Low Complexity Decoding Scheme for LDPC Codes Based on Belief Propagation Algorithm

Wenshuo Zhang, Liming Zheng, Yue Wu(✉), Gang Wang, and Aijun Liu

Communication Research Center, Harbin Institute of Technology, Harbin, China
17S005039@stu.hit.edu.cn, {zheng,wuy,gwang51}@hit.edu.cn,
hitlaj@163.com

Abstract. The low-density parity check codes (LDPC codes) are block codes whose performances are close to the Shannon limit. LDPC codes have the strong ability for error correction. The decoding algorithm of LDPC codes has a great influence on their performances. The belief propagation (BP) algorithm is a commonly used soft decision decoding algorithm. The algorithm decodes by information iterations, and its complexity does not increase rapidly with the increase of code length. This paper mainly analyze the probabilistic domain BP decoding algorithm, log-domain BP decoding algorithm and minimum sum decoding algorithm, the bit error performance of LDPC codes under BP algorithm is studied, and the influence of the number of iterations on the BP decoding algorithm is also shown by simulation results.

Keywords: LDPC codes · Soft decision decoding algorithm · BP algorithm

1 Introduction

Robert Gallager discussed error correction codes based on low-density parity check codes (LDPC codes) in 1962 [1]. In 1990's, LDPC codes were rediscovered by MacKay and Neal [2]. LDPC codes are linear block codes whose performances are close to the Shannon limit [3]. The LDPC codes are low-density parity check codes, and they are constructed according to a low-density sparse check matrix or a tanner graph. A low-density check code of length n corresponds to the sparse check matrix $\mathbf{H}$ and represented by $\mathbf{H}$(n,p,q). There are q non-zero elements in each row of $\mathbf{H}$ and p non-zero elements in each column. Both p and q are very small compared to m and n, which makes the number of non-zero elements in the check matrix is much smaller than the number of zero elements.

The low-density sparse check matrix can be constructed by Gallager construction method [1] and progressive edge-growth (PEG) method [4], The PEG construction method establishes an edge link between a information node and

M. Jia et al. (Eds.): WiSATS 2019, LNICST 280, pp. 727–737, 2019.
https://doi.org/10.1007/978-3-030-19153-5_71

a check node according to the edge-by-edge method [5]. MacKay and Neal have showed LDPC codes can have very good performances when decoded with the belief-propagation (BP) algorithm [2]. Linear functions make the implementation of the BP algorithm difficult, so the BP algorithm is simplified in BP-based algorithm [6], it can reduce the complexity but it makes decoding performance decline. Among all the BP-based algorithms, LLR BP decoding algorithm [7] improve LDPC codes performance greatly. The UMP_BP_based (minimum and) algorithm reduce the computational complexity to a great extent.

This paper is organized as follows. Section 2 corresponds to general construction methods of the low-density sparse check matrix. Section 3 describes the BP-Based algorithm and its different variants. Simulation results are proposed in Sect. 4. Finally, the conclusion and perspectives of this paper are given in Sect. 5.

2 Construction Methods of Low-Density Sparse Check Matrix

Gallager Construction Method. The Gallager construction method is a construction method given by Gallager when the LDPC codes are proposed [1]. The basic idea is to give the row weight d_v and column weight d_c of the $\mathbf{H}$ matrix and make it satisfy the constraint $d_c \geq 3$. The $\mathbf{H}$ matrix is divided into sub-matrices by row, and the sub-matrices have only one non-zero element per column. In the first sub-matrix, non-zero elements in row i start from column $(i-1)d_v + 1$ to column id_v and the rest elements are zero elements. Except for the first sub-matrix, the remaining sub-matrices are obtained by column transformation of the first matrix.

The following matrix (1) is a typical Gallager check matrix $\mathbf{H}(20, 3, 4)$. It can be seen that in the first five rows, that is, In the first sub-matrix, the elements in row i are non-zero elements form column $(i-1)d_v + 1$ to column id_v and the rest elements are zero elements. The remaining two sub-matrices are all obtained from the first sub-matrix by switching the column randomly.

