This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Evaluation of the Extremely Low Block Error Rate of Irregular LDPC Codes

X. Zheng[1], F. C. M. Lau[1], C. K. Tse[1], Yejun He[2] and M. Z. Wang[1]
1. Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong
2. Department of Communication Engineering, College of Information Engineering, Shenzhen University, China
Email: {xia.zheng, encmlau, encktse, enmzwang}@polyu.edu.hk; yjhe@szu.edu.cn

***Abstract*— In this paper, we attempt to evaluate *irregular* LDPC code performance at the high SNR region using the importance sampling (IS) approach in conjunction with primary-trapping-set identification. Results have indicated that our proposed IS scheme can produce speed-up gains up to 3.9×10^9 times compared with Monte Carlo simulations.**

***Index Terms*— Block error rate, error floor, importance sampling, irregular low-density-parity-check codes, trapping set.**

I. Introduction

Trapping set (TS) has first been proposed by Richardson [1] to define a set of variable nodes that may contribute to the error events at high signal-to-noise ratio (SNR) for regular low-density-parity-check (LDPC) codes. The trapping set is labeled as $[w; u]$, where w is the size of the set and u is the number of neighboring check nodes having odd number of connections to the TS. In [2], we have demonstrated that trapping sets (TSs) with the same label $[w; u]$ but different induced connected subgraphs (ICSs) may contribute very differently to the error floor, particularly in the case of *irregular* LDPC codes. To distinguish different types of TSs, we have categorized TSs based on their ICSs having (i) no cycle, (ii) a single-cycle or (iii) multiple cycles. However, using w and u alone is not adequate to provide concrete conclusions on the configuration of a TS-ICS. To overcome this problem, we have introduced a parameter called "cycle indicator (CI)", which is denoted by e and is computed by $e = \sum_{i=1}^{w}(d_i - 2) - u$ where d_i represents the degree of the ith variable node in the TS. Then we have shown that regardless of regular or irregular codes, for a TS-ICS with

- no cycle, then $e < 0$;
- a single cycle, then $e = 0$;
- multiple cycles, T-type variable nodes[1] but no T-type check nodes, then $e > 0$.

Since the value of CI provides extra information on the properties of a TS, we have refined the label of a TS from $[w; u]$ to $[w; u; e]$. Moreover, TSs with different $[w; u; e]$ labels can contribute very different error rates to the overall performance of the code at the high SNR region. We further observe that TS-ICSs with one cycle or multiple cycles containing no T-type check nodes are the main causes of error floor. Subsequently, we have defined a "primary trapping set" (PTS) as a $[w; u; e]$ TS where no check node of degree three or larger exists in the ICS of the TS if the TS-ICS contains two or more cycles [2]. Thus, single-cycle and multiple-cycle $[w; u; e]$ PTS-ICSs with small w and relatively smaller u, which we name as "detrimental PTS-ICSs", are more harmful to the decoder at high SNR region.

Based on the aforementioned observations and findings, we have proposed an approach to design short-length LDPC codes with low error floor by avoiding detrimental PTS-ICSs. Monte Carlo (MC) simulations giving block error rates (BLERs) as small as 10^{-6} have been performed. The results show that our proposed construction algorithm produces codes with lower error floor compared with other codes. Using the MC technique to evaluate the BLER at higher SNR becomes not feasible due to the extremely low BLERs and hence the prohibitive amount of simulation time needed to arrive at a meaningful BLER.

In this paper, we attempt to evaluate *irregular* LDPC code performance at the high SNR region using the importance sampling (IS) approach in conjunction with PTS identification. To the best of our knowledge, no concrete suggestions nor results have been produced for evaluating *irregular* LDPC codes, not to mention irregular LDPC codes identified with $[w; u; e]$ TSs. Here, we will first propose a three-step technique to search as many detrimental PTS-ICSs as possible for a given irregular LDPC code. Next, the error region of the code will be divided into various sub-regions in terms of PTS-ICSs. The sub-regions subject to PTS-ICSs with the same label $[w; u; e]$ will be classified as the same group. Then, we apply IS to a representative from each PTS-ICS group and also to all elements in some selected groups to evaluate their BLERs. Based on the results, the BLER of the LDPC code at high SNR is estimated.

