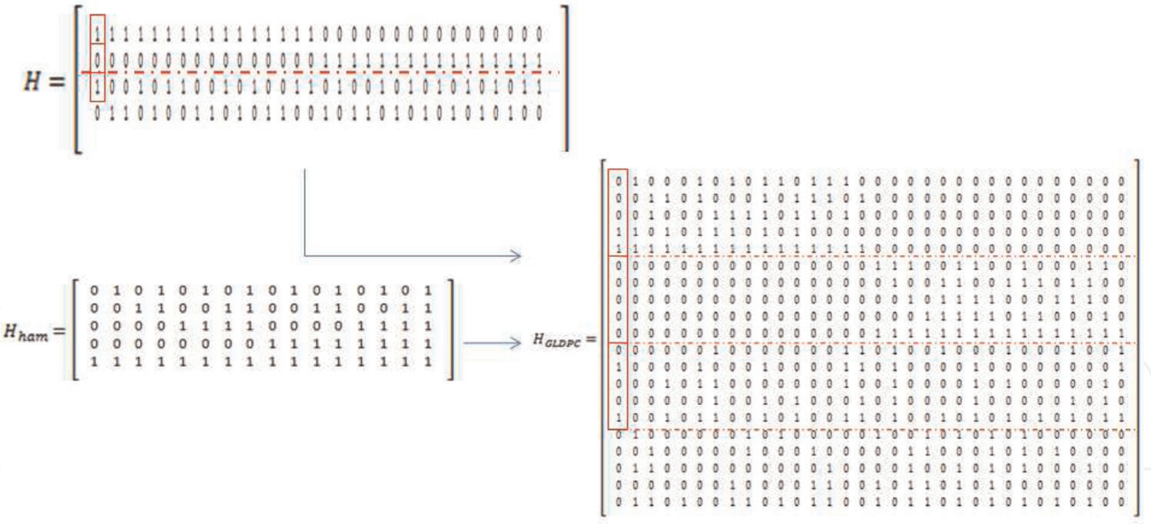


Figure 1.
The bipartite graph of the strict-sense (32,2,16) regular GLDPC code.



$$H = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&0&0&1&0&1&1&0&0&1&0&1&0&0&1&1&0&1&0&0&1&0&1&0&1&0&1&0&1&0&1&1 \\ 0&1&1&0&1&0&0&1&1&0&1&0&1&1&0&0&1&0&1&1&0&1&0&1&0&1&0&1&0&1&0&0 \end{bmatrix}$$

$$H_{ham} = \begin{bmatrix} 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

$$H_{GLDPC} = \begin{bmatrix} 0&1&0&0&0&1&0&1&0&1&1&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&1&0&0&0&1&0&1&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&1&1&1&1&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&1&0&1&1&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&1&1&0&0&1&0&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&0&0&1&1&1&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&1&1&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&1&1&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&1&0&0&1&0&0&0&1&0&0&0&1&0&0&1 \\ 1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&1&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&1 \\ 0&0&0&1&0&1&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1&0&0&1&0&1&0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0&1&0&1&0 \\ 1&0&0&1&0&1&1&0&0&1&0&1&0&0&1&1&0&1&0&0&1&0&1&0&1&0&1&0&1&0&1&1 \\ 0&1&0&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&1&0&1&0&1&0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&1&1&0&0&0&1&0&1&0&1&0&0&0&0 \\ 0&1&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1&0&1&0&1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&1&1&0&0&1&0&1&1&0&1&0&1&0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&1&1&0&1&0&1&1&0&0&1&0&1&1&0&1&0&1&0&1&0&1&0&1&0&0 \end{bmatrix}$$

Figure 2.
The procedure of generating a (32,2,16) Hamming-based GLDPC code from its base matrix.

where J is the column weight in the parity-check matrix of the global LDPC code and N is the overall code block length. The GLDPC code is otherwise called "hybrid code" (if not all the GCNs has the same constituent code) [8, 13].

The GLDPC code rate is given by $R = K/N \geq 1 - J(1 - k/n)$, where K denotes its code dimension and N denotes its block length. The GLDPC code, according to the chosen values of its parameters, has multiple wonderful properties such as the better minimum distance (compared to LDPC code with the same code rate) [2]. The GLDPC also converges faster, and it is distinguished by the lower error floor [3]. We are interested here with GLDPC codes based on Hamming codes for simplicity and fast decoding purposes.

Figure 1 elaborates the bipartite graph of a (N, J, n) regular Hamming-based GLDPC code with (4×32) global LDPC matrix. The extended Hamming (8,4) constituent code is represented in every GCN. **Figure 2** depicts the procedures to get the overall parity-check matrix of this code from the global LDPC matrix (or referred to as the graph adjacency matrix). Every "1" in every row in the global matrix is replaced with a column from the columns of the constituent code parity-check matrix, and every "0" is replaced with a zero column. The assignments of the constituent code H columns should be randomly done to generate a code with good characteristics. There are other constructions of GLDPC which can be further discussed in [14–16].

3. SDD of GLDPC codes

SDD of Hamming-based GLDPC codes was presented in the literature and shows that GLDPC codes are asymptotically good and can achieve the capacity by iterative decoding using soft-input/soft-output subcode decoders [2, 6, 17]. The SISO decoder is typically a sub-optimal erasure decoder extended to deliver soft outputs, e.g., chase-II decoder [18], which is used in this section.

A (N, J, n) GLDPC code is constructed from a $(N.J/n)$ by N random sparse matrix H, with row weights of n and column weights of J, and the parity-check matrix H_c of a (n, k, d) subcode. The resultant GLDPC parity-check matrix is denoted as H_{GLDPC} as discussed in Section 2.

At any GCN c, the input to the chase decoder is $R_c = \{r_{c,1}, \cdots, r_{c,i}, \cdots, r_{c,n}\}$ corresponding to the transmitted word $X_c = \{x_{c,1}, \cdots, x_{c,i}, \cdots, x_{c,n}\}$ and its hard demodulated values $Y_c = \left\{y_{c,1}, \cdots, y_{c,i}, \cdots, y_{c,n}\right\}$.

A group of codewords are selected as the most likely ones that hold the transmitted words with the minimum errors. The algorithm operates on the available data (reliability) and flips all possible combinations of $p = d/2$ demodulated symbols with the least-reliable positions (LRPs). Setting the reliability information in [19] in the log-likelihood ratio (LLR) of decision $y_{c,i}$ as

$$\Lambda\left(y_{c,i}\right) = \ln\left(\frac{pr(x_{c,i} = +1/x_{c,i})}{pr(x_{c,i} = -1/x_{c,i})}\right) = \left(\frac{2}{\sigma^2}\right) r_{c,i} \tag{1}$$

A set $Z_c = \left\{Z_c^q\right\}$ of error patterns with all possible errors confined to p LRP positions of Y_c is used to modify Y_c yielding list of test patterns $T_c^q (q \in \{1, \cdots, 2^p\}$, $T_c^q = Z_c^q + Y_c)$. Then every T_c^q in the list is decoded using algebraic decoder, and the valid decoded codeword C^q is stored in a list Ω as a candidate codeword. Decision codeword $D_c = \{d_{c,1}, \cdots, d_{c,i}, \cdots, d_{c,n}\}$ should be chosen from this list as it is obtained using the rule [20]:

$$D_c = C^q \;\; if \;\; |R_c - C^q|^2 \leq \left|R_c - C^l\right|^2 \text{for every } C^l \in \Omega \tag{2}$$

Now, every decoded symbol resulting value in the subcode (soft) is to be estimated to be passed back on the edges, linked to symbol nodes with steps as the MP algorithm iteratively to output a final estimate after performing a predefined amount of iterations or satisfying the syndrome check.

In order to calculate the reliability of each bit, $d_{c,i}$, in the decision D_c (i.e., the ith soft output of the soft-input decoder), two codewords $C^{+1(i)}$ and $C^{-1(i)}$ are to be selected from two sets of Ω with minimum Euclidean distance from R. The decision D_c is one of them, and the second one, $B_c = \{b_{c,1}, \cdots, b_{c,i}, \cdots, b_{c,n}\}$, called as competing codeword of D with $b_{c,i} \neq d_{c,i}$, should be found.

The soft outputs are generated in the LLR domain outgoing from GCN c using the following approximation formula:

$$r'_{c,i} = \left(\frac{|R_c - B_c|^2 - |R_c - D_c|^2}{4}\right) d_{c,i} \tag{3}$$

If the competing codeword B_c is not found, the following alternative and efficient formula is used:

$$r'_{c,i} = \beta \times d_{c,i} \;\; \text{with} \;\; \beta \geq 0 \tag{4}$$

where β is a reliability factor. Due to the variation of sample deviation in the input and in the output of the soft decoders, we put a scaling factor, α, to increase the convergence rate:

$$W_c(t+1) = R_c + \alpha(t) W_c(t) \tag{5}$$

By subtracting the soft input $r_{c,i}$ from the soft output $r'_{c,i}$ for each $i \in \{1, 2, \cdots, n\}$ at GCN c, the extrinsic information during the tth iteration, $W(t)$, is obtained. It is then multiplied by the scaling factor, $\alpha(t)$, added to the channel observed values R_c,

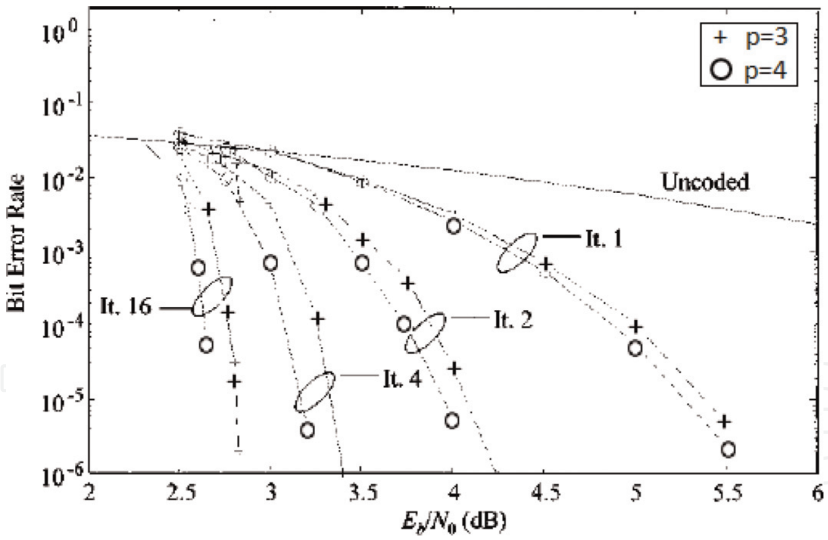


Figure 3.
Performance variation with erasures p for (65,536,2,64) GLD code.

and the result is considered as a priori information for the decoder at the next $(t+1)$th iteration.