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

PEG Construction Method. The following matrix (2) is a **H** matrix in binary domain. And its corresponding tenner graph is shown in Fig. 1.

$$\mathbf{H} = \begin{bmatrix} 1\,1\,1\,0\,0\,1\,1\,0\,0\,0\,1\,0 \\ 1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,1 \\ 1\,0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,1 \\ 0\,1\,0\,1\,1\,0\,1\,1\,1\,0\,0\,0 \\ 0\,0\,1\,0\,1\,1\,0\,0\,1\,1\,1\,0 \end{bmatrix} \tag{2}$$

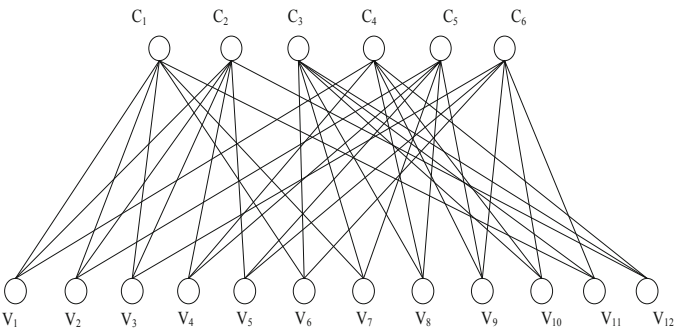

Fig. 1. The tanner graph corresponding to check matrix in (2).

The PEG construction method establishes an edge link between an information node and a check node according to the edge-by-edge method. When a new edge is added, it maximum the girth of the graph. The degree sequence of the information nodes is defined by the following Eq. (3).

$$D_b = (d_{b0}, d_{b1}, \cdots d_{bn-1}) \tag{3}$$

Where d_{bi} is the degree of the ith information node, it satisfies the non-decreasing order $d_{b0} \leq d_{b1} \leq \cdots \leq d_{bm-1}$. From the point of view, the information node set is V_{b}, the set of edges is $E = E_{b0} \cup E_{b1} \cup \cdots \cup E_{bn-1}$, E_{b_i} corresponding to the information nodes b_i $(0 \leq \mathrm{i} \leq n-1)$. The kth edge connected to the information nodes b_i is defined as $E^k_{\mathrm{b}_i}$ $(0 \leq \mathrm{k} \leq db_i - 1)$. If nodes x and y are connected, then (x, y) is an edge, and the set of nodes connected to node x is called the neighbors of x. According to the tanner graph, for a given information node b_i, a tree graph with a depth of l is expanded along the information node b_i. The set of all the check nodes included at this time is called the neighbor of the information nodes b_i with l depth, and is represented as $N^l_{b_{\mathrm{i}}}$. It means that we start from node b_i then go through all the edges connected to the check nodes, removing the edges that we have passed until we get the desired depth. There may be some nodes and edges appearing multiple times in this process.

The process of setting the edge should satisfy the requirement that the newly introduced edge has the smallest effect on the girth of the graph. Therefore, the key is to find the check node that is the farthest from this information node and then set a new edge between them. The PEG algorithm is summarized as follows.

Algorithm for PEG

For i=0 to n-1
 For k=0 to n-1
  If k=0
   Then the edge $(b_i, c_j) \rightarrow E^0_{\mathrm{b}_i}$, where $E^0_{\mathrm{b}_i}$ is the first incident edge of the information nodes b_i, c_j is the check node with the lowest degree in the current graph set $E_{b_0} \cup E_{b_1} \cup ... \cup E_{b_{i-1}}$.
  Else
   Based on the current set of graphs, expand the information nodes b_i to a sub map of depth l until the number of elements in the collection $N^l_{b_i}$ reaches m, or $\overline{N}^l_{b_i} \neq \varphi$ but $\overline{N}^{l+1}_{b_i} = \varphi$. Then, there exists $(\mathrm{b_i}, c_j) \rightarrow E^0_{\mathrm{b_k}}$, where $E^0_{\mathrm{b_k}}$ is the kth incident edge of the information nodes b_i, which c_j is the check node with the lowest degree in the set $\overline{N}^l_{b_i}$.
  End If
 End
End

According to the algorithm above, The first layer of the sub-graph whose depth is l of Fig. 1 is shown in Fig. 2.

