Performance Analysis of Iterative Decoding Algorithms for PEG LDPC Codes in Nakagami Fading Channels

Omran Al Rasheed, Dajana M. Radović, *StudentMember, IEEE,*
and Predrag N. Ivaniš, *Member, IEEE*

***Abstract*—In this paper we give a comparative analysis of decoding algorithms of Low Density Parity Check (LDPC) codes in a channel with the Nakagami distribution of the fading envelope. We consider the Progressive Edge-Growth (PEG) method and Improved PEG method for the parity check matrix construction, which can be used to avoid short girths, small trapping sets and a high level of error floor. A comparative analysis of several classes of LDPC codes in various propagation conditions and decoded using different decoding algorithms is also presented.**

***Keywords* — Progressive edge-growth, extrinsic message degree, Belief propagation, Nakagami fading, trapping sets, Gradient Descent Bit-Flipping.**

I. Introduction

Low density parity check codes (LDPC) are forward error-correction codes, proposed in Gallager's PhD thesis at MIT in the 1960s [1], and since then many methods have been introduced in order to construct these codes. All of these methods can be divided into two groups. The first group contains pseudo-random LDPC codes or random codes with constraints. Gallager's codes belong to this group, as well as MacKay and Neal's codes [2]-[3]. The other group consists of structured LDPC codes. A number of structured LDPC codes and their constructing algorithms are described in [4]-[7]. Among these are codes based on finite geometries, projective and Euclidean. Even Gallager's codes can be obtained using finite geometries. Each of these methods is an attempt to overcome problems typical for this class of codes such as cycles or trapping sets, since the optimum decoding can be provided by iterative decoders. Many of them were successful which led to codes approaching to the Shannon limit within a few hundredths of a decibel [8]-[9].

In this paper, we analyze a class of LDPC codes proposed in [10]. These codes are a result of a carefully constructed Tanner graph, created by adequately connecting variable and check nodes in an edge-by-edge manner. Hence the name progressive edge-growth (PEG) codes. Based on a number of variable and check nodes and a variable - node - degree sequence, the optimum connection at the time is established. That way, the graph is updated an edge by an edge, taking care that girth is as large as possible. As a result, regular and irregular PEG LDPC codes can be produced. This is a very simple and efficient algorithm that provides flexibility in codeword length and code rate. Here, we investigate the performance of PEG codes under iterative decoders, while the properties of the code such as the upper and lower bound on the girth or the lower bound on a minimum distance are presented in [10].

The performance of irregular PEG LDPC codes can be improved for a higher signal-to-noise ratio with a slight modification in constructing algorithm [11]. If there is more than one check node with which a connection can be established, so the girth under the current graph structure is maximum, in the regular PEG algorithm a check node with the smallest index will be chosen, or a check node will be picked randomly. In the modified algorithm, we choose a check node which maintains the highest degree of connectivity for the newly created cycles to the rest of the graph. That way these cycles will receive a great amount of information from the rest of the graph which will decrease their negative impact. Effects of the modification are also presented in this paper.

For decoding, many algorithms based on soft decision and message passing between variable and check nodes, such as Sum Product Algorithm (SPA) [9], Min-sum, and λ-min algorithm [12] were used, where λ-min algorithm and Min-sum algorithm are derived from the SPA algorithm. Another class of decoders used here contains Bit flipping (BF) [1], Two-bit bit-flipping (TBBF) [13], Gradient Descent Bit-Flipping (GDBF) and Multi GDBF [14].

The rest of the paper is organized as follows. In section II, channel and system models with the Nakagami-m fading are defined. Conditions, under which the performance of codes and their decoding algorithms have been examined, are presented. Section III contains an exposure of method for constructing PEG LDPC codes as well as the illustration through an example. It is followed by a description of modified PEG method that results in

This work was partially supported by the Serbian Ministry of Education and Science under Technology development project TR32028 - "Advanced Techniques for Efficient Use of Spectrum in Wireless Systems".

Omran Al Rasheed is with the School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: omrano84@hotmail.com).

Dajana M. Radović is with the Innovation Centre of School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: radovic.dajana@gmail.com).

Predrag N. Ivaniš is with the School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (phone: +381113218381; e-mail: predrag.ivanis@etf.rs).

improved PEG codes. In section IV, an overview of the iterative decoding algorithms, used in this paper, is given. Each of these algorithms is briefly described and the most significant characteristics and differences are depicted. The main part of this paper is contained in section V, where the performance of mentioned codes and their decoding algorithms are analyzed and presented. Based on obtained results, conclusions are drawn and pointed out in section VI.