Figure 3 shows that a chase rather than optimal SISO decoder can be successfully employed in the decoding of high-rate extended Hamming-based GLDPC codes and the BERs are close to the capacity with the efficient fast chase decoding in [21].

4. HDD of GLDPC codes

The HDD such as BF decoding or any other algebraic decoding scheme can be generalized to GLDPC codes especially over BEC or BSC and can be applied in very high-speed applications such as the 40 GBps optical communications. The error-correcting capability of the subcodes, at the GCNs, is used to more accurately determine the position of least-reliable symbols. The iterative HDD algorithms for decoding GLDPC codes will be described in the next subsections.

4.1 WBFV algorithm

As mentioned before in BF algorithm of LDPC codes, symbols belonging to the maximum number of unsatisfied CNs, in each iteration, have the binary bits inverted before the following iteration. Subsequently, the failed CNs convey their votes of unit weight to the corresponding connected VNs, and the algorithm inverts the LR bits with the highest amount of votes.

The presented iterative weighted bit-flip voting (WBFV) in [22] employs subcode hard decision decoders (HDDs) at the GCNs that have a greater range of vote weights passed to the connected VNs. It passes the high-weight votes to a specific symbol if the HDD, at a given GCN, considers it to be in error.

As the GLDPC codes of Gallager construction, $J=2$, and Hamming subcode are only concerned, all nonzero syndromes of these subcodes will be error-correctable. The algebraic decoders only allow for two instances:

- All-zero syndromes imply a valid codeword; in this case a vote V will be returned to all connected VNs (symbols) as in GCN 1 in **Figure 4**.

- Nonzero syndromes imply nonvalid codewords; in this case the indicated error position will be sent a vote E, and all other bits a vote e as in GCN 2 in **Figure 4**.

After votes have been cast, each symbol has received a vote pair: either VV, eV, ee, EV, Ee, or EE. In the example shown, the symbol at the two subcodes' intersection has a vote pair eV.

The strategy proceeds by passing current bit values from the VNs to the Hamming decoders at GCNs. The HDDs at these GCNs pass back n individual votes to the symbols they connect.

The magnitude of a vote marks the power of the decision for a GCN about the current symbol (reliable or not). The higher magnitudes mark the unreliable bits, and the lower ones mark the bits of more reliability.

The J arriving votes to every VN are collected such that all N variable symbols are sorted by the reliability information and the group of LRP bits inverted before the upcoming iteration.

The iterative algorithm proceeds until all symbols become of weight pair VV or the maximum number of iterations is reached. For Hamming subcodes there are only three votes V, e, and E generated by the subcode HDDs described before. The vote rules, as sets of vote weights, are defined and listed in **Table 1**.

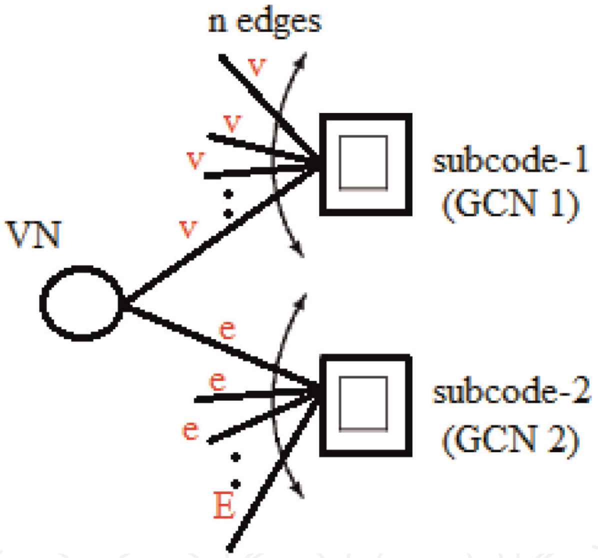


Figure 4.
Example votes from subcode decoders.

Rule A		Rule B		Rule C	
$V = 0, e = 2, E = 3$		$V = 0, e = 1, E = 2$		$V = 0, e = 1, E = 3$	
Vote pair	Total	Vote pair	Total	Vote pair	Total
EE	6	EE	4	EE	6
Ee	5	Ee	3	Ee	4
ee	4	EV, ee	2	EV	3
eV	2	VV	0	eV	1
VV	0			VV	0

Table 1.
Vote pair orderings for three vote rules.

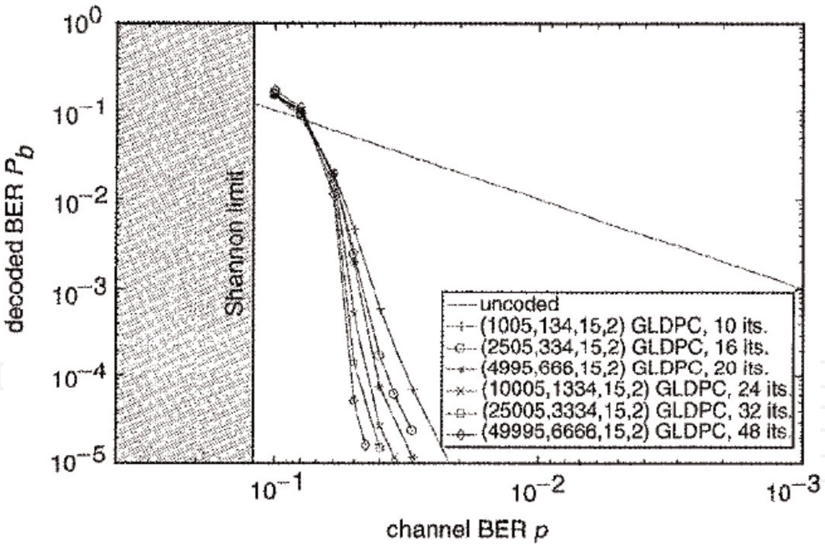


Figure 5.
Variation in performance with block length for (N,M,15,2) GLDPC codes.

Figure 5 shows that there is an obvious coding gain with increasing N. Actually, the BER curve p required to give $P_b = 10^{-5}$ versus N showing a linear relationship between $\log p$ and $\log N$.

4.2 BCH-based Fossorier decoding algorithm

In [6] GLDPC codes with BCH subcodes (instead of Hamming subcodes) were considered, but for AWGN channel and ML soft decoding due to its higher error-correcting capability. The algorithm presented here also uses the high-rate BCH and RS codes, but as HDD algorithm, and can be efficiently applied in very high-speed (40 GBps) optical systems as the soft information is not available due to optical-electrical conversions [7].

The actions of this algorithm are updated as follows: VN i is considered to be connected to the GCNs j and k. Two messages will be sent from GCN j to VN i. First, it outputs u_{ji} (the value of VN i) taken out from the sub-decoder. Second, it outputs U_{ji} (represents data about success or failure of GCN decoding). U_{ji} is then a binary signaling with estimate 1 if there is a valid word or estimate 0 if there is not. W.r.t the arriving messages from node k, the same action is applied.

The VN message v_{ij} depends on the value received from the channel y_i, and the messages received from the GCNs other than node j. Since GLDPC codes of high code rate ($J = 2$) is only concerned, there is only one other GCN k). Hence, the updating rule in the symbol node can be expressed as

$$v_{ij} = y_i \cdot \overline{U_{ki}} + u_{ki} \cdot U_{ki} \tag{6}$$

where $\overline{U_{ki}}$ is the complement of U_{ki}. The processes of the decoder are carried out until the satisfaction of all GCNs or until reaching a certain amount of iterations. Finally, any symbol (VN) connected to a satisfied GCN will take its proposed value. If the VN connected to two unsatisfied GCNs, it will take the original received value.

RS-based GLDPC code of rate $r = 0.467$ and different lengths are examined (**Figure 6**). It shows that RS(15,11,5) product code performs better than the corresponding GLDPC of length 225, because it possesses a better minimum distance. As N increases, the BER is improved particularly in the region of error floor.

