doi:10.3772/j. issn. 1006-6748. 2015. 04. 017

Design of bilayer lengthened LDPC codes over expanded graph for relay channels[①]

Liu Yang (刘　洋), Li Ying[②], Su Yuping
(State Key Laboratory of ISN, Xidian University, Xi'an 710071, P. R. China)

Abstract

This paper presents a low complexity optimized algorithm for design of bilayer lengthened LDPC (BL-LDPC) code for decode-and-forward relay system. The design is performed over the expanded graph of the BL-LDPC code, which consists of the original bilayer graph and the extra added relay-generated parity check bits. To build up our proposed optimized algorithm, we present a modified Gaussian approximation algorithm for the expanded structure of the BL-LDPC code. Then using the proposed optimized algorithm, we find the optimum overall expanded graph of the BL-LDPC code. Simulation results show that the BL-LDPC codes obtained by our proposed optimized algorithm have excellent bit-error-rate performances and small gaps between the convergence thresholds and the theoretical limits when transmitted over the additive white Gaussian noise channels.

Key words: bilayer LDPC codes, relay channel, decode-and-forward, Gaussian approximation, channel capacity

0 Introduction

The three-node relay channel was introduced in Ref. [1] and the first capacity results were presented in Ref. [2]. While the capacity of the general relay channel is still unknown, recent years there have been a vast amount of researches on this topic, both in the information theory and coding communities. Among them, one main relay protocol is the decode-and-forward (DF) relay scheme, which is proved to achieve the capacity of the relay channel for certain special cases[2]. The classic DF technique is based on random binning[1], where the relay decodes the source data and provides a re-encoded copy (bin index) of the source message to the destination.

One of the important researches for practical implementation of a DF relay scheme is the design of practical codes. In this research field, authors in Ref. [3] proposed a novel bilayer LDPC code used in Gaussian relay channel, which was designed to approach the information theoretic limit of the DF scheme. Bilayer LDPC codes consist of two general types called bilayer-expurgated LDPC (BE-LDPC) codes and bilayer-lengthened LDPC (BL-LDPC) codes.

A bilayer LDPC code consists of one higher rate code optimized for the source - to - relay link as well as one lower rate code optimized for the source-to-destination link. As for BL-LDPC code, the lower rate code is designed first and then the higher rate code is designed by increasing the codeword length while keeping the parameters of the lower rate code unchanged. In other words, designing a BL-LDPC code corresponds to finding a bilayer graph so that the lower graph corresponds to a good LDPC code at rate R_- optimized for SNR_-, while the bilayer graph represents a good LDPC code at rate R_+ optimized for SNR_+. More researches on the design of BL-LDPC codes were found in Refs[4-6]. In Ref. [4], the authors designed BL-LDPC codes for the Gaussian relay channel using irregular check degrees. The design of BL-LDPC code for Rayleigh fading relay channels was studied in Ref. [5]. Authors in Ref. [6] proposed a technique to design complexity-optimized BL-LDPC codes. However, these designs are all based on the bilayer graph not on the overall expanded graph of the BL-LDPC codes, which expands with an addition of extra parity check bits from the relay.

The authors in Ref. [7] first proposed a BL-LDPC code design when joint decoding was performed over the overall expanded graph. The optimization algorithm was based on density evolution, including three steps.

[①] Supported by the National Basic Research Program of China (No. 2012CB316100) and the National Natural Science Foundation of China (No. 61072064, 61201140, 61301177).

[②] To whom correspondence should be addressed. E-mail: yli@mail. xidian. edu. cn

Received on July 14, 2014

They first optimized the lower graph of BL-LDPC codes at R_-, then fixing the lower graph, optimized the upper graph of BL-LDPC codes so that the bilayer graph of BL-LDPC codes was optimized at R_+. In the last stage, they optimized the overall expanded graph at R_- by fixing the optimized BL-LDPC code. This three-stage method resulted in a lack of global optimization. Meanwhile, since density evolution algorithm is an infinite-dimensional problem, this optimization algorithm is very much complex.