II. Channel and System Model

In this paper, we consider a point-to-point wireless communication system that can be represented by using a block diagram presented in Fig. 1. It is assumed that a randomly generated sequence of k information bits is encoded into the codeword with n bits, and the parity matrix of the corresponding LDPC code is designed using the method explained in the next section. The encoded sequence is then Binary Phase Shift Keying (BPSK) modulated and a resulting signal is sent over the wireless channel. At the receiver, BPSK symbols are coherently demodulated and decoded using different decoding algorithms.

A bit error rate in BPSK system over the Nakagami fading channel can be obtained by averaging the instantaneous BER expression over realizations of SNR process. However, in a LDPC encoded system the BER expression cannot be obtained in a closed form and therefore we will apply Monte Carlo simulation to estimate it.

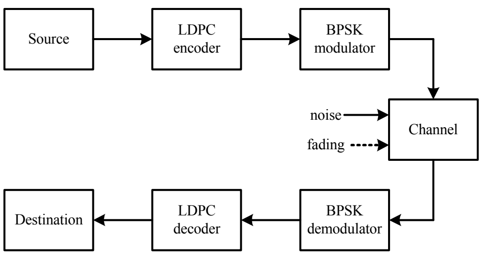

Fig. 1. System block diagram.

It was assumed that encoded and modulated symbols were sent over the channel with the Nakagami-m distributed fading envelope. Therefore, the probability density function (PDF) of fading envelope r, denoted as $p(r)$, is defined as

$$p(r)=\frac{2m^m r^{2m-1}}{\Gamma(m)\Omega^m}\exp\left(\frac{mr^2}{\Omega}\right), \tag{1}$$

where $m=E^2[r^2]/\mathrm{var}(r^2)$, $\Omega=E[r^2]$, and $\Gamma(m)$ denotes a Gamma function. As the received signal is also influenced by noise with power σ_n^2, signal-to-noise ratio (SNR) at the receiver is defined as

$$\gamma = r^2/\sigma_n^2. \tag{2}$$

III. Construction of PEG LDPC Codes

Procedures for code construction for PEG codes are explained in [10], and a short overview of these procedures will be given in this section. First, we determine n variable nodes $V=\{v_0, v_1, v_2, \ldots, v_{n-1}\}$ and m parity check nodes $C=\{c_0, c_1, c_2, \ldots, c_{m-1}\}$. Then, the degree distribution is defined as $\lambda(x)=\sum_{i\geq 2}^{d_{\max}}\lambda_i x^i$, where λ_i presents a fraction of edges emanating from variable nodes of degree i and $d_{\max}$ is a maximum degree of variable nodes. After that, using density evolution, a variable node degree sequence $D_v=\{d_{v0}, d_{v1}, d_{v2}, \ldots, d_{vn-1}\}$ has to be determined in a non-decreasing order $\{d_{v0}\leq d_{v1}\leq d_{v2}\leq \ldots \leq d_{vn-1}\}$. The rest of the procedure will be explained in one illustrative example.

A. *PEG algorithm*

For n=6, m=3 and degree distribution $\lambda(x)$ = $0.8576x+0.14237x^2$, using density evolution, we get five variable nodes with a degree two and one variable node with a degree three. At this moment, bipartite graph is given in Fig. 2 and parity check matrix is

$$\mathrm{H}=\begin{bmatrix}1&1&0&1&1&0\\1&0&1&1&0&0\\0&1&1&0&1&0\end{bmatrix}. \tag{3}$$

Now we arrive to v_5 and use PEG to put connections from v_5 to three check nodes.

Stage 1: As the first connection (k =1), PEG directly selects a check node which has a minimum degree. In this case we have two check nodes with a degree of three (c_1 and c_2), so PEG randomly selects one of them, then adds an edge to c_1, as it is shown in Fig. 3.

Stage 2: As the second connection (k=2), PEG starts from v_5 as follows: v_5 has only one connection to c_1 so during depth (0): $\{\mathcal{N}_{v5}^0=\{c_1\}, \bar{\mathcal{N}}_{v5}^0=\{c_0, c_2\}\}$, PEG moves to depth (1): $\{\mathcal{N}_{v5}^1=\{c_0, c_1, c_2\}, \bar{\mathcal{N}}_{v5}^1=\emptyset\}$, PEG calculates the summation of check nodes in the termination of every depth, so if summation is equal to m, then PEG stops. In this case, summation is equal to three, so PEG returns to the previous depth and selects from $\bar{\mathcal{N}}_{v5}^{l-1}$ check nodes one that has a minimum degree (c_2), Fig. 2.