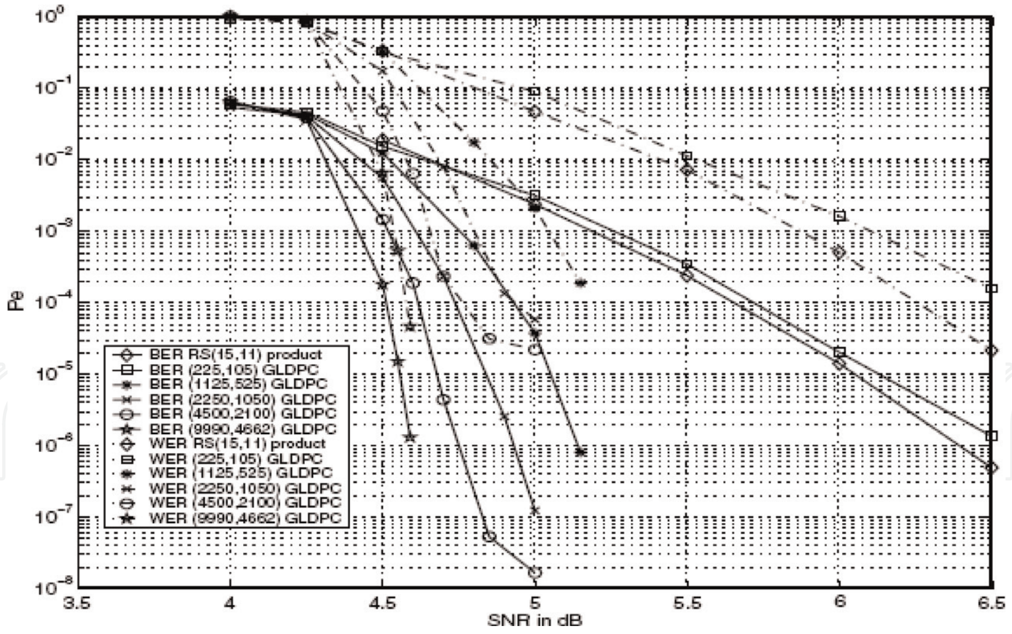


Figure 6.
Performance of GLDPC codes with RS(15,11,5) codes.

5. Modified HDD algorithms for improving the error performance

5.1 Two-side state-aided bit-flipping (TSSA-BF) algorithm

A HDD decoder founded on BF algorithm is presented in [23]. The main view depends on taking advantage of the different sub-decoder states. Therefore it adds this data representing these states as a reliability factor to be utilized for the rest of the decoding process. This additional data (in the form of additional bit) is inserted in two sides (VNs and CNs).

The goal is generally to remove all likely produced trapping sets (that generate non-editable errors) from the code construction. This approach is presently still not obtainable due to constraints in the processing speed and implementation. An alternative approach that fulfills the most of this goal is suggested (with reasonable processing speeds). That approach adds resources only operated in unusual circumstances when the previously mentioned trapping sets happen.

Taking commonness into account and due to its simplicity, the extended Hamming code is studied inside the GLDPC codes.

5.1.1 Failure analysis

In the case of using the ext-Hamming code $d_{min} = 4$, the sub-decoders of GLDPC code may output errors in both cases of decoding success or failure. At GCN failure case, it cannot clearly locate the error place. In the case of GCN decoding success, the errors may be generated from undiscovered errors (when a received word decoded to non-sent valid ones, i.e., $e \geq 4$) or faulty repair (when $e > 2$). Therefore, the errors at a given GCN can be distinguished by the following names:

1. Plain single error (element P): one error and true correction take place.

2. Unknown set (U-set): multiple errors ($e > 1$) with decoder failure (detects but can't mark errors).

3. Ambiguous set (A-set): multiple errors ($e > 2$) making the decoder flipping an assumed-correct bit (false correction).

4. Dark set (D-set): multiple errors ($e \geq 4$) not detected by the decoder which produce zero-syndrome vector.

5.1.2 Algorithm description

As GLDPC code with 1B-construction is studied in [22] as in **Figure 7**, a further bit is inserted beside the main bit moving between the VN and GCN. For VNs, it acts as the reliability of its bit value (bit 1 if suspect or bit 0 if assumed correct), and it forwards this additional data to be used by GCNs. For GCNs, this additional bit acts as the power of its decision by raising the reliability levels to 4. The GCN decodes the received word, forwards a signal (bit 1 or 0, namely, flip or keep), and appends this additional bit as the power of this signal (in ascending reliability level arrangement, $1^{+}(11)$, $1^{-}(10)$, $0^{-}(00)$ and $0^{+}(01)$, corresponding strong flip, weak flip, weak keep, and strong keep, respectively).

5.1.2.1 Horizontal processing

Specifying the ext-Hamming decoder, there are three states at a given GCN as state "0" in the case of zero syndrome, state "1" in the case of one-error repair, and state "2" in the case of decoder failure.

For any GCN c, $c = 1, 2, \cdots, M$, let $d_c^{(l)}$ be the cth GCN decoder state at the lth iteration. If $d_c^{(l)} = 0$, the decoder forwards a message (0^{-}) to all set elements of its connected VNs, $W(c) = \{w_1, w_2, .., w_i, .., w_n\}$. The message (0^{-}) with reliability level 3 is forwarded assuming $W(c)$ may contain a dark set (D-set). If $d_c^{(l)} = 1$, the decoder forwards (1^{+}) with level 1 on the assumed-error place w^{*} and (0^{-}) to the remaining set elements $W^{'}(c)$. Elements of $W^{'}(c)$ are given 0^{-} (not 0^{+}) assuming that they may contain A-set. Finally, If $d_c^{(l)} = 2$, it sends (1^{-}) with level 2 to all elements of $W(c)$ which contains a U-set.

As illustrated in **Figure 8**, let $U_{c,w_i}^{(l)} = \left[U_{c,w_i}^{(l)}(1), U_{c,w_i}^{(l)}(2)\right]$ be the two bits representing the outgoing flip message and its reliability, respectively, from GCN c to VN w_i. **Table 2** illustrates the four possible outgoing messages from GCN c to every connected VN w_i.

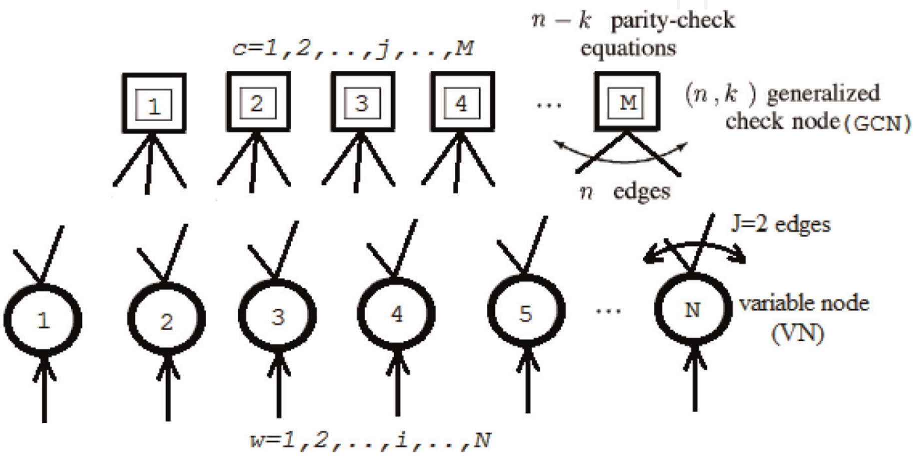


Figure 7.
GLDPC bipartite graph example.

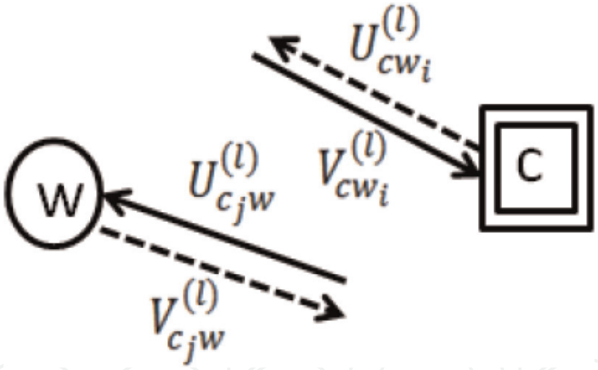


Figure 8.
Incoming and outgoing messages between GCNs and VNs.

$U^{(l)}_{c,w_i}(1)U^{(l)}_{c,w_i}(2)$	Alternative denotation	Reliability grade	Message meaning
1 1	1^+	1	Strong flip
1 0	1^-	2	Weak flip
0 0	0^-	3	Weak keep
0 1	0^+	4	Strong keep

Table 2.
Possible outgoing messages from GCN c to any connected VN.

Let $V^{(l)}_{c,w_i} = \left[V^{(l)}_{c,w_i}(1), V^{(l)}_{c,w_i}(2)\right]$ be the two bits representing arriving bit value and its reliability, respectively, at GCN c from VN w_i. For any GCN c, let initial reliability bit be $V^{(0)}_{c,w_i}(2) = 0$ for all $w_i \in W(c)$. As previously mentioned, $V^{(l)}_{c,w_i}(2) = 1$ if a suspect bit and 0 if assumed correct one.

Now a local GCN counter will be introduced to compare the present GCN state with the one of previous iteration. Let $\alpha^{(l)}_c$ be the number of consecutive previous repetitions of the state $d^{(l)}_c$, and let $\beta^{(l)}_c$ be the number of incoming suspects among the n bits connected to GCN c. According to $\alpha^{(l)}_c$, $\beta^{(l)}_c$ values and the additional information $V^{(l)}_{c,w_i}(2)$, the algorithm can improve its decision, and the horizontal process iteratively continues as illustrated in [23].

For $d^{(l)}_c = 0$, the counter role is to enhance the reliability from 0^- to 0^+ if the state remains for two consecutive iterations. For any GCN c and $d^{(l)}_c = 2$, if the message $V^{(l)}_{c,w_i}(2)$ from the bit w_i indicates a suspected bit and the decoder state $(d^{(l)}_c = 2)$ remains for three consecutive iterations, the decoder recalculates the syndrome after flipping this suspect. If the syndrome check is satisfied (i.e., valid codeword), the decoder estimates that bit as error and degrades its reliability from 1^- to 1^+.