In this study, a low complexity algorithm is proposed for the design of the BL-LDPC codes over the overall expanded graph, denoted as expanded BL-LDPC codes. To simplify the process, only the mean of message updates will be tracked, which are supposed to be Gaussian mixtures[8]. Based on this fact, a modified Gaussian approximation for the expanded BL-LDPC codes will be presented. The proposed algorithm aims at optimizing the degree distributions of the lower variable nodes and the upper variable nodes simultaneously. the bit-error-rate (BER) performances of the expanded BL-LDPC codes obtained by the proposed optimization algorithm will be simulated and their decoding thresholds will be calculated using the modified Gaussian approximation. Simulation results show that the proposed expanded BL-LDPC codes are excellent in BER performances and small gaps between the convergence thresholds and the theoretical limits.

1 Preliminaries

1.1 DF strategy

A Gaussian degraded relay channel is shown in Fig. 1, where X_1 and X_2 denote the transmitted signals from the source and relay while Y and Y_1 denote the received signals at the destination and relay. The transmission can be defined by

$$\begin{aligned} Y_1 &= X_1 + Z_1 \\ Y &= X_1 + X_2 + Z_2 \end{aligned} \tag{1}$$

where $Z_1 \sim N(0, N_1)$ and $Z_2 \sim N(0, N_1 + N_2)$ denote Gaussian noises at relay and destination respectively. The power constraints at the source and relay are P_1 and P_2. The overall DF rate at source is

$$R = \max_{\alpha} \min\left\{ \frac{1}{2}\log\left(1 + \frac{\alpha P_1}{N_1}\right), \right.$$
$$\left. \frac{1}{2}\log\left(1 + \frac{P_1 + P_2 + 2\sqrt{(1-\alpha)P_1P_2}}{N_1 + N_2}\right) \right\} \tag{2}$$

where α is the optimal cooperation factor to maximize the rate. To ensure a successful decoding at the destination, the rate for relay's codeword X_2 must satisfy

$$R_2 \leqslant \frac{1}{2}\log\left(1 + \frac{(\sqrt{P_2} + \sqrt{(1-\alpha)P_1})^2}{\alpha P_1 + N_1 + N_2}\right) \tag{3}$$

Please refer to Ref. [2] to review a full DF strategy.

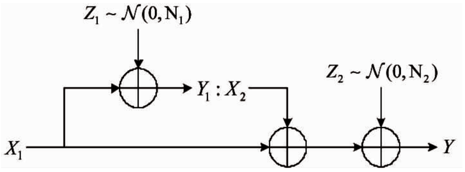

$Z_1 \sim \mathcal{N}(0, N_1)$
$Y_1 : X_2$
$Z_2 \sim \mathcal{N}(0, N_2)$
X_1
Y

Fig. 1 Gaussian degraded relay channel model

1.2 Coding for DF

Based on the DF strategy in Ref. [1], authors in Ref. [3] formulized a general code design problem for the scheme. The code design involves the construction of two codebooks: source codebook X_1 and relay codebook X_2. The relay codebook X_2 can be constructed as a conventional error-correcting code that guarantees successful decoding at the destination. In contrast, the source codebook X_1 must be constructed so that it can be decoded successfully at both relay and destination. The source's codebook needs to be designed at two SNR values. The first SNR value ensures the relay can successfully decode X_1 at $SNR_+ = \alpha P_1 / N_1$ while the second SNR value ensures a successful decoding of X_1 with the help of extra parity bits from the relay at $SNR_- = \alpha P_1 / (N_1 + N_2)$.

The code construction problem is illustrated in a schematic form shown in Fig. 2, where

$$\begin{aligned} R_+ &= \frac{1}{2}\log\left(1 + \frac{\alpha P_1}{N_1}\right) \\ R_- &= \frac{1}{2}\log\left(1 + \frac{\alpha P_1}{N_1 + N_2}\right) \end{aligned} \tag{4}$$

denote the effective source-to-relay and source-to-destination rates. These two rates are the targeted rates for the bilayer LDPC code design.