Stage 3: As the third connection (k=3), as in the previous stage, PEG adds an edge to c_0 and the resulting parity check matrix is:

$$\mathrm{H}=\begin{bmatrix}1&1&0&1&1&0\\1&0&1&1&0&1\\0&1&1&0&1&1\end{bmatrix}. \tag{4}$$

B. *Improved PEG algorithm*

In order to improve the performance of irregular LDPC codes at a high SNR, Hua Xiao modified the PEG procedure [11]. In this modification, the procedure depends on the concept of Extrinsic Message Degree (EMD) of a variable node set defined as a number of check nodes such that are singly connected to this set (shown in Fig. 4). Axiomatically, EMD of a stopping set equals zero.

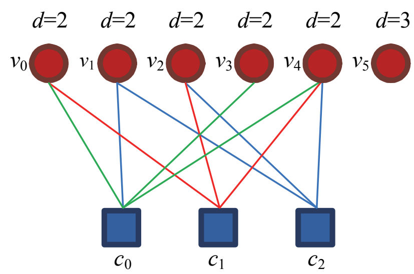

Fig. 2. Bipartite graph before PEG procedure.

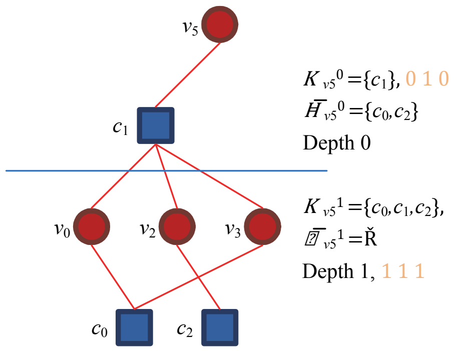

Fig. 3. The second stage from procedure PEG for v_5.

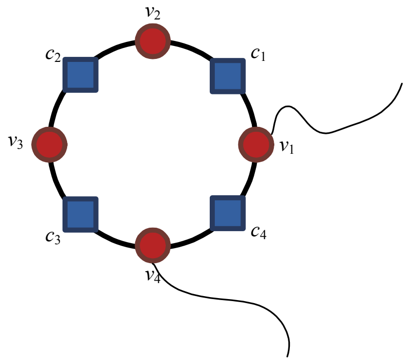

Fig. 4. EMD for this cycle is 2.

In the improved progressive edge-growth (IPEG) we can calculate an Approximate Cycle EMD (ACE) as $\sum_i d_i$ - 2, where d_i is the degree of the ith variable node in this cycle. We must indicate that the value of ACE equals EMD when a cycle does not have any sub-cycles, otherwise ACE>EMD. We can take into account that ACE for any variable node is d_{vj} - 2 and ACE for any check node is zero. Each node (variable or check) receives values from the above edges and selects a minimum degree and adds itself value, then sends the ACE value to its downward edge, as it is shown in Fig. 5.

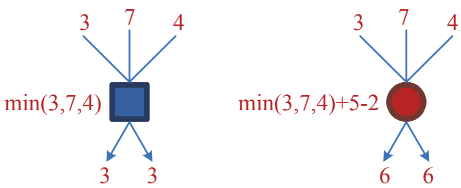

Fig. 5. ACE for check and variable node.

Clearly, the IPEG algorithm is only effective for the construction of irregular codes. For regular codes, the ACE of all created is the same regardless of the selected check node. IPEG selects a maximized ACE, so there are no small stopping sets in IPEG's matrix.

IV. Overview of Iterative Decoding Algorithms

We have previously mentioned that there are various decoding algorithms for LDPC codes, that can be roughly divided into hard-decision algorithms and soft-decision algorithms. Here we introduce a hard-decision algorithm named bit-flipping algorithm, and a soft-decision algorithm named Sum Product Algorithm. The greatest difference between hard-decision decoding and soft-decision decoding is that the former propagates the message of 0 or 1 while the latter propagates the probability of 0 or 1.

It is agreed that BF decoding is very simple and takes little time to get a result. On the other hand, SPA decoding is very complex and takes more time. But there is no comparison between their performances, as it will be shown. To achieve a trade-off between the performance of the algorithm and the time required to decode, many improvements for BF algorithm and simplifications for SPA have been made.