5.1.2.2 Vertical processing

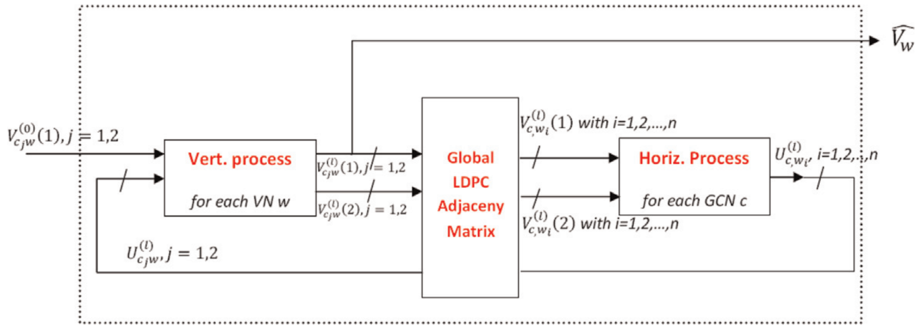
With four reliability levels and $J = 2$, there will be a set of 10 possible combinations of incoming messages at VN w, $w = 1, 2, \cdots, N$. This set can be divided, according to the failure analysis, into three subsets $\{s_i, i = 1, 2, 3\}$. For any VN w, its reliability state $r^{(l)}_w$ is determined based on its incoming messages (from GCNs c_j, with $j = \{1, 2\}$) to which subset it belongs:

$$
\begin{aligned}
s_1 &= \{(1^+,1^+),(1^+,1^-),(1^+,0^-)\},\\
s_2 &= \{(1^-,1^-),(1^-,0^-),(0^+,1^+)\},\\
s_3 &= \{(0^-,0^-),(0^+,1^-),(0^+,0^-),(0^+,0^+)\}
\end{aligned}
$$

$$
r_w^{(l)} \quad = \begin{cases} 0 & \text{if } \left\{U_{c_jw}^{(l)}, j=1,2\right\} \in s_1 \\ 1 & \text{if } \left\{U_{c_jw}^{(l)}, j=1,2\right\} \in s_2 \\ 2 & \text{if } \left\{U_{c_jw}^{(l)}, j=1,2\right\} \in s_3 \end{cases}
$$

The VN w with least-reliable level, $r_w^{(l)} = 0$, needs to be flipped. For $r_w^{(l)} = 1$, VN state counter is appointed to make a comparison between the VN current reliability state and the one of the previous iteration. Let $\gamma_w^{(l)}$ be the number of consecutive previous repetitions of the state $r_w^{(l)}$. With $r_w^{(l)} = 1$ and according to $\gamma_w^{(l)}$ values, the VN will not be flipped but counted as a suspected bit. It sends such reliability information $\left(V_{c_jw}^{(l)}(2) = 1, j = 1,2\right)$ to GCNs to be taken into account. For $r_w^{(l)} = 2$, the VN is considered a reliable bit and kept with $V_{c_jw}^{(l)}(2) = 0, j = 1, 2$ (i.e., assumed correct bit).

Figures 9–11 show the block diagrams of the overall decoder, horizontal process, and vertical process, respectively. **Table 3** illustrates an example for ext-Hamming (8,4) constituent decoder employed at GCN $c(n = 8)$ at lth iteration as the shaded parts represent certain conditions satisfied.



$\widehat{V_w}$
$V_{c_jw}^{(0)}(1), j = 1,2$
Vert. process
for each VN w
$V_{c_jw}^{(l)}(1), j = 1,2$
$V_{c_jw}^{(l)}(2), j = 1,2$
$U_{c_jw}^{(l)}, j = 1,2$
Global
LDPC
Adjaceny
Matrix
$V_{c,w_i}^{(l)}(1)$ with i=1,2,…,n
$V_{c,w_i}^{(l)}(2)$ with i=1,2,…,n
Horiz. Process
for each GCN c
$U_{c,w_i}^{(l)}$, i=1,2,…,n

Figure 9.
Block diagram of the overall TSSA-BF decoder.

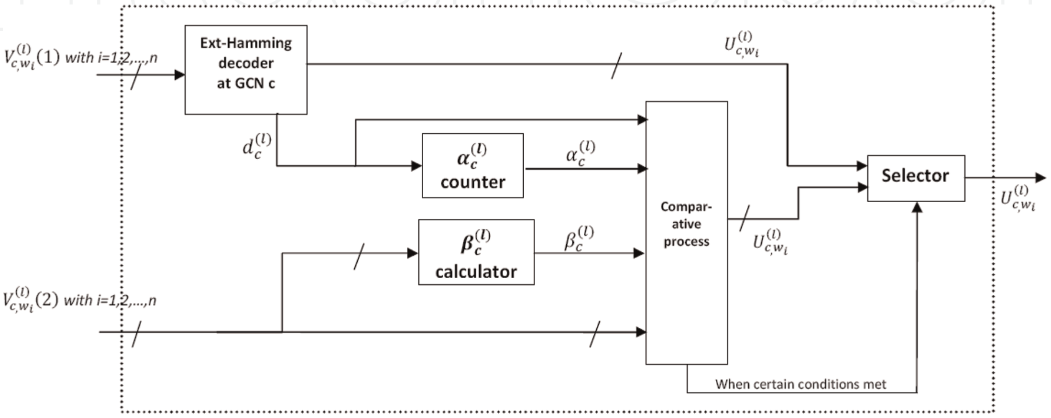


Figure 10.
Block diagram of the horizontal process of the TSSA-BF decoder.

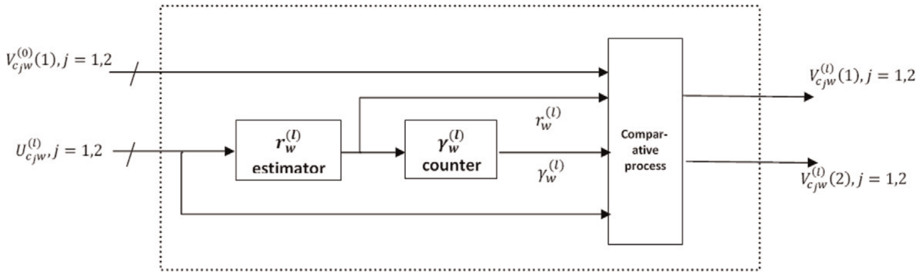

$V_{c_jw}^{(0)}(1), j = 1{,}2$
$U_{c_jw}^{(l)}, j = 1{,}2$
$r_w^{(l)}$
estimator
$\gamma_w^{(l)}$
counter
$r_w^{(l)}$
$\gamma_w^{(l)}$
Compar-ative process
$V_{c_jw}^{(l)}(1), j = 1{,}2$
$V_{c_jw}^{(l)}(2), j = 1{,}2$

Figure 11.
Block diagram of the vertical process of the TSSA-BF decoder.

GCN State ($d_c^{(l)}$)	STATE Counter ($\alpha_c^{(l)}$)	# of suspects ($\beta_c^{(l)}$) if existed	$V_{c,w_i}^{(l)}(2), i = 1,2,..,n$ with example on certain locations for suspect bits								o/p messages $U_{c,w_i}^{(l)}, i = 1,2,..,n$							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
0	< 2	-									0^-	0^-	0^-	0^-	0^-	0^-	0^-	0^-
	≥2	$\beta_c^{(l)} = 0$	0	0	0	0	0	0	0	0	0^+	0^+	0^+	0^+	0^+	0^+	0^+	0^+
		$\beta_c^{(l)} \geq 4$	1	1	0	1	0	0	0	1	0^-	0^-	0^+	0^-	0^+	0^+	0^+	0^-
1 (with $w^* = w_3$)		-									0^-	0^-	1^+	0^-	0^-	0^-	0^-	0^-
2	<3	-									1^-	1^-	1^-	1^-	1^-	1^-	1^-	1^-
	≥3	$\beta_c^{(l)} = 1$	0	1	0	0	0	0	0	0	0^-	1^+	0^-	0^-	0^-	0^-	0^-	0^-
		$\beta_c^{(l)} = 2$	0	0	1	0	0	1	0	0	0^-	0^-	1^+	0^-	0^-	1^+	0^-	0^-

Table 3.
Example of ext-Hamming (8,4) constituent decoder employed at GCN c at lth iteration.

5.1.3 Important notes on the algorithm

- The output messages from VN w to its two connected GCNs $c_j, j = 1, 2$ are the same (i.e., $V_{c_1w}^{(l)} = V_{c_2w}^{(l)}$).

- The initial incoming message at GCN c $V_{c_jw}^{(0)}(1) = y_w$, $V_{c_jw}^{(0)}(2) = 0, j = 1, 2$ as the overall demodulated binary sequence $Y = \{y_w, w = 1, 2, \cdots, N, y_w \in \{0, 1\}\}$.

- $U_{cw_i}^{(l)} = \left[U_{cw_i}^{(l)}(1)U_{cw_i}^{(l)}(2)\right]$ is the outgoing message from GCN c to VN w_i consisting of two bits representing the reliability level (one of four possible values 0^+, 0^-, 1^-, or 1^+).

- The GCN decoder does not output actual decoded words to its connected VNs. Instead, it sends a reliability signal (taking a value of four possible values) which is represented by two bits $U_{cw_i}^{(l)}(1)$ and $U_{cw_i}^{(l)}(2)$. On the other hand, the VN has to take a decision based on its incoming messages. The decision is (Flip), (Keep but as suspect) or (Keep as assumed correct).

- The function of GCN state counter $\left(\alpha_c^{(l)}\right)$ or VN state counter $\left(\gamma_w^{(l)}\right)$ is to count the number of consecutive iterations with the same state (at this GCN or VN) up to this present iteration (l), respectively.