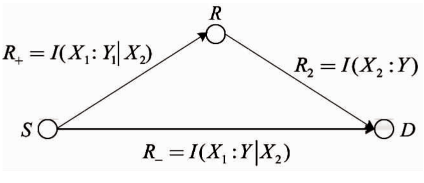


Fig. 2 The achievable rates R_+, R_- and R_2 using DF strategy

2 Expanded BL-LDPC code design

2.1 Expanded BL-LDPC code description

The Tanner graph of the BL-LDPC code structure[3] is depicted in Fig. 3. It is divided into lower and upper graphs, which have n_1 and n_2 variable nodes respectively. Both sets of the variable nodes are connected to k_1 check nodes. Note that the bilayer graph corresponds to a lengthened version of the lower graph by adding n_2 variable nodes while keeping k_1 check nodes fixed. As mentioned above, designing a BL-LDPC code corresponds to finding the bilayer graph so that the lower graph corresponds to a good LDPC code at rate R_- optimized for SNR_- while the bilayer graph represents a good LDPC code at rate R_+ optimized for SNR_+.

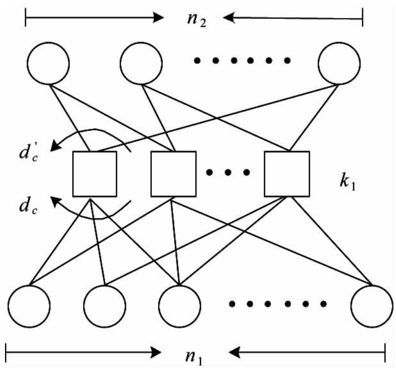


Fig. 3 Tanner graph of a BL-LDPC code

In this work, BL-LDPC code design over the overall expanded graph is proposed which consists of the bilayer graph of the BL-LDPC code and the relay-generated parity bits. The overall expanded graph of the BL-LDPC code is depicted in Fig. 4, where it adds up k_2 extra relay-generated parity bits. At this point onwards, the Tanner graph in Fig. 4 is refered as an expanded bilayer lengthened LDPC code (EBL-LDPC) and the Tanner graph in Fig. 3 as a bilayer lengthened LDPC code (BL-LDPC).

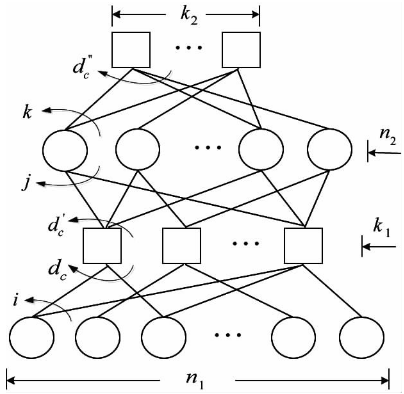


Fig. 4 Tanner graph of an expanded BL-LDPC code

As shown in Fig. 4, assuming regular check degrees, the ensemble of EBL-LDPC codes is defined by the lower variable degree distribution of n_1 variable nodes, the upper variable degree distribution of n_2 variable nodes and three regular check degrees d_c, d'_c and d''_c. The lower variable degree distribution $\lambda_{[i,0,0]}$ denotes the probability that an edge is connected to a degree i variable node in the lower graph. Similarly, $\lambda_{[0,j,k]}$ corresponds to the upper variable degree distribution which denotes the probability that an edge is connected to a degree (j,k) variable node in the upper graph. A variable node is said to have degree (j,k) if it has a lower degree j and an upper degree k in the upper graph. The variable degree distributions $\lambda_{[i,0,0]}$ and $\lambda_{[0,j,k]}$ must satisfy $\sum_i \lambda_{[i,0,0]} + \sum_{j,k} \lambda_{[0,j,k]} = 1$, where i,j,k are not all zero at the same time.