One of the improvements of BF algorithm is the so called Two-Bit Bit Flipping (TBBF) algorithm, in which employment of one additional bit at a variable node, representing its "strength", is suggested. The introduction of this additional bit increases the guaranteed error correction capability by a factor of at least 2. An additional bit can also be employed at a check node to capture information which is beneficial to decoding [13].

Generally, BF algorithms are classified into two classes: single-bit flipping (single BF) algorithms and multiple bits flipping (multi BF) algorithms like GDBF and Multi-GDBF, respectively. The gradient descent inversion function [14] is given by:

$$\Delta_k^{(GDBF)}(x) = x_k y_k + \sum_{i\in M(k)} \prod_{j\in N(i)} x_j \qquad (5)$$

It is composed of the correlation between the hard decision and the received soft value of the corresponding bit plus the syndrome sum, where $\boldsymbol{H}_{mxn}$, $N(i) \triangleq \{j \in [1,n]: h_{ij} = 1\}$ and $M(j) \triangleq \{i \in [1,m]: h_{ij} = 1\}$. In the decoding process of the single BF algorithm, only one bit (k) is flipped if $\Delta_k^{(GDBF)}(x)$ is minimum. On the other hand, in the multi BF algorithm all bits are flipped per iteration if $\Delta_k^{(GDBF)}(x) < \theta$, where θ is an inversion threshold. The multi bit flipping algorithm is expected to have a faster convergence speed than that of the single bit flipping algorithm.

To simplify the SPA algorithm, we have a λ-min algorithm and a Min-sum algorithm. All these algorithms try to approximate this function:

$$\varphi(x) = \log\frac{e^x+1}{e^x-1} \qquad (6)$$

by selecting some of the smaller values of x (λ-min algorithm) or selecting the minimum value of x (min-sum algorithm). It means that a message from the variable node to each check node v_{ji} depends on the minimum absolute value of a messages from check to variable nodes u_{ij} [12].

V. Numerical Results

In this section, numerical results obtained by Monte Carlo simulation method are presented. The aim was to compare PEG LDPC codes in the presence of the Nakagami fading. Codes of various lengths have been examined and performances are compared to other classes of LDPC codes to see how choosing optimum connections between variable and check nodes affects the performance of codes.

In our simulation, we send a sequence of codewords using a matrix $\boldsymbol{H}$ generated according to the PEG algorithm. The noise is generated as a pseudorandom sequence of normal distribution, with zero mean and standard deviation $\sigma = \sqrt{P_s/2}\,10^{-(SNR)/20}$, where *SNR* is a signal-to-noise ratio at a receiver and P_s denotes a power of a received signal. A detailed description of the Nakagami-m fading envelope generation process is given in [16].

First, we analyze PEG LDPC codes with parameters (n,k)=(256,128). As we can see from Fig. 6 and Fig. 7, system performances are greatly improved, and the coding gains are larger for lower fading factor values. It is also clear that the coding gain is much higher for the case of SPA decoding algorithm, even for the case where the maximum number of iterations (denoted by *MaxIt*) is five times reduced when compared to BF and TBBF decoding algorithm. For easier comparison, numbers are given in Table 1.

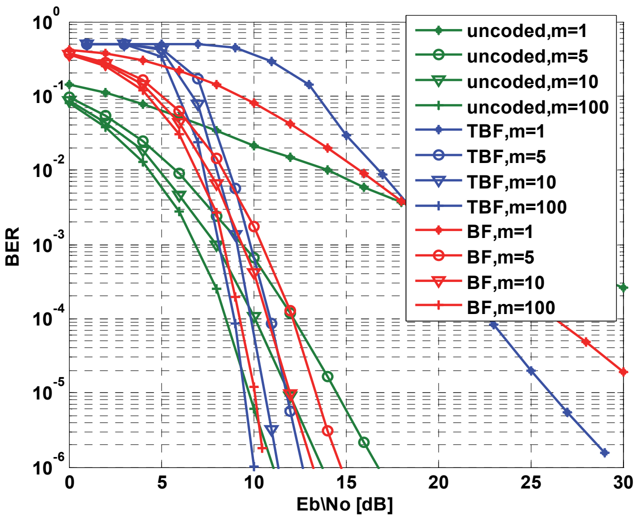

Fig. 6. Performances of PEG LDPC code, BF decoding, *MaxIt*=50, *n*=256, *R*=1/2.