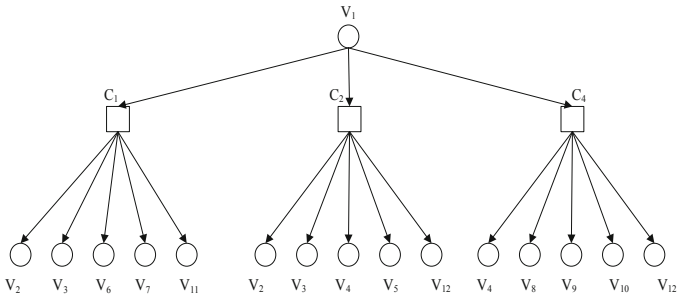

Fig. 2. The first layer of Fig. 1.

3 Decoding Methods of LDPC Codes

There are mainly two types of decoding algorithms for LDPC codes, mainly based on hard decision decoding and soft decision decoding. The advantage of

decoding based on hard decision is mainly that the amount of computation is small [8]. The soft decision uses posterior probability decoding, which can achieve good performance. The decoding algorithm based on BP is the most popular decoding scheme, mainly including probability BP and LLR BP. The basic idea is to process the check nodes and information nodes for each iteration. Figure 3 shows various decoding algorithms based on BP.

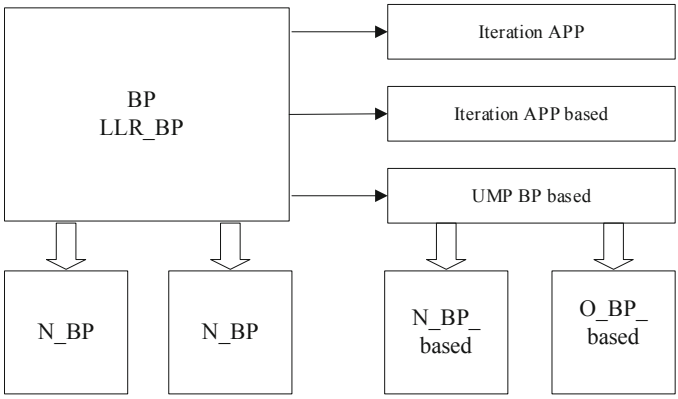

Fig. 3. Decoding algorithms based on BP.

Probabilistic BP Algorithm. Each modulated codeword $C = (c_1, c_2, ..., c_n)$ is mapped to a transmission sequence $X = (x_1, x_2, ..., x_n)$, and the received sequence is $Y = (y_1, y_2, ..., y_n)$.

The external probability that check node j passes to information node i, under the condition of given information bits and other information bits with independent probability distribution, is the probability that check equation j is met. It is represented as $r_{ji}(b)(b = 0, 1)$. The external probability information passed from information node i to check node j, that is, after obtaining the external information of channels and all the check nodes except j, the probability of the information nodes that is judged as $c_i = b$ and is represented as $q_{ij}(b)$. The set of the check nodes connected to the information nodes i is represented as $C(i)$. A collection of connected information nodes by the check node j is represented by $R(j)$. A set of check nodes connected to the information nodes i except j is represented as $C(i)\backslash j$. A set of information nodes connected to the check node j except i is represented as $R(j)\backslash i$. c_{kj} is the kth bit in the included jth check equation. y_{kj} is the corresponding received value to the c_{kj}, the bit in $\hat{c}$ satisfies the d_c check equation which include c_i is represented as S_i. The posterior probability of the transmitted bit (or information node) which is determined as $c_i = 1$ after receiving y_i is P_i. The posterior probability to judge the kth bit that is included in the jth check equation which contains c_i as $c_{kj} = 1$ after receiving y_{kj} is P_{kj}.

$$P_i = \Pr(c_i = 1|y_i) \tag{4}$$

$$P_{kj} = \Pr(c_{kj} = 1|y_{kj}) \tag{5}$$

For a binary sequence $\alpha = (a_1, a_2, ..., a_{\mathrm{m}})$, where $p(a_k = 1) = p_k$, the probability that α has an even number of 1 and an odd number of 1 are as follows (6) and (7) respectively [8].