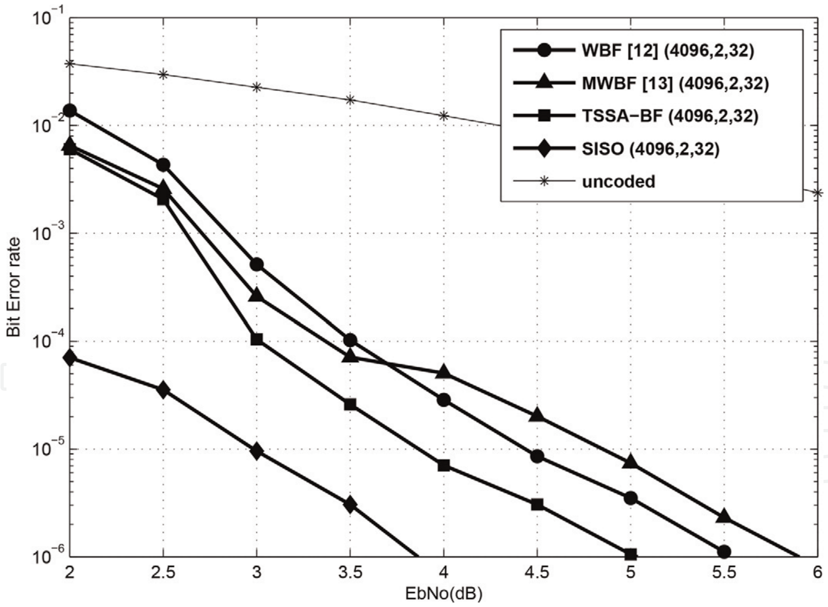


Figure 12.
Performance of GLDPC with (32,26) ext-Hamming subcodes.

Figure 12 is showing the BER performance of the ext-Hamming-based GLDPC code (with overall rate $R = 0.625$) by the TSSA-BF algorithm compared to HD decoder BF algorithms in [7, 22] and SD chase sub-decoder (number of LRPs $p = 2$). The finite-length 1B-construction-GLDPC codes of block length $N = 4096$ is used, and the maximum number of iterations (I_{max}) is set to 20. It is noted that this algorithm outperforms the other HDD ones along various values of E_b/N_o with gain not <0.5 dB at the expense of a little increase in resources used by the algorithm.

5.2 Classification-based algorithm for BF decoding with initial soft information

This algorithm is a modern bit-flipping decoding approach [24]. It is established on taking advantage of the fast BF HDD method with the help of the data extracted from the AWGN channel.

However it exploits this data at only the start phase of the decoding intentionally to make a certain classification operation. This algorithm also improves its performance by adding an additional bit in the arriving messages at VNs from CNs as a technique to enhance the decision reliability at both VNs and GCNs. The main role of this additional bit is to benefit from the subcodes states as in [23] but with a distinct fashion. This approach allows for considerable enhancement in BER at the expense of additional resources at only the side of VNs. SDD is characterized by complexity and the need to a large amount of real calculations (according to the channel soft information) all during the whole decoding procedure. However, this algorithm (accounted as HDD) needs them only in the start phase, and all processed data thereafter are hard values. Therefore this technique could reduce a significant part of the computational complexity, which was noticed in [23].

5.2.1 Algorithm description

The trapping sets, producing unrepaired errors, form the major part of the error performance letdown of the bipartite graph-based codes. The goal here is to diminish the damaging effect of most of the generated sets even by inserting supplemental resources which will be discussed below in this section.

The algorithm uses the soft channel values at the beginning of the decoding to classify the received symbols (VNs) by a predetermined threshold. Taking commonness into account and due to its simplicity, the extended Hamming code is studied inside the GLDPC codes. The extended Hamming code with increased $d_{min}(d_{min} = 4)$ is powerful and suitable for constructing standard-relevant GLDPC codes. The ext-Hamming sub-decoders may produce errors in both cases, decoding success and failure. At the GCN sub-decoder failure, it cannot locate the places of errors. At the GCN sub-decoder success, the decoder errors may emerge from two reasons: the undiscovered errors (as the received word decoded to non-transmitted valid one ($e \geq 4$), where e is the number of errors at the input of the local decoder) and the false repair ($e > 2$).

An additional bit is inserted next to the main bit in the message from the GCN to a connected VN. It acts as the decision power of GCN by raising the reliability from two to four levels. The GCN sub-decoder decodes the received word and forwards two bits (the main bit and the additional one) to every connected VN. If the main bit is seen as a decision (flip (1) or keep (0)), the additional one is seen as the decision power (strong (1) or weak (0)). The four reliability levels in descending arrangement are $0^+(01)$, $0^-(00)$, $1^-(10)$, and $1^+(11)$, corresponding to strong keep, weak keep, weak flip, and strong flip, respectively. This decoding contains two processes, the GCN processing and the VN processing, as will be explained below.

5.2.1.1 GCN processing

The three possible states, in which the ext-Hamming sub-decoder at a given GCN can be one of them, are defined as follows:

1. State 0: when the syndrome gives zero (i.e., the received sequence is a valid codeword).

2. State 1: is the case of one-error repair when the syndrome vector is one of the subcode parity-check matrix columns (i.e., error discovered and can be corrected). Therefore, it decodes to the right transmitted codeword or decodes to another valid codeword.

3. State 2: at the decoder failure (errors detected and cannot be corrected).

Using the same notations as in **Figure 8**, for any GCN c, $c = 1, 2, \cdots, M$, let $d_c^{(l)}$ be the decoder state of the cth GCN at the lth iteration. The procedures of the GCN sub-decoder of this algorithm are illustrated below:

- If $d_c^{(l)} = 0$, it sends 0^+ to all elements of the connected VNs set, $W(c) = \{w_1, w_2, .., w_i, .., w_n\}$. The message (0^+) is the highest reliability level.

- If $d_c^{(l)} = 1$, it sends 1^+ (level 1) on the place b^* (an assumed-error bit) and 0^- (level 3) to the rest of set elements $W^{'}(c)$. Elements of $W^{'}(c)$ are given 0^- (not 0^+) assuming some errors may be involved (with $e > 2$) making the decoder perform a wrong repair (an assumed-correct bit is flipped).

- If $d_c^{(l)} = 2$, it sends 1^- (level 2) to all of $W(c)$ elements as they contain a set of errors (with $e > 1$) leading to a decoder failure (errors detected with no repair capability).

Let $U^{(l)}_{c,w_i} = [U^{(l)}_{c,w_i}(1)\ U^{(l)}_{c,w_i}(2)]$ be the two bits which represent the outgoing decision message and its power, respectively, from GCN a to VN w_i. Let $V^{(l)}_{c,w_i}$ be the incoming binary bit value of the constituent codeword at c from VN w_i.

The overall set of N hard demodulated sequence bits is $\mathrm{Y} = \{y_w, w = 1, 2, \cdots, N, y_w \in \{0, 1\}\}$ as $y_w = \frac{1}{2}(\operatorname{sgn}(r_w) + 1)$ and r_w is the soft value of the wth bit in the received sequence from AWGN channel. For any VN w, the initial values $V^{(0)}_{c_j w} = y_w$ for $j = 1, 2$. Therefore at any GCN c, the initial values $V^{(0)}_{c,w_i} = y_{w_i}$ for $i = 1, 2, \cdots, n$. **Table 4** illustrates an example for ext-Hamming (8,4) constituent decoder employed at GCN c (with $n = 8$) at lth iteration.

5.2.1.2 VN processing

For any VN, it is represented by two bits. The main bit is the symbol binary value. The additional bit represents the initial reliability of the symbol value, (1) if a suspect bit or (0) if an assumed-correct bit, and the VN will use this extra information as will be discussed later. The codeword symbols which are represented by VNs are classified into two categories: most-reliable bits (MR) and least-reliable bits (LR). The classification is only initiated according to the soft information received through the channel based on a predetermined threshold.

For any transmitted codeword of length N, the wth code bit $v_w \in \{0, 1\}$ are mapped to $x_w \in \{-1, 1\}$, respectively, and transmitted over AWGN channel which is characterized by the probability density function (pdf) $p(r/x)$ given by

$$p(r_w/x_w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-(r_w - x_w)^2/2\sigma^2\right] \tag{7}$$

where σ^2 is the variance of the zero-mean Gaussian noise n_w that the channel adds to the transmitted value x_w (so that $r_w = x_w + n_w$) [25].

As illustrated in **Figure 13**, let η_o be the standard threshold on which the hard demodulator decision is based. For BPSK over AWGN channel, $\eta_o = 0$ and at this value

$$p(r/x = 1) = p(r/x = -1) \tag{8}$$

Let $\delta(r)$ be the absolute difference between the two probabilities as

$$\delta(r) = |p(r/x = 1) - p(r/x = -1)| \tag{9}$$

The received symbol is assumed to be accounted as LR bit if its value r_w gets near to the zero point or η_o ($\delta = 0$). The value of r with maximum difference δ_{max} can be

GCN State $d_c^{(l)}$	o/p messages $U^{(l)}_{c,w_i}$, $i = 1, 2, \cdots, n$							
	1	2	3	4	5	6	7	8
0	0^+	0^+	0^+	0^+	0^+	0^+	0^+	0^+
1 (with $w^* = w_3$)	0^-	0^-	1^+	0^-	0^-	0^-	0^-	v
2	1^-	1^-	1^-	1^-	1^-	1^-	1^-	1^-

Table 4.
Output messages at GCN c with ext-Hamming (8,4) decoder.