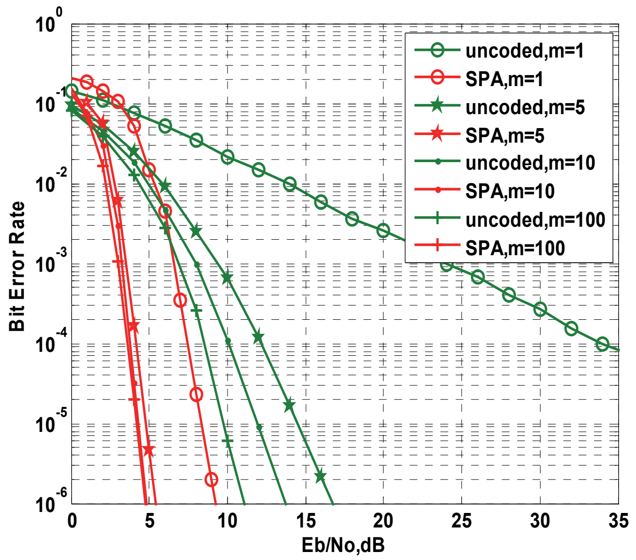

Fig. 7. Performances of PEG LDPC code, SPA decoding, maximum number of iterations *MaxIt*=10, *n*=256, *R*=1/2.

Although the system has better performances for larger values of *m*, the coding gain is smaller as the fluctuations of the signal strength are reduced compared to the Rayleigh fading case when *m*=1. Coding gains presented in Table 1 demonstrate that PEG LDPC codes are especially effective for small and moderate values of fading factor.

TABLE 1: CODING GAINS FOR BER=10^{-6}.

Fading factor	1	5	10	100
BF, 50 iter.	19.5 dB	2 dB	0.5 dB	0.5 dB
TBF, 50 iter.	25 dB	4 dB	2.4 dB	1.05 dB
SPA, 10 iter.	44.8 dB	11.4 dB	8.9 dB	6.4 dB

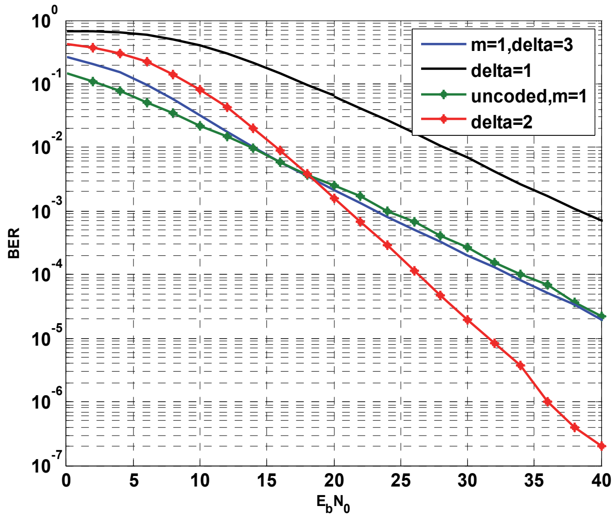

Fig. 8. BER performances for BF decoding algorithm, Δ_k=1,2,3, *n*=256, *MaxIt*=50, *m*=1.

From Fig. 8 we can see the impact of value inversion function Δ_k on the BF algorithm. For a fading factor *m*=1, *n*=256, *R*=0.5 and Δ_k=1, the performance of BF is very bad, but for Δ_k =2 the performance of BF is better. This does not mean that an increased value of Δ_k will improve the performance of BF. We can see that when Δ_k=3, its performance is declining.

We have generated two matrices, the first for a regular PEG LDPC code with parameters (n,k) = (256,128), and we also use MacKay's matrix [3] with the same code rate *R*=0.5. Cycles of length 4 are avoided in Mackay's matrix to reduce the influence of short cycles. We use the same number of iterations and number of error blocks for every SNR value. We can observe that PEG has a better performance than codes by Mackay's matrix because they have succeeded in overcoming an error floor region for regular LDPC codes. The improvement is more noticeable for a larger number of iterations, as it is shown in Fig. 9.

In Fig. 10, we present the performances of PEG LDPC codes for a codeword of length *n*=1000 and *n*=200 and the same code rate *R*=0.5. Numerical results are presented for a typical fading factor value *m*=5, both BF and SPA decoding algorithms and a maximum number of iterations *MaxIt*=10. It can be observed that the impact of the applied decoding algorithm is high, and longer codewords result in better performances but the improvement is noticeable only for lower values of BER.