Note that there are three edge types in Fig. 4. The first edge type connects the sets of n_1 variable nodes and k_1 check nodes while the second edge type connects n_2 variable nodes and k_1 check nodes. The third edge type corresponds to the edges between n_2 variable nodes and k_2 extra parity bits. Now define two parameters η_1 and η_2 which denote the percentages of the first edge type and the second edge type respectively. It is easy to obtain the following relations.

$$\begin{aligned} \eta_1 &= d_c k_1 / ((d_c + d'_c) k_1 + d''_c k_2) \\ \eta_2 &= d'_c k_1 / ((d_c + d'_c) k_1 + d''_c k_2) \end{aligned} \tag{5}$$

The conventional design of EBL-LDPC codes can be divided into three stages. In the first stage, optimize the first edge type at R_- which represents a good LDPC code for the lower graph of the BL-LDPC code in Fig. 3. In the second stage, fixing the first edge type, optimize the second edge type so that the bilayer graph of the BL-LDPC code is a capacity-approaching LDPC code at R_+. In the last stage, fixing the optimized BL-LDPC code at R_+, search the third edge type in order that the overall expanded graph of the BL-LDPC code is optimized at R_-.

2.2 A modified Gaussian approximation for the EBL-LDPC codes

In this section, only the mean of message updates is tracked instead of the complete distribution, based on which a modified Gaussian approximation algorithm is presented to design the EBL-LDPC codes over the overall expanded graph.

Let $m_{v,1}^{(l)}$ denote the average mean of message updates at the l^{th} iteration at the output of n_1 variable nodes in the lower graph. Likewise, $m_{v,2}^{(1),(l)}$ and $m_{v,2}^{(2),(l)}$ denote the average means of message updates at the l^{th} iteration at the input of k_1 lower check nodes and k_2 upper check nodes respectively. Similarly, let $m_u^{(l)}(0,0,d''_c)$ denote the mean of message updates at the l^{th} iteration at the output of k_2 check nodes. Then $m_u^{(1),l)}(d_c, d'_c, 0)$ and $m_u^{(2),(l)}(d_c, d'_c, 0)$ denote the means of messages updates at the l^{th} iteration at the input of n_1 and n_2 variable nodes respectively. Using the message updating rule at the variable node for the belief propagation (BP) algorithm, the following equations are obtained.

$$\begin{aligned} m_{v,1}^{(l)}(i,0,0) &= m_{u_0} + (i-1)m_u^{(1),(l-1)}(d_c, d'_c, 0) \\ m_{v,2}^{(1),(l)}(0,j,k) &= m_{u_0} + (j-1)m_u^{(2),(l-1)}(d_c, d'_c, 0) \\ &\quad + km_u^{(l-1)}(0,0,d''_c) \\ m_{v,2}^{(2),(l)}(0,j,k) &= m_{u_0} + jm_u^{(2),(l-1)}(d_c, d'_c, 0) \\ &\quad + (k-1)m_u^{(l-1)}(0,0,d''_c) \end{aligned} \tag{6}$$

where $m_{v,1}^{(l)}(i,0,0)$ denotes the mean of the Gaussian output from n_1 variable nodes with degree i, $m_{v,2}^{(1),(l)}(0, j,k)$ and $m_{v,2}^{(2),(l)}(0,j,k)$ denote the means of the Gaussian outputs from n_2 variable nodes with degree (j, k) to k_1 and k_2 check nodes respectively. Here, m_{u_0} denotes the channel log likelihood ratio (LLR) mean and we have $m_{u_0} = 2/\sigma_c^2$ where σ_c^2 is the channel noise variance.

Hence, the average means of message updates at the n_1 and n_2 variable nodes are obtained as follows.

$$\begin{aligned} m_{v,1}^{(l)} &= \frac{1}{\eta_1}\sum_i \lambda_{[i,0,0]} m_{v,1}^{(l)}(i,0,0) \\ m_{v,2}^{(1),(l)} &= \frac{1}{\eta_2}\sum_{j,k}\left(\frac{j}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(1),(l)}(0,j,k)\right) \\ m_{v,2}^{(2),(l)} &= \frac{1}{1-\eta_1-\eta_2}\sum_{j,k}\left(\frac{k}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(2),(l)}(0,j,k)\right) \end{aligned} \tag{7}$$