$$1/2 + 1/2 \prod_{k=1}^{m} (1 - 2p_k) \tag{6}$$

$$1/2 - 1/2 \prod_{k=1}^{m} (1 - 2p_k) \tag{7}$$

According to Gallager's theory [1] we know that,

$$r_{ji}(0) = 1/2 + 1/2 \prod_{i' \in R_j \backslash i}^{m} (1 - 2p_{i'j}) \tag{8}$$

$$r_{ji}(1) = 1/2 - 1/2 \prod_{i' \in R_j \backslash i}^{m} (1 - 2p_{i'j}) \tag{9}$$

Gallager's theory gives the calculation method of posterior probability when the bit rates are independent to each other. The Gallager's theory can be rewritten as,

$$\frac{P(c_i = 0|\bar{y}, S_i)}{P(c_i = 1|\bar{y}, S_i)} = \frac{1 - P_i}{P_i} \frac{\prod_{j \in C_i} r_{ji}(0)}{\prod_{j \in C_i} r_{ji}(1)} \tag{10}$$

$$q_{ij}(0) = (1 - P) \prod_{j' \in C_i \backslash j} r_{j'i}(0) q_{ij}(1) = P_i \prod_{j' \in C_i \backslash j} r_{j'i}(1) \tag{11}$$

Therefore, the BP decoding step is summarized as follows,

Initialization. Calculate the initial probability $P_i(1)$, $P_i(0) = 1 - P_i(1)(i = 1, 2, 3, ..., n)$ that the channel passes to the information nodes. Then, for each information node i and the adjacent check node $j \in C(i)$, an initial message from the information nodes to the check node is set as,

$$q_{ij}^{(0)}(0) = P_i(0) \tag{12}$$

$$q_{ij}^{(0)}(1) = P_i(1) \tag{13}$$

Iterative Processing. Step 1: Information processing of check nodes

For check node j and its adjacent information nodes $i \in R(j)$, in the first iteration, the message that the information nodes passes to the check nodes is calculated.

$$\begin{cases} r_{ji}^{l}(0) = 1/2 + 1/2 \prod\limits_{i' \in R_j \backslash i} (1 - 2q_{i'j}^{(l-1)}(1)) \\ r_{ji}^{l}(1) = 1/2 - 1/2 \prod\limits_{i' \in R_j \backslash i} (1 - 2q_{i'j}^{(l-1)}(1)) \end{cases} \tag{14}$$

Step 2: Information processing of information nodes

For information nodes i and its neighboring check nodes $j \in C(i)$, the message that the check nodes passes to the information nodes is calculated.

$$\begin{cases} q_{ij}^{l}(0) = k_{ij} P_i(0) \prod\limits_{j' \in C_j \backslash j} r_{j'i}^{(l)}(0) \\ q_{ij}^{l}(1) = k_{ij} P_i(1) \prod\limits_{j' \in C_j \backslash j} r_{j'i}^{(l)}(1) \end{cases} \tag{15}$$

k_{ij} is the correction factor that makes $q_{ij}^{(l)}(0) + q_{ij}^{(l)}(1) = 1$.

Step 3: Decoding decision

Hard decision messages are computed for all information nodes.

$$\begin{cases} q_{i}^{l}(0) = k_i P_i(0) \prod\limits_{j \in C_j} r_{ji}^{(l)}(0) \\ q_{i}^{l}(1) = k_i P_i(1) \prod\limits_{j \in C_j} r_{ji}^{(l)}(1) \end{cases} \tag{16}$$

Where k_i is the correction factor, so that $q_i^{(l)}(0) + q_i^{(l)}(1) = 1$. If $q_i^{(l)}(1) > q_i^{(l)}(0)$, then, $\hat{c}_i = 1$, otherwise $\hat{c}_i = 0$.

Stop. If the maximum number of iterations is reached or $\mathbf{H}\hat{\mathbf{c}}^T = 0$, the operation ends, otherwise the iteration is continued from step 1.