We also compare two presented construction methods for LDPC – one parity check matrix is generated using

PEG and the second matrix using IPEG with the same degree distribution, codeword length n=256, and code rate R=0.5. We send irregular codewords LDPC on the channel with noise and the Nakagami-m fading with different values of fading factors. As it can be observed from Fig. 7, the region between Eb/N_0=6dB and Eb/N_0=8dB is the most important for m=10 and m=100, and therefore the improvement of IPEG is investigated for this region only. We confirm that IPEG outperforms PEG, especially when a factor m has larger values, as it is shown in Fig. 11.

We use another decoding algorithm (soft decision), Fig. 12 for IPEG construction for n=256, R=0.5, and we can note that the performances of λ-min algorithm for λ=2,3 are close to the performance of SPA algorithm for all values of the Nakagami-m fading and λ-min has better performances than Min-sum algorithm for all cases. The appropriate choice for λ is important to get a better performance [15].

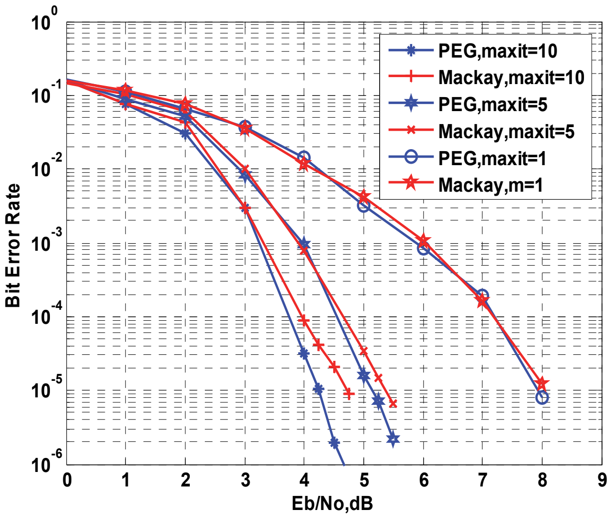

Fig. 9. Performances of PEG and MacKay LDPC codes, SPA decoding, n=256, R=1/2, m=5.

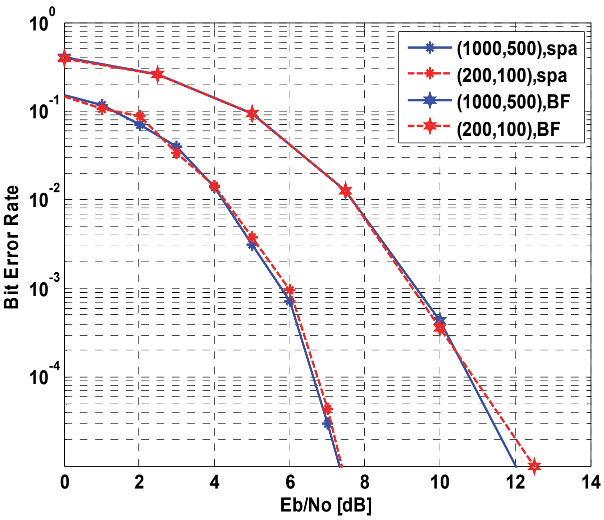

Fig. 10. BER performances for different decoding algorithms and codeword lengths, *MaxIt*=10, m=5.

Finally, we present performances for other decoding algorithms, improvements of BF. From Fig. 13, we can see the performances of BF, TBBF, GDBF and multi-GDBF for IPEG construction, n=256, R=0.5. It is obvious that multi-GDBF has a better performance for m=1 and m=20, especially when m takes small values, m=1. It means that the gains of GDBF and multi-GDBF are much higher in comparison with BF and TBBF. We can see from Fig. 12, Fig. 13 and Table 2 that despite the good performance of Multi-GDBF, the performance of min-sum is better than multi-GDBF in all cases.