Similarly, working with the message updating rule at the check nodes, it is easy to obtain:

$$\begin{aligned} m_u^{(l-1)}(0,0,d''_c) &= \phi^{-1}\Bigg(1 - \Bigg[1 - \frac{1}{1-\eta_1-\eta_2}\sum_{j,k}\frac{k}{j+k} \\ &\qquad \lambda_{[0,j,k]}\phi(m_{v,2}^{(2),(l-1)}(0,j,k))\Bigg]^{d''_c - 1}\Bigg) \\ m_u^{(1),(l-1)}(d_c, d'_c, 0) &= \\ \phi^{-1}\Bigg(1 - \Bigg[1 - \frac{1}{\eta_2}&\sum_{j,k}\frac{j}{j+k}\lambda_{[0,j,k]}\phi(m_{v,2}^{(1),(l-1)}(0,j,k))\Bigg]^{d'_c} \\ &\times\Bigg[1 - \frac{1}{\eta_1}\sum_i \lambda_{[i,0,0]}\phi(m_{v,1}^{(l-1)}(i,0,0))\Bigg]^{d_c - 1}\Bigg) \\ m_u^{(2),(l-1)}(d_c, d'_c, 0) &= \\ \phi^{-1}\Bigg(1 - \Bigg[1 - \frac{1}{\eta_2}&\sum_{j,k}\frac{j}{j+k}\lambda_{[0,j,k]}\phi(m_{v,2}^{(1),(l-1)}(0,j,k))\Bigg]^{d'_c - 1} \\ &\times\Bigg[1 - \frac{1}{\eta_1}\sum_i \lambda_{[i,0,0]}\phi(m_{v,1}^{(l-1)}(i,0,0))\Bigg]^{d_c}\Bigg) \end{aligned} \tag{8}$$

The function $\phi(x)$ is defined as

$$\phi(x) = \begin{cases} 1 - \dfrac{1}{\sqrt{4\pi x}}\displaystyle\int_{-\infty}^{+\infty}\tanh\left(\frac{u}{2}\right)e^{-(u-x)^2/4x}\mathrm{d}u & x > 0 \\ 1 & x = 0 \end{cases} \tag{9}$$

Based on Eq. (6), the overall mean of message updates at the variable nodes is defined. This yields to

$$\begin{aligned} m_v^{(l)} = \sum_i \lambda_{[i,0,0]} m_{v,1}^{(l)}(i,0,0) + \sum_{j,k}\Bigg(\frac{j}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(1),(l)} \\ (0,j,k) + \frac{k}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(2),(l)}(0,j,k)\Bigg) \end{aligned} \tag{10}$$

The mean $m_v^{(l)}$ can be either computed by separating the contributions of the three edge types with percentage η_1 of the first edge type and η_2 for the second edge type in the overall expanded graph. The computation of m_v is shown as

$$m_v^{(l)} = \eta_1 m_{v,1}^{(l)} + \eta_2 m_{v,2}^{(1),(l)} + (1-\eta_1-\eta_2)m_{v,2}^{(2),(l)} \tag{11}$$

According to Ref. [8], a necessary condition to obtain a successful decoding is given by

$$\begin{aligned} \sum_i \lambda_{[i,0,0]} m_{v,1}^{(l+1)}(i,0,0) + \sum_{j,k}\Bigg(\frac{j}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(1),(l+1)} \\ (0,j,k) + \frac{k}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(2),(l+1)}(0,j,k)\Bigg) \\ < \eta_1 m_{v,1}^{(l)} + \eta_2 m_{v,2}^{(1),(l)} + (1-\eta_1-\eta_2)m_{v,2}^{(2),(l)} \end{aligned} \tag{12}$$

Eq. (12) is approximated linearly in $\lambda_{[i,0,0]}$ and $\lambda_{[0,j,k]}$, which allows for linear optimization programming.

2.3 EBL-LDPC code optimization

The overall expanded BL-LDPC codes can be now optimized. The most powerful algorithm probably consists of optimizing simultaneously n_1 variable nodes with degree distribution $\lambda_{[i,0,0]}$ and n_2 variable nodes with degree distribution $\lambda_{[0,j,k]}$.