LLR BP Algorithm. When the probability BP algorithm is represented by a likelihood ratio the LLR BP [4] algorithm is obtained, and the multiplication operation is converted into an additional operation to reduce the operation time. The likelihood is defined as follows. The channel initial message is defined by the following Eq. (17).

$$L(P_i) = \ln \frac{P_i(0)}{P_i(1)} = \ln \frac{P_r\{x_i = 1 | y_i\}}{P_r\{x_i = -1 | y_i\}} \tag{17}$$

The message that the check node passes to the information node is defined as the following Eq. (18).

$$Lr_{ji} = \ln \frac{r_{ji}(0)}{r_{ji}(1)} \tag{18}$$

The information that the information node passes to the check node is calculated in (19).

$$Lq_{ij} = \ln \frac{q_{ij}(0)}{q_{ij}(1)} \tag{19}$$

All messages collected by the information nodes by (20).

$$Lq_i = \ln \frac{q_i(0)}{q_i(1)} \tag{20}$$

Then check nodes message processing can be written as formula (21).

$$1-2r_{ij}(1) = 1 + \prod_{i' \varepsilon R_j \backslash i} (1 - 2q_{i'j}(1)) \tag{21}$$

Information nodes message processing can be written as formula (22).

$$L(q_{ij}) = L(P_i) + \prod_{j' \in C_i \backslash j} L(r_{j'i}) \tag{22}$$

Therefore, the LLR BP algorithm is summarized as follows,

Initialization. Calculate the initial probability likelihood ratio $L(P_i)$ passed from the channel to the information nodes. Then, for each information node i and the adjacent check node $j \in C(i)$, the initial message that the information node passes to the check node is (23).

$$L^{(0)}(q_{ij}) = L(P_i) \tag{23}$$

Iterative Processing. Step 1: Information processing of Check nodes

For check node j and its adjacent information nodes $i \in R(j)$, in the first iteration, the message that the information nodes passes to the check node is calculated.

$$L^{(l)}(r_{ji}) = 2\tanh^{-1}(\prod_{i' \in R_j \backslash i} \tanh(1/2L^{(l-1)}(q_{i'j}))) \tag{24}$$

Step 2: Information processing of information nodes

For information node i and its neighboring check node $j \in C(i)$, a message is sent from the check node to the information note.

$$L^{(l)}(q_{ij}) = l(P_i) + \prod_{j' \in C_i \backslash j} L^{(l)}(r_{j'i}) \tag{25}$$

Step 3: Decoding decision

Hard decision messages are computed for all information nodes.

$$L^{(l)}(q_i) = l(P_i) + \prod_{j' \in C_i} L^{(l)}(r_{ji}) \tag{26}$$

If $L^{(l)}(q_i) > 0$, then $\hat{c}_i = 0$, otherwise $\hat{c}_i = 1$.

Stop. If the maximum number of iterations is reached or $\mathbf{H}\hat{\mathbf{c}}^T = 0$, the operation ends, otherwise the iteration is continued from step 1.

If the nature of the utilization $\tanh x, \tanh^{-1} x$ is used, the formula of the check node processing in the LLR BP algorithm is processed, and the Eq. (27) can be obtained.

$$L(r_{ji}) = \prod_{i' \in R_j \setminus i} \operatorname{sgn}(L(q_{ij})). \min(|L(q_{i'j})|) \tag{27}$$

If it is processed by the above formula, it is called the minimum sum or maximum product algorithm. This processing can make the iteration of the check nodes much more simple.

4 LDPC Codes Simulation Results and Performance Analysis

When a low-density parity check matrix **H** is choosen, we decode the (2048,1024) LDPC codes recommended by CCSDS standard with minimum sum decoding algorithm the simulation result shows the bit error rate performance in Figs. 4 and 5.

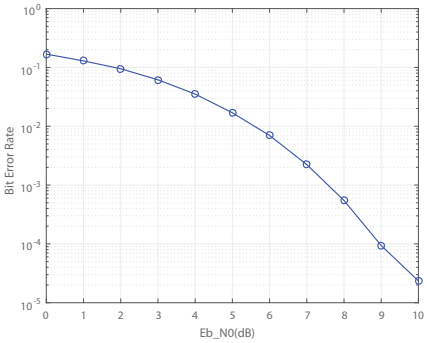

Fig. 4. BER performance with minimum sum decoding algorithm (iteration number = 1)

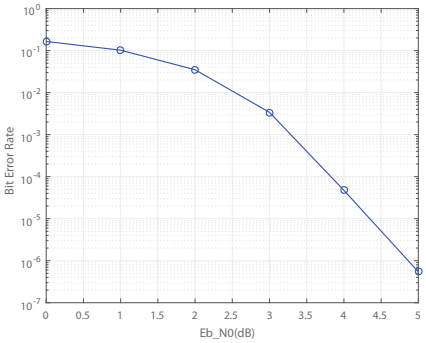

Fig. 5. BER performance with minimum sum decoding algorithm (iteration number = 5)