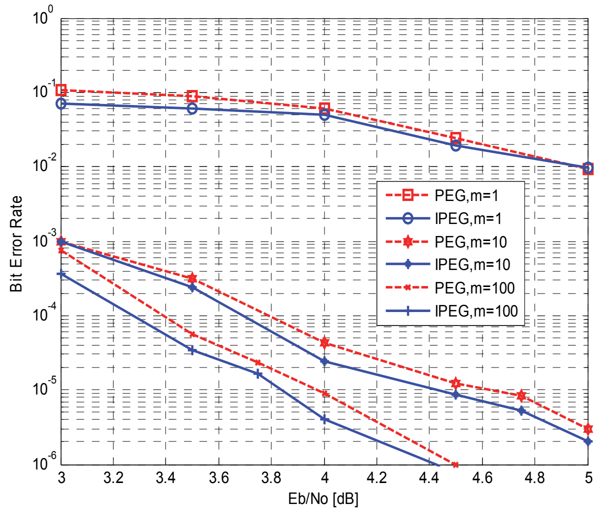

Fig. 11. Performances for IPEG and PEG, n=256, R=1/2, maximum number of iterations *MaxIt*=10.

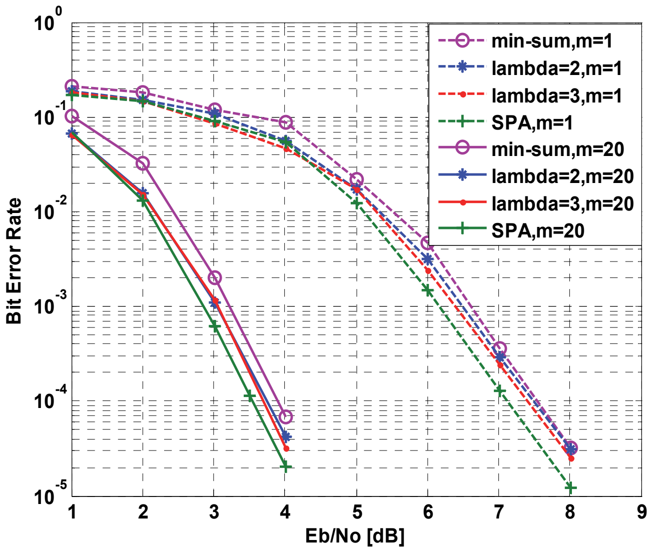

Fig. 12. BER performances for different decoding algorithms and codeword lengths, *MaxIt*=15, m=1, 20.

This means that the best performance of BF still remains below the worst performance of complex decoding algorithms. We have also seen that improved BF algorithms like TBBF, GDBF and Multi-GDBF come close to the performance of lower complexity algorithms like Min-sum algorithm, but do not achieve it.

Finally, the running time required by each algorithm depends highly on its computational complexity and the processing platform used. Table 3 shows the computational complexity of the two algorithms SPA and Multi-GDBF in terms of the number of additions and multiplications needed per iteration. So we have two choices, a high complexity decoding algorithm with a high performance (SPA) or lower complexity with a lower performance.

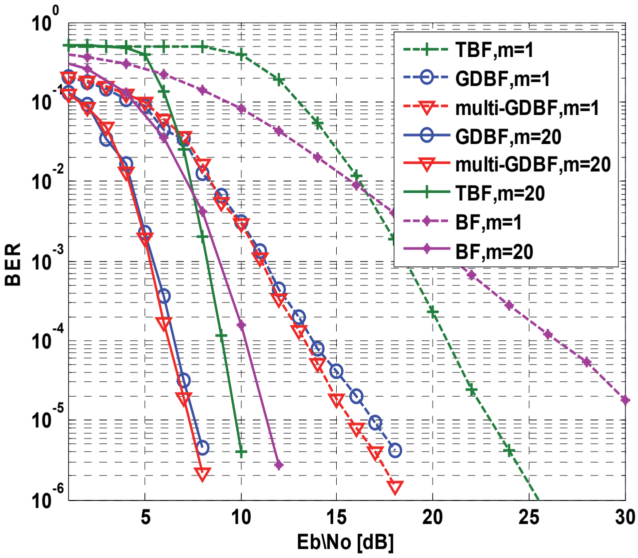

Fig. 13. BER performances for different decoding algorithms and codeword lengths, *MaxIt*=100, *m*=1,20.

TABLE 2: CODING GAINS FOR BER=10^{-4}.

Fading factor	1	20
Multi-GDBF, 100 iter	20.75	2.85
Min-sum, 15 iter	26.50	5.52

TABLE 3: COMPUTATIONAL COMPLEXITY FOR EACH ITERATION.