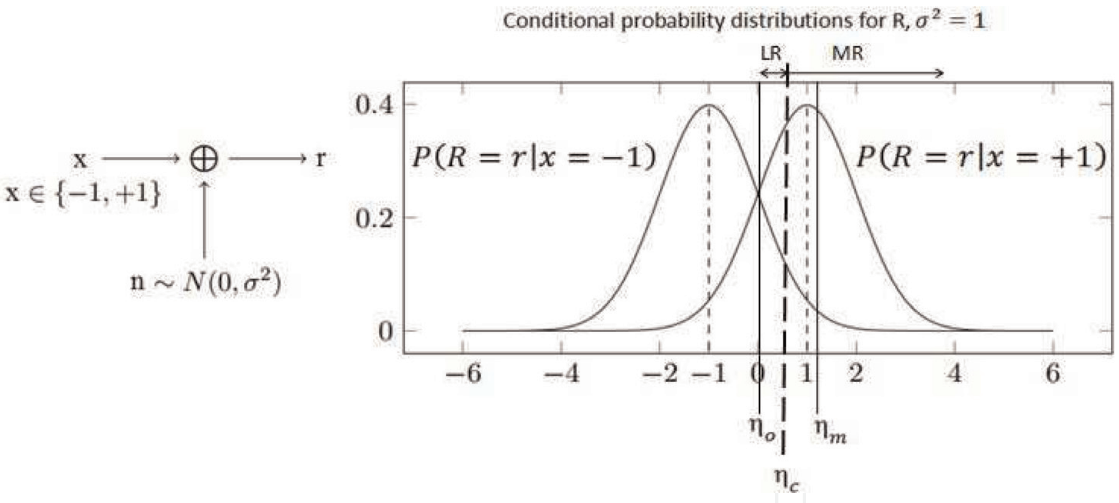


Figure 13.
The threshold η_c over AWGN channel.

taken as a second point η_m that the received symbol is accounted as MR bit if its value r_w approaches it. The absolute of the second point $|\eta_m|$ is the same for both $\pm r_w$ values as the two probability density functions are symmetric around the zero point. The equation $\delta'(r)\big|_{r=\eta_m} = 0$ should be solved to get η_m:

$$\delta = \frac{1}{\sqrt{2\pi\sigma^2}}\left[e^{-(r-1)^2/2\sigma^2} - e^{-(r+1)^2/2\sigma^2}\right] \tag{10}$$

then

$$\delta'(r)|_{r=\eta_m} = \frac{1}{\sqrt{2\pi\sigma^2}}\left[\frac{1-r}{\sigma^2}e^{-(r-1)^2/2\sigma^2} - \cdots\right. \tag{11}$$

$$\left.\frac{-(1+r)}{\sigma^2}e^{-(r+1)^2/2\sigma^2}\right] = 0 \tag{12}$$

therefore

$$(1-\eta_m)e^{\frac{\eta_m}{\sigma^2}} = -(1+\eta_m)e^{\frac{-\eta_m}{\sigma^2}} \tag{13}$$

This equation can be solved numerically by Newton-Raphson method and for various values of σ^2 (0.1–0.9), $\eta_m \approx 1.04$.

The classification threshold η_c for this algorithm is set to be in the middle between the two points. Therefore $\eta_c = \frac{\eta_o+\eta_m}{2} = \frac{0+1.04}{2} = 0.52$. Denote the initial reliability of the wth received bit as λ_w. According to the algorithm, this bit is classified as MR ($\lambda_w = 0$) if $|r_w| \geq \eta_c$; else it is classified as LR ($\lambda_w = 1$).

Using column weight of two and four levels of reliability, there are 10 probable combinations of these incoming messages to VN w, $w = 1, 2, \cdots, N$. These combinations are categorized into three subsets $\{s_i, i = 1, 2, 3\}$. For any VN w, its reliability state $g_w^{(l)}$ is set according to the arriving messages (belong to which subset). Motivated by TSSA-BF in [8] with inserting the new parameter λ_w

$$s_1 = \{(1^+, 1^+), (1^+, 1^-), (1^+, 0^-)\},$$

$$s_2 = \{(1^-, 1^-), (1^-, 0^-), (0^+, 1^+)\},$$

$$s_3 = \{(0^-, 0^-), (0^+, 1^-), (0^+, 0^-), (0^+, 0^+)\}$$

$$g_w^{(l)} = \begin{cases} 0 & if \ \left\{U_{c_j w}^{(l)}, j = 1, 2\right\} \in s_1 \\ 1 & if \ \left\{U_{c_j w}^{(l)}, j = 1, 2\right\} \in s_2 \\ 2 & if \ \left\{U_{c_j w}^{(l)}, j = 1, 2\right\} \in s_3 \end{cases}$$

- The VN w with least reliability, $g_w^{(l)} = 0$, needs flipping immediately.

- For $g_w^{(l)} = 1$, VN state counter is employed to compare the VN present in reliability state $g_w^{(l)}$ with one of the previous iterations $g_w^{(l-1)}$. Let $\gamma_w^{(l)}$ be the number of previous successive repetitions of this state $g_w^{(l)}$. If $g_w^{(l)} = 1$ with $\gamma_w^{(l)} = 2$ and, in the same time, $\lambda_w = 1$ (i.e., it is considered as LR bit), the VN will be flipped.

- Else, the VN is assumed a correct bit and kept without flipping.

By using the previously mentioned rules, the messages are renewed, and the algorithm proceeds until a zero overall syndrome output or it reaches a predefined number of iterations.

It is worth emphasizing that the GCN decoder does not output actual decoded word to its connected VNs. Instead, it sends a reliability signal (taking a value of four possible values) which is represented by two bits $U_{c,\,w_i}^{(l)}(1)$ and $U_{c,\,w_i}^{(l)}(2)$. On the other hand, the VN has to take a decision (flip or keep) based on its incoming messages and its initial reliability λ_w (MR or LR). The function of VN state counter $\gamma_w^{(l)}$ is to count a number of consecutive iterations with the same state $g_w^{(l)}$ (at this VN) up to this present iteration (l).

Figures 14–16 show the block diagrams of the overall decoder, horizontal process, and vertical process, respectively.

5.2.2 Important notes on the algorithm

- The output messages from VN w to its two connected GCNs $c_j, j = 1, 2$ are the same (i.e., $V_{c_1 w}^{(l)} = V_{c_2 w}^{(l)}$).

- The initial incoming message at GCN c $V_{c_j w}^{(0)} = y_w, j = 1, 2$ is the demodulated bit $V_w^{(0)}$ as the overall demodulated binary sequence $Y = \{y_w, w = 1, 2, \cdots, N, y_w \in \{0, 1\}\}$.

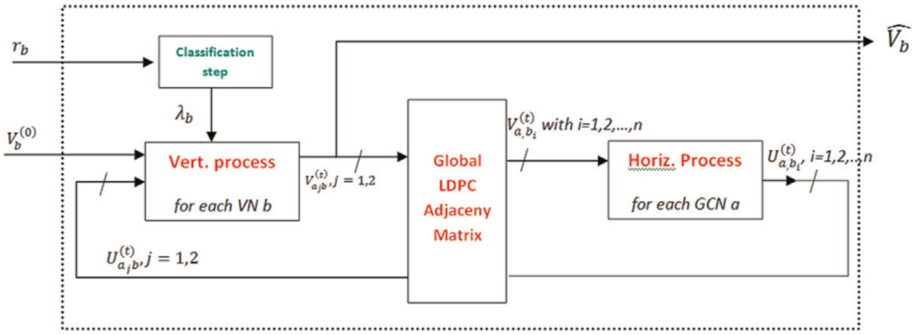

r_b
Classification step
λ_b
$V_b^{(0)}$
Vert. process
for each VN b
$V_{a_j b}^{(t)}, j = 1,2$
Global LDPC Adjaceny Matrix
$V_{a,b_i}^{(t)}$ with i=1,2,...,n
Horiz. Process
for each GCN a
$U_{a,b_i}^{(t)}$, i=1,2,...,n
$U_{a_j b}^{(t)}, j = 1,2$
$\widehat{V_b}$

Figure 14.
Block diagram of the overall classification-based decoder.

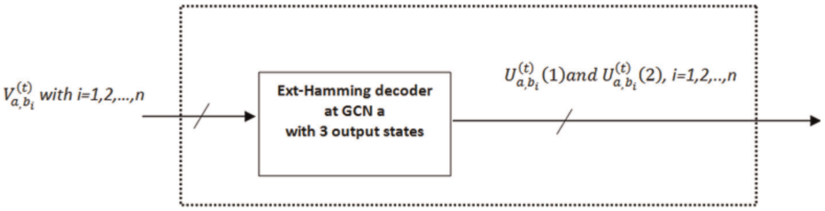

$V_{a,b_i}^{(t)}$ with i=1,2,...,n
Ext-Hamming decoder at GCN a with 3 output states
$U_{a,b_i}^{(t)}(1)$and $U_{a,b_i}^{(t)}(2)$, i=1,2,..,n

Figure 15.
Block diagram of the horizontal process of the classification-based decoder.

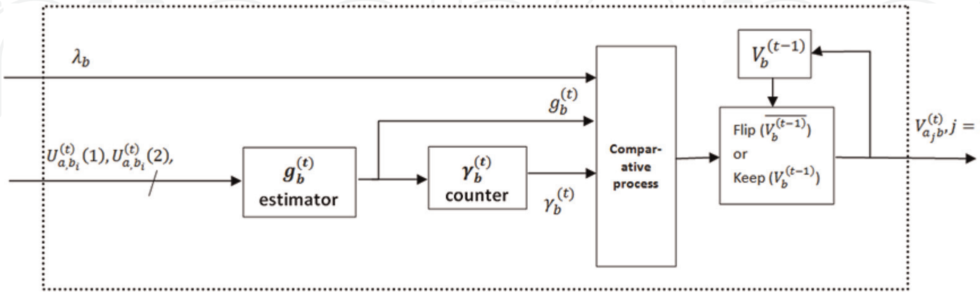

λ_b
$U_{a,b_i}^{(t)}(1), U_{a,b_i}^{(t)}(2),$
$g_b^{(t)}$ estimator
$\gamma_b^{(t)}$ counter
$g_b^{(t)}$
$\gamma_b^{(t)}$
Compar-ative process
$V_b^{(t-1)}$
Flip ($\overline{V_b^{(t-1)}}$) or Keep ($V_b^{(t-1)}$)
$V_{a_j b}^{(t)}, j = 1,2$

Figure 16.
Block diagram of the vertical process of the classification-based decoder.