For simplicity, fix the check degrees d_c, d'_c and d''_c. The rates of the lower and overall BL-LDPC code are $R_- = 1 - k_1/n_1$ and $R_+ = 1 - k_1/(n_1 + n_2)$. The rate of the EBL-LDPC code is $R_- = 1 - (k_1 + k_2)/(n_1 + n_2)$. It is easy to obtain the following relations.

$$\begin{aligned} k_1 &= \frac{n_1\eta_1}{d_c}\times\left(1/\sum_i\frac{\lambda_{[i,0,0]}}{i}\right) \\ &= \frac{n_2\eta_2}{d'_c}\times\left(1/\sum_{j,k}\frac{\lambda_{[0,j,k]}}{j+k}\right) \\ &= \frac{n_1(1-\eta_1-\eta_2)}{d''_c}\times\left(1/\sum_{j,k}\frac{\lambda_{[0,j,k]}}{j+k}\right) \end{aligned} \tag{13}$$

$$k_2 = \frac{n_2(1-\eta_1-\eta_2)}{d''_c}\times\left(1/\sum_{j,k}\frac{\lambda_{[0,j,k]}}{j+k}\right) \tag{14}$$

From Eq. (13), the following relation can be obtained:

$$\sum_{j,k}\frac{\lambda_{[0,j,k]}}{j+k} = \frac{(1-\eta_1-\eta_2)d_c}{\eta_1 d''_c}\times\sum_i\frac{\lambda_{[i,0,0]}}{i} \tag{15}$$

Recall that the design of EBL-LDPC codes involves finding a pair of variable degree distributions $\lambda_{[i,0,0]}$, $\lambda_{[0,j,k]}$ and a group of check degrees d_c, d'_c and d''_c such that the lower graph corresponds to a good LDPC code at rate R_- optimized for SNR_- while the bilayer graph and the overall expanded graph represent good LDPC codes at rates R_+ and R_- optimized for SNR_+ and SNR_- respectively. Fixing the parameters k_1, k_2, d_c, d'_c, d''_c and combining the expressions in Eqs(13)-(15), it is concluded that rates R_-, R_+ can be maximized by maximizing $\sum_i \lambda_{[i,0,0]}/i$. Taking into account the constraint in Eq. (12), the linear programming can be formulated as follows.

$$\max \sum_i \frac{\lambda_{[i,0,0]}}{i} \tag{16}$$

subject to:

$$\sum_i \lambda_{[i,0,0]} m_{v,1}^{(l+1)}(i,0,0) + \sum_{j,k}\left(\frac{j}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(1),(l+1)}(0,j,k) + \frac{k}{j+k}\lambda_{[0,j,k]} m_{v,2}^{(2),(l+1)}(0,j,k)\right) < \mu_k(\eta_1 m_{v,1}^{(l)} + \eta_2 m_{v,2}^{(1),(l)} + (1-\eta_1-\eta_2) m_{v,2}^{(2),(l)}) \tag{17}$$

$$\sum_{j,k} \frac{\lambda_{[0,j,k]}}{j+k} = \frac{(1-\eta_1-\eta_2)d_c}{\eta_1 d''_c} \times \sum_i \frac{\lambda_{[i,0,0]}}{i} \tag{18}$$

$$\begin{aligned} &\frac{1}{\eta_1}\sum_i \lambda_{[i,0,0]} = 1 \\ &\frac{1}{\eta_2}\sum_{j,k} \frac{j}{j+k}\lambda_{[0,j,k]} = 1 \\ &\frac{1}{1-\eta_1-\eta_2}\sum_{j,k} \frac{k}{j+k}\lambda_{[0,j,k]} = 1 \end{aligned} \tag{19}$$

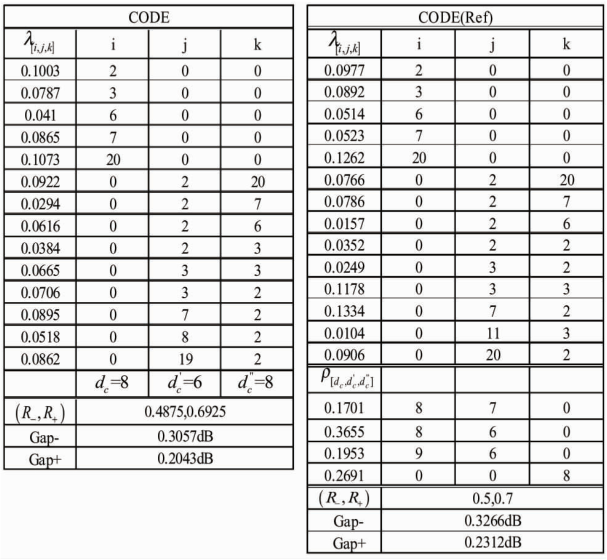
A parameter μ_k is introduced in Ref. [3]. It is slowly increased at each optimization iteration and approaches 1 eventually.