	+	*
SPA(H)	$3(u-4)Mq(q-1)$	$(3u-4)Mq^2$
SPA(V)	$uM(q-1)$	$Mtuq$
Multi-GDBF	$Nm+N+M-2$	$M(n-1)+N$

N represents the number of variable nodes, M check nodes, t is the mean column weight of H($t \geq 2$), u is the mean row weight of H ($u \geq 2$), n and m are sizes of sets of non-zero elements in any row and in any column of H, respectively, q=2 over the Galois field F_2 and (H)orizontal and (V)ertical step.

VI. CONCLUSION

In this paper we have determined the average bit error rates for BPSK modulated system in the Nakagami fading channel, where two classes of LDPC codes are applied. We have considered PEG and IPEG methods such that the graph grows in an edge-by-edge manner, to construct LDPC codes with a suitable degree distribution.

We have shown the performances of PEG and IPEG codes for different fading factors and codeword lengths for SPA and BF decoding algorithms. It has been observed that the coding gain is especially large for small values of fading factor, when the SPA decoding algorithm is applied.

It has been shown that PEG LDPC codes outperform Mac Key's codes for a large maximum number of iterations, and the error floor at high SNR (as a result of the fading influence) is also reduced. Also, we have noticed that the increased codeword length slightly improves system performances, and the performance improvement of IPEG method, when compared to PEG, is more noticeable for large values of the Nakagami fading factor.

We have seen that λ-min algorithm and Min-sum algorithm reduce the complexity of SPA decoding algorithm and decrease the required time of decoding but at the expense of achieving the best performance.

REFERENCES

[1] R. G. Gallager, “Low Density Parity Check Codes,” MIT Press, Cambridge, Mass., 1963

[2] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[3] D. MacKay, “Good error correcting codes based on very sparse matrices,” *IEEE Trans. Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[4] S. Lin, D. J. Costello, Jr., *Error Control Coding – Fundamentals and Applications,* 2nd edition, Prentice Hall, Upper Saddle River, N. J., 2004.

[5] Y. Kou, S. Lin and M. Fossorier, “Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results”, *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2711-2736, Nov. 2001.

[6] R. M. Tanner, D. Sridhara, and T. Fuja, “A class of group-structured LDPC codes,” in *Proc. ICSTA 2001*, Ambleside, UK, 2001.

[7] S. J. Johnson and S. R.Weller, “Regular low-density parity-check codes from combinatorial designs,” in *Proc. IEEE Information Theory Workshop*, Cairns, Australia, Sep. 2001, pp. 90–92.

[8] S.-Y. Chung, D. Forney, T. J. Richardson and R. L Urbanke,“On the Design of Low-Density Parity Check Codes within 0.0045 dB of the Shannon Limit,“ *IEEE Communications Letters*, vol. 5, no. 2, pp. 58-60, Feb. 2001.

[9] D. J. C. MacKay and R. M. Neal,“Near Shannon Limit Performance of Low Density Parity Check Codes”, *Electronics Letters,* vol. 32, no. 18, pp. 1645-1646, Aug. 1996.

[10] H. Xiao, E. Evangelos and A. M. Dieter, “Regular and Irregular Progressive Edge-Growth Tanner Graphs,” *IEEE Trans. Information Theory,* vol. 51, no. 1, pp. 386-398, Jan. 2005.

[11] H. Xiao and A. H. Banihashemi “Improved Progressive-Edge-Growth (PEG) Construction of Irregular LDPC codes,” *IEEE Communications Letters*, vol.8, no. 12, pp 715-717, Dec. 2004.

[12] E. Boutillon, F. Guilloud and J. Danger, “λ-Min Decoding Algorithm of Regular and Irregular LDPC Codes”. in *Proc. 3rd International Symposium on Turbo Codes and Related Topics*, Brest, France, Sep. 2003. pp. 451-454.

[13] D. V. Nguyen, B. V. Vasic, and M. W. Marcellin, “Two-Bit Bit Flipping Decoding of LDPC Codes”, *ISIT*, St. Petersburg, Russia, 2011, pp. 1995–1999,

[14] T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usami, and I. Takumi, “Gradient descent bit flipping algorithms for decoding LDPC codes,” *IEEE Trans. Communications*, vol. 58, no. 6, pp. 1610–1614, Jun. 2010.

[15] H. Ge, “Investigation of LDPC code in DVB-S2,” Linköpings universitet SE-581 83 Linköping, Sweden, 2012.

[16] M. D. Yacoub, J. E. V. Bautista, L. G. R. Guedes, “On Higher Order Statistics of the Nakagami-m Distribution,” *IEEE Trans. Vehicular Technology*, vol. 48, no.3, pp. 790-794, May 1999.