Under the condition that the code length is fixed, as the signal-to-noise ratio increases, the bit error rate decreases. In the case of iteration number is 1, the LDPC codes performance can reach about 10^{-5} at 10 dB. In the case of iteration number is 5, the LDPC codes performance can reach about 10^{-6} at 5 dB. It seems the iteration number has significant impact on bit error rate performance of LDPC codes. So the influence of number of iterations is studied in Fig. 6. Simulation result in Fig. 6 shows that the BP decoding algorithm can receive

better performance with the increasing number of iterations. Under the same signal-to-noise ratio condition, the bit error rate decreases with the increase of the number of iterations. When the number of iterations is 1, 5, 10, 15, the bit error rate increases with the number of iterations. The degradation is very fast, and the decoding performance is greatly improved. However, when the number of iterations is further increased, the bit error rate decreases at a slower speed, and the increase in the number of iterations leads to an increase in the demand for hardware resources.

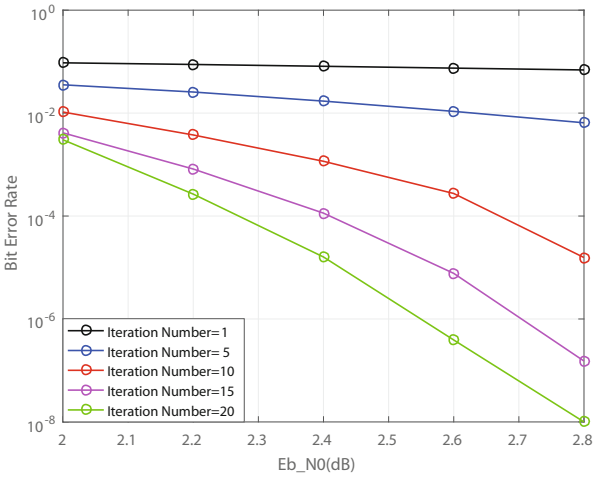

Fig. 6. BER performance with different iteration numbers.

5 Conclusion

According to the LDPC codes decoding methods applied in this paper, the increase of iteration numbers have a significant influence on bit error performance under the same code length and signal-to-noise ratio. With the increase of iteration numbers the bit error rate drop dramatically. This paper has analyzed BP based decoding algorithms and the simulation results suggest that the LDPC codes have a strong capability to correct error bits during the transmission of information.

Acknowledgments. This work was supported by National Natural Science Foundation of China (NSFC) (61671184).

References

1. Gallager, R.G.: Low-density parity-check codes. IRE Trans. Inf. Theory, **IT-18**, 21–28 (1962)
2. MacKay, D.J.C., Neal, R.M.: Near Shannon limit performance of low density parity check codes. Electron. Lett. **32**(18), 1645–1646 (1996)
3. Richardson, T.J., Shokrollahi, M.A., Urbanke, R.: Design of capacity-approaching irregular low-densityparity-check codes. IEEE Trans. Inf. Theory **47**, 619–637 (2001)
4. Hu, X.-Y., Eleftheriou, E., Arnold, D.-M.: Regular and irregular progressive edge-growth tanner graphs. IEEE Trans. Inform. Theory **51**, 386–98 (2005)
5. Prompakdee, P., Phakphisut, W., Supnithi, P.: Quasi cyclic-LDPC codes based on PEG algorithm with maximized girth property. In: IEEE International Symposium on Intelligent Signal Processing and Communications Systems, pp. 1–4 (2012)
6. Fossorier, M.P.C., Mihaljevic, M., Imai, I.: Reduced complexity iterative decoding of low density parity check codes based on belief propagation. IEEE Trans. Commun. **47**, 673–680 (1999)
7. Ning, H., Hua, Q.: Research of LDPC decoding based on modified LLR BP algorithm. J. Southwest Univ. (Nat. Sci. Ed.) 119–124 (2009)
8. Jia, H.: Principle and Application of LDPC, 1st edn. People's Posts and Telecommunications Press, Beijing (2009)