3 Simulation results

In this paper, since it is needed to compare the results with those in Ref. [7], only the case of additive white Gaussian noise (AWGN) channel is dealt with.

Recall that in Ref. [7], using density evolution, the code design starts with optimizing the bilayer graph of the BL-LDPC code at $R_+ = 0.7$, which takes place in three stages. First, optimize the first edge type using an optimized irregular LDPC code at $R_- = 0.5$. Second, fix the optimized first edge type, and design the BL-LDPC code at $R_+ = 0.7$ by adding n_2 variable nodes using the second edge type. In the next step, the optimized bilayer graph is fixed and the overall expanded graph is designed at $R_- = 0.5$ using the third edge type. They designed the optimized degree distribution (node-perspective) in the right side of Table 1. Denote this code as CODE(Ref). The gaps between the convergence thresholds of CODE(Ref) and the theoretical limits are calculated, denoted by gap$_+$ and gap$_-$ respectively.

Table 1 The degree distributions of CODE and CODE(Ref) at target rate $(R_+, R_-) = (0.7, 0.5)$



CODE			
$\lambda_{[i,j,k]}$	i	j	k
0.1003	2	0	0
0.0787	3	0	0
0.041	6	0	0
0.0865	7	0	0
0.1073	20	0	0
0.0922	0	2	20
0.0294	0	2	7
0.0616	0	2	6
0.0384	0	2	3
0.0665	0	3	3
0.0706	0	3	2
0.0895	0	7	2
0.0518	0	8	2
0.0862	0	19	2
	d_c=8	d'_c=6	d''_c=8
(R_-, R_+)	0.4875,0.6925		
Gap-	0.3057dB		
Gap+	0.2043dB		

CODE(Ref)			
$\lambda_{[i,j,k]}$	i	j	k
0.0977	2	0	0
0.0892	3	0	0
0.0514	6	0	0
0.0523	7	0	0
0.1262	20	0	0
0.0766	0	2	20
0.0786	0	2	7
0.0157	0	2	6
0.0352	0	2	2
0.0249	0	3	2
0.1178	0	3	3
0.1334	0	7	2
0.0104	0	11	3
0.0906	0	20	2
$\rho_{[d_c,d'_c,d''_c]}$			
0.1701	8	7	0
0.3655	8	6	0
0.1953	9	6	0
0.2691	0	0	8
(R_-, R_+)	0.5,0.7		
Gap-	0.3266dB		
Gap+	0.2312dB		

For ease of comparison, take an overall code length of 10000 bits and denote this code as CODE. Applying our proposed optimized algorithm, the EBL-LDPC code is designed for $(R_+, R_-) = (0.7, 0.5)$ and then the optimized degree distributions (node-perspective) are given in the left side of Table 1.

The convergence thresholds of CODE (Ref) are within 0.3266dB and 0.2312dB from the theoretical limits at $R_+ = 0.7$ and $R_- = 0.5$, which are calculated using Gaussian approximation. In comparison, the convergence thresholds of CODE obtained by our proposed algorithm are within 0.3040dB and 0.205dB from the theoretical limits at $R_+ = 0.6925$ and $R_- = 0.4875$. The gaps are slightly smaller than those in Ref. [7]. Although there exists minor rate loss, this is admissible as stated in Ref. [3]. Fig. 5 plots the BER performances of these two codes. These curves show that the BER performance of CODE designed by our proposed algorithm is also slightly better than that of CODE(Ref) designed by density evolution. Here, it should be noted that the performance gains are more than the drop in theoretical limits due to the minor rate loss.