- $U_{cw_i}^{(l)} = \left[U_{cw_i}^{(l)}(1)U_{cw_i}^{(l)}(2)\right]$ is the outgoing message from GCN c to VN w_i consisting of two bits representing the reliability level (one of four possible values 0^+, 0^-, 1^-, or 1^+). On the other hand, $V_{c_j w}^{(l)}$ is the outgoing message from VN w to GCN c_j containing just the decoded symbol bit value.

- The GCN decoder does not output actual decoded words to its connected VNs. Instead, it sends a reliability signal (taking a value of four possible values) which is represented by two bits $U_{cw_i}^{(l)}(1)$ and $U_{cw_i}^{(l)}(2)$. On the other hand, the VN has to take a decision based on its incoming messages. The decision is flip or keep.

- The function of VN state counter $\left(\gamma_w^{(l)}\right)$ is to count the number of consecutive iterations with the same state (at this VN) up to this present iteration (l).

- The algorithm manages without the greater portion of the overhead of the algorithm in [23] which was specially located in the horizontal (GCN) process.

Figure 17 shows the GLDPC BER performance using the (32,26,4) extended Hamming subcode by this decoding with respect to the bit-flipping algorithms in [7, 22] and [23]. It is noticed that this algorithm surpasses the other ones at the cost of a slight increase in computational complexity resulting from the comparison operations made at the initial classification step. It is also noticed that as N increases, a slow improvement in the performance is achieved.

The predefined number of iterations (20) is found to be very sufficient for good performance as the additional iterations beyond this limit have no considerable difference in the performance and latency in the decoding process which should be avoided for fast decoding purposes.

Not similar to the conventional GLDPC HDD BF decoding, the received sequence soft values are utilized to make appropriate classification of the received bits (variable nodes) into two classes.

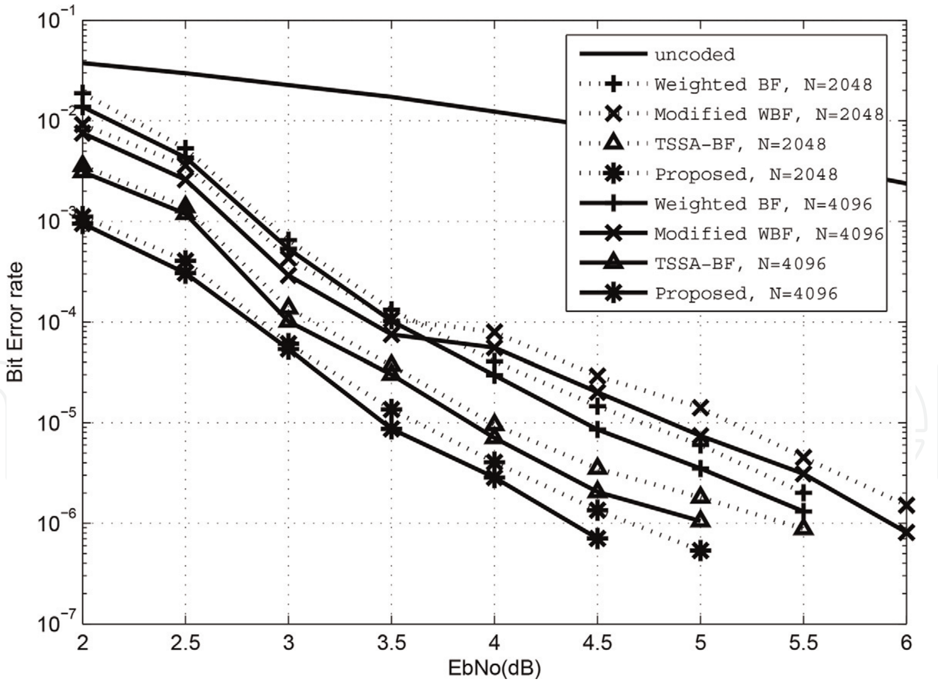


Figure 17.
Simulated BER curves of $(N, 2, 32)$ *GLDPC codes.*

The algorithm not only achieves a better error performance but also requires less iterations than the other competent algorithms. In terms of the impacts of the soft information (from AWGN channel) on coding gain of the GLDPC, the algorithm is revealed to exhibit considerable performance to decoder complexity trade-off. The algorithm can be adapted to handle generalized and more robust subcodes with the capability to correct more errors to improve the performance.

As discussed in [24], the computational complexity is provided in terms of the average number of executed operations for this algorithm against the rather comparable TSSA-BF algorithm. It is noticed that the complexity of this decoder is reduced by more than 60%.

6. Simplified SDD algorithm over AWGN channels

The algorithm, in [26], serves to use the chase SD decoders as minimum as possible to lower their complexity and expedites the decoding procedures. This algorithm is a variant approach from a previous one by [27] to lower the complexity of turbo product code (TPC) with multiple-error correction BCH subcodes. The chase decoder at every row or column input sequence in the product code was used as it attempts to decrease the HDD operations executed on the test patterns (TPs) produced in the chase decoder. The algorithm, explained below, will benefit from the algorithm in [27] for more reduction in the complexity.

The introduced algorithm is the MP method for decoding GLDPC with the chase-II algorithm operated as a posteriori probability decoding on GCNs. It will use extended double-error BCH (with high error-correcting capability) as a subcode to obtain a better performance. For simplicity, all GCNs are represented with the same eBCH code of parameters (n, k, d). The overall block diagram of this algorithm is depicted in **Figure 18**.

As discussed before in Section 3 and using the same notations and considerations, the soft-output value of every decoded symbol of the subcode should be calculated, by Eqs. (3) or (4), to be sent back on the connected edges to the GLDPC

VNs as followed by the MP algorithm in an iterative method to obtain a final decision after a certain number of iterations or a syndrome condition should be satisfied.

W.r.t any given GCN c, the chase-II-based SISO decoder produces 2^p TPs by making a perturbation of the p LRPs in the demodulated word with length n (subcode word). Therefore 2^p HDDs should be performed to obtain a decided codeword. Therefore algorithm will get the syndromes of Y_c.

For extended BCH2 (double-error correction), the algorithm computes two syndromes S_c^1 and S_c^3 as follows:

$$\begin{aligned} S_c^1 &= y_{c2} \oplus y_{c3}x \oplus \cdots \oplus y_{cn-1}x^{n-3} \oplus y_{cn}x^{n-2}\big|_{x=\alpha}, \\ S_c^3 &= y_{c2} \oplus y_{c3}x \oplus \cdots \oplus y_{cn-1}x^{n-3} \oplus y_{cn}x^{n-2}\big|_{x=\alpha^3} \end{aligned} \tag{14}$$

where α is the primitive element of GF(2^m) that generates the BCH code polynomial.

According to the values of the syndromes as illustrated in **Table 5**, the algorithm estimates the number of errors contained in the sequence.

If there are no errors $(e = 0)$, it is likely (with high percentage) that the demodulated word is the valid transmitted one and the decoder will not do its task. If $0 < e \leq 2$, the algorithm may execute the HDD (Berlekamp-Massey algorithm) and outputs the decoding decision. In these two preceding cases, the soft-output values can be estimated as the decision is highly probable to be correct as follows:

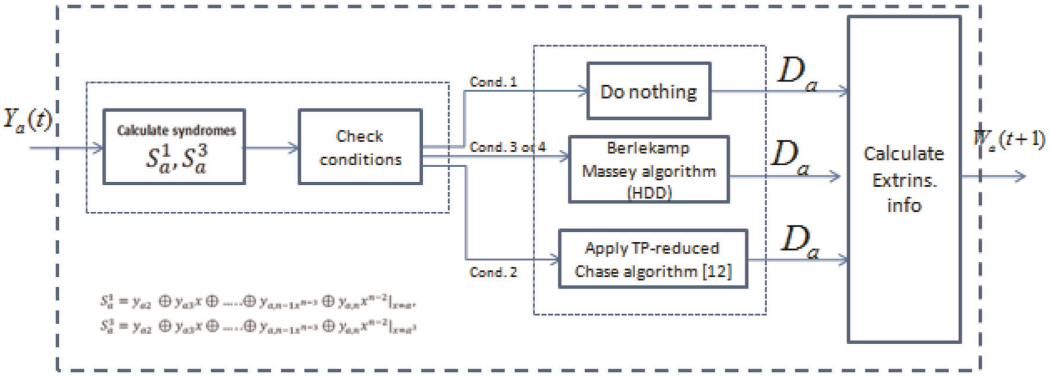

$Y_a(t)$
Calculate syndromes
S_a^1, S_a^3
Check conditions
Cond. 1
Cond. 3 or 4
Cond. 2
Do nothing
Berlekamp Massey algorithm (HDD)
Apply TP-reduced Chase algorithm [12]
D_a
D_a
D_a
Calculate Extrins. info
$Y_a(t+1)$
$S_a^1 = y_{a2} \oplus y_{a3}x \oplus \ldots \oplus y_{a,n-1}x^{n-3} \oplus y_{a,n}x^{n-2}|_{x=\alpha}$,
$S_a^3 = y_{a2} \oplus y_{a3}x \oplus \ldots \oplus y_{a,n-1}x^{n-3} \oplus y_{a,n}x^{n-2}|_{x=\alpha^3}$

Figure 18.
The block diagram of the lowered-complexity chase-based decoding algorithm.