To summarize, compared with the optimization process in Ref. [7], the proposed algorithm has low complexity while maintaining the same optimization accuracy as density evolution. Specially, our optimization algorithm is based on a modified Gaussian approximation which can convert an infinite-dimension problem of density evolution to a one-dimensional problem. By doing this, the complexity can be significantly reduced. Besides, the optimization process in Ref. [7] is divided into three stages which lacks of global optimization. Our algorithm is performed in one step which can optimize the degree distributions $\lambda_{[i,0,0]}$ and $\lambda_{[0,j,k]}$ simultaneously and the gaps to the theoretical limits of the proposed codes are smaller.

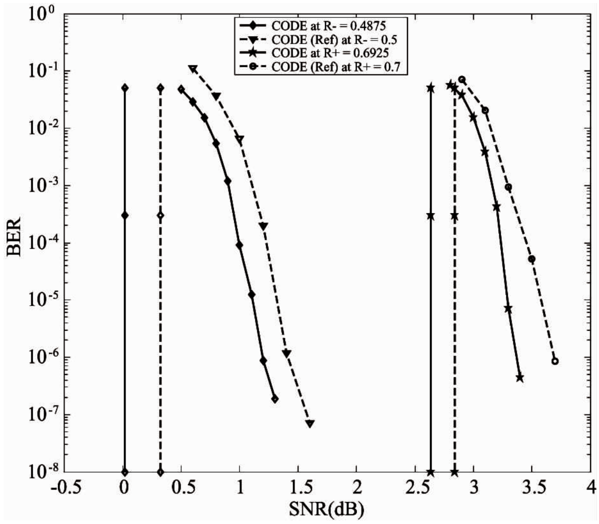


Fig. 5 Compare the BER performances of CODE with that of CODE(Ref). Solid straight lines represent theoretical limits of CODE; dashed straight lines represent the convergence thresholds of CODE

4 Conclusion

In this study, a low complexity optimized algorithm is proposed for the design of BL-LDPC codes over the expanded graph by adding up the extra relay-generated parity check bits. A modified Gaussian approximation algorithm is presented to build up the optimized algorithm and analyze the asymptotic performances. Our proposed algorithm shows that it is possible to optimize the degree distributions of the lower and the upper variable nodes simultaneously. Simulation results show that the BL-LDPC codes obtained by using the proposed low complexity algorithm have slightly better BER performances and smaller gaps to theoretical limits compared with those obtained by density evolution algorithm in Ref. [7].

Reference

[1] Thomas M C, Abbas A E G. Capacity theorems for the relay channel. *IEEE Transactions on Informactions Theory*, 1979, 25(5): 572-584

[2] Gerhard K, Michael G, Piyush G. Cooperative strategies and capacity theorems for relay networks. *IEEE Transactions on Informactions Theory*, 2005, 51(9): 3037-3063

[3] Peyman R, Wei Y. Bilayer low-density parity-check codes for decode-and-forward in relay channels. *IEEE Transactions on Informactions Theory*, 2007, 53(10): 3723-3739

[4] Marwan H A, Jinhong Y, Jun N, et al. Improved bilayer LDPC codes using irregular check node degree distribution. In: Proceedings of IEEE Symposium Information Theory, Toronto, Canada, 2008. 141-145

[5] Osso V, Masoud S. Design of bilayer lengthened LDPC codes for Rayleigh fading relay channels. In: Proceedings of the 45th Annual Conference on Information Sciences and Systems, Baltimore, USA, 2011. 1-5

[6] Osso V, Masoud S. Design of complexity- optimized bilayer lengthened LDPC codes for relay channels. In: Proceedings of the 49th Annual Allerton Conference Allerton House, Monticello, USA, 2011. 1019-1024

[7] Azmi M H, Yuan J H. Performance of bilayer-lengthened LDPC codes under joint decoding. In: Proceedings of IEEE Information Theory Workshop, Taormina, Italy, 2009. 163-167

[8] Chung S Y, Richardson T J, et al. Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation. *IEEE Transactions on Informaction Theory*, 2001, 47(2): 657-670

Liu Yang, born in 1988. She is now studying for a Ph. D degree in State Key Lab of ISN, Xidian University. She received her B. S. degree from Xidian University in 2010. Her research interests include the construction of LDPC codes, spatially coupled codes and the design of LDPC codes for relay systems.