Syndrome of demodulated vector $S_a = \boldsymbol{Y_a}.\boldsymbol{H_{bch}}$	Number of contained errors (e)	Action taken (method of estimating the decision Da)	Calculation model of extrinsic information $r'_{a,i}$
$S_a^1 = S_a^3 = 0$	0	Nothing (the demodulated vector is the decision codeword)	$r'_{a,i} = \beta' x d_i$
$S_a^1 = 0, S_a^3 \neq 0$	>2	Apply TP-reduced chase algorithm [12]	Pyndiah model [15]
$(S_a^1) \oplus S_a^3 = 0$	1	Apply HDD (Berlekamp-Massey algorithm)	$r'_{a,i} = \beta' x d_i$
Else	2	Apply HDD (Berlekamp-Massey algorithm)	$r'_{a,i} = \beta' x d_i$

Table 5.
The actions of the proposed algorithm.

$$r'_{c,i} = \beta \times d_{c,i} \text{ with } \beta \geq 0$$

where β is chosen to be evolved with the decoding iterations, $\beta(l) = [0.4, 0.6, 0.8, 1, 1, 1, \cdots]$.

If $e > 2$, the chase algorithm is needed to extract a decision codeword but will not decode a complete list of 2^p test patterns (TPs). The proposed algorithm in this case will benefit from the lowered-complexity TP-reduced algorithm in [27]. The amount of reduction in HDDs of the algorithm compared to the standard one is listed in **Table 6**.

The computational complexity of this algorithm is estimated by the number of hard decision decoding processes (Berlekamp-Massey algorithm) employed at GCNs. The (64,51) eBCH subcode is chosen with double-error correction capability to exploit the multiple calculated syndroms, while keeping a moderate code rate ($R \cong 0.6$). Therefore, for keeping this rate, only GLDPC codes with column weight ($j = 2$) are considered. As shown in **Figure 19**, the number of HDDs in the decoder is calculated for two numbers of LRPs ($p = 3, p = 4$) and up to five iterations

BCH code (n,k)	N	J	Code rate	No. of LRPs (p)	Number of HDDs in standard alg. [7]	Avg. number of HDDs in proposed alg.	Percentage of complexity reduction (%)
eBCH2 (64,51)	4096	2	0.6	3	5120	3072	60
eBCH2 (64,51)	4096	2	0.6	4	10,240	5734	56
eBCH3 (64,45)	4096	2	0.41	4	10,240	5232	51.1
e8CH3 (64,45)	4096	2	0.41	5	20,480	9011	44

Table 6.
The reduction of HDDs in the lowered-complexity chase-based decoding algorithm (SNR = 2 dB at $I_{max} = 5$).

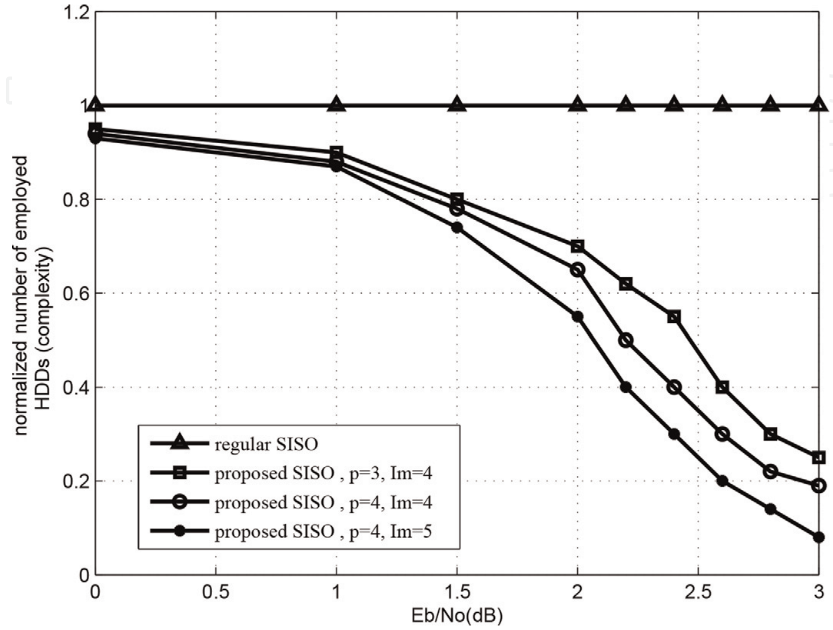


Figure 19.
Comparison of computational complexity of less-complex and conventional SISO decoding algorithms for decoding (64,51) eBCH-based GLDPC codes with length N = 4096 for various values of p and I_m.

$(I_m = 5)$. For clarification, the number of HDDs is normalized to the one of the conventional chase decoder. It is shown that a considerable reduction occurs especially after $E_b/N_o = 2$ dB.

The results show a significant lowering in the soft decoding operations executed at GCNs compared to conventional chase decoders with little wastage in the BER performance. This scheme is highly required in low error rate applications such as optical communication systems.

Author details

Sherif Elsanadily
EAEAT, Ministry of Military Production, Cairo, Egypt

*Address all correspondence to: sherif.elsanadily@eaeat.edu.eg

References

[1] Tanner R. A recursive approach to low complexity codes. IEEE Transactions on Information Theory. 1981;**27**:533-547

[2] Lentmaier M, Zigangirov K. On generalized low-density parity check codes based on Hamming component codes. IEEE Communications Letters. 1999;**3**:248-250

[3] Liva G, Ryan W, Chiani M. Quasi-cyclic generalized LDPC codes with low error floors. IEEE Transactions on Communications. 2008;**56**:49-57

[4] Wang Y, Fossorier M. Doubly generalized LDPC codes over the AWGN channel. IEEE Transactions on Communications. 2009; **57**:1312-1319

[5] Bahl L, Cocke J, Jelinek F, Raviv J. Optimal decoding of linear codes for minimizing symbol error rate. IEEE Transactions on Information Theory. 1974;**20**:284-287

[6] Boutros J, Pothier O, Zemor G. Generalized low density (tanner) codes. In: Proc. IEEE. Int. Conf. Commun.; 1999; vol. 1, pp. 441-445

[7] Miladinovic N, Fossorier M. Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels. In: Proc. IEEE Global Telecommun. Conf.; St. Louis, MO; 2005; vol. 3; p. 6

[8] Chen J, Tanner RM. A hybrid coding scheme for the Gilbert-Elliott channel. IEEE Transactions on Communications. 2006;**54**:1787-1796

[9] Abu-Surra S, Liva G, Ryan WE. Low-floor Tanner codes via Hamming-node or RSCC-node doping. In: Proc. the 16th international conf. on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes; 2006; pp. 245-254

[10] Yue G, Ping L, Wang X. Generalized low-density parity-check codes based on Hadamard constraints. IEEE Transactions on Information Theory. 2007;**53**:1058-1079

[11] Hirst S, Honary B. Application of efficient chase algorithm in decoding of generalized low-density parity-check codes. IEEE Communications Letters. 2002;**6**:385-387

[12] Djordjevic I, Milenkovic O, Vasic B. Generalized low-density parity-check codes for optical communication systems. IEEE: Journal of Lightwave Technology. 2005;**23**:1939-1946

[13] Guan R, Zhang L. Hybrid Hamming GLDPC codes over the binary erasure channel. In: The 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID); 2017; pp. 130-133

[14] Olmos PM, Mitchell DGM, Costello DJ. Analyzing the finite-length performance of generalized ldpc codes. In: 2015 IEEE International Symposium on Information Theory (ISIT); 2015; pp. 2683-2687

[15] Yu Y, Han Y, Zhang L. Hamming-GLDPC codes in BEC and AWGN channel. In: The 6th International Conference on Wireless, Mobile and Multi-Media (ICWMMN 2015); 2015; pp. 103-106

[16] Beemer A, Habib S, Kelley CA, Kliewer J. A generalized algebraic approach to optimizing SC-LDPC codes. In: The 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton); 2017; pp. 672-679

[17] Pothier O, Brunel L, Boutros J. A low complexity FEC scheme based on the intersection of interleaved block codes.

In: IEEE 49th Vehicular Technology Conf. (VTC 99); Houston; 1999; vol. 1; pp. 274-278

[18] Chase D. A class of algorithms for decoding block codes with channel measurement information. IEEE Transactions on Information Theory. 1972;**18**:170-182

[19] Pyndiah RM. Near-optimum decoding of product codes: Block turbo codes. IEEE Transactions on Communications. 1998;**46**:1003-1010

[20] Pyndiah R, Glavieux A, Picart A, Jacq S. Near optimum decoding of products codes. In: Proc. IEEE GLOBECOM94 Conf., vol. 1/3; San Francisco, CA; 1994; pp. 339-343

[21] Hirst S, Honary B, Markarian G. Fast chase algorithm with application in turbo decoding. IEEE Transactions on Communications. 2001;**49**:1693-1699

[22] Hirst S, Honary B. Decoding of generalized low-density paritycheck codes using weighted bit-flip voting. IEEE Proceedings Communications. 2002;**149**:1-5

[23] Elsanadily S, Mahran A, Elghandour O. Two-side state-aided bit-flipping decoding of generalized low density parity check codes. IEEE Communications Letters. 2017;**21**: 2122-2125

[24] Elsanadily S, Mahran A, Elghandour O. Classification-based algorithm for bit-flipping decoding of GLDPC codes over AWGN channels. IEEE Communications Letters. 2018;**22**: 1520-1523

[25] Ryan WE, Lin S. Channel Codes: Classical and Modern. New York: Cambridge University Press; 2009

[26] Elsanadily S, Mahran A, Elghandour O. Lowered-complexity soft decoding of generalized LDPC codes over AWGN channels. In: The IEEE 2017 12th International Conference on Computer Engineering and Systems (ICCES); 2017; pp. 320-324

[27] Chen GT, Cao L, Yu L, Chen CW. Test-pattern-reduced decoding for turbo product codes with multi-error-correcting EBCH codes. IEEE Transactions on Communications. 2009; **57**:307-310