II. Importance sampling for regular LDPC codes

A. Review of MC simulator and IS simulator

For a random variable vector $\boldsymbol{x} = [x_1, x_2, ..., x_N]$ with joint pdf $f(\boldsymbol{x})$ and corresponding "error region" E, the block error rate (BLER) of $\boldsymbol{x}$ equals $P_E = \int_E (\boldsymbol{x} \in E)$. Then the MC estimator of P_E using N_{MC} simulation runs is expressed as $\hat{P}_{\mathrm{MC}} = \frac{1}{N_{\mathrm{MC}}} \sum_i^{N_{\mathrm{MC}}} 1_E(\boldsymbol{x})$, where $1_E(\boldsymbol{x})$ is the indicator

This work was supported by a research grant (G-U586) provided by Hong Kong Polytechnic University. The authors would like to thank the anonymous reviewers for their constructive comments.

[1] A "T-type node" is defined as a node that connects to two or more different cycles [2].

function of the error region E, i.e.,

$$1_E(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in E \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The MC approach becomes not viable when the error events have an extremely low probability of occurrence. It is because most of the time, the simulated events are correct and therefore of no importance.

In an IS estimator, a new random vector $\boldsymbol{x}^*$ sampled from the biased distribution $f^*(\boldsymbol{x}^*)$ is designed to increase the occurrence of rare error events. Then, using the biased distribution, the IS simulator of P_E for N_{IS} runs can be expressed as $\hat{P}_{\text{IS}} = \frac{1}{N_{\text{IS}}} \sum_i^{N_{\text{IS}}} [1_E(\boldsymbol{x}_i^*)\omega(\boldsymbol{x}_i^*)]$ [3], where $\omega(\boldsymbol{x}^*)$ is the weight function given by $\omega(\boldsymbol{x}^*) = \frac{f(\boldsymbol{x}^*)}{f^*(\boldsymbol{x}^*)}$. Mean translation (MT) is one of the popular techniques to form the biased density. The idea of MT is to shift the mean of the original density function to the boundary of the error region, i.e., $f^*(\boldsymbol{x}^*) = f(\boldsymbol{x}^* - \boldsymbol{\mu})$, where $\boldsymbol{\mu} = [\mu_1, \mu_2, ..., \mu_N]$ is a point lying on the boundary. Then the weight function can be re-written as $\omega(\boldsymbol{x}^*) = \frac{f(\boldsymbol{x}^*)}{f(\boldsymbol{x}^* - \boldsymbol{\mu})}$.

Denote $\text{E}(\cdot)$ and $\text{var}(\cdot)$ as the expectation operator and variance operator, respectively. Theoretically,

$$\begin{aligned} \text{E}(\hat{P}_{\text{IS}}) &= \int_E \frac{f(\boldsymbol{x}^*)}{f^*(\boldsymbol{x}^*)} f^*(\boldsymbol{x}^*) dx \\ &= \int_E f(\boldsymbol{x}^*) dx = \text{E}(\hat{P}_{\text{MC}}) = P_E, \end{aligned} \quad (2)$$

which shows that both MC and IS are unbiased estimators of P_E. Moreover, the well-known variance formulas of standard MC and IS estimators are given, respectively, by [4]

$$\text{var}(\hat{P}_{\text{MC}}) = \frac{\hat{P}_{\text{MC}} - \hat{P}_{\text{MC}}^2}{N_{\text{MC}}} \approx \frac{\hat{P}_{\text{MC}}}{N_{\text{MC}}}, \quad (3)$$

and

$$\text{var}(\hat{P}_{\text{IS}}) = \frac{\frac{1}{N_{\text{IS}}} \sum_i^{N_{\text{IS}}} [1_E(\boldsymbol{x}^*)\omega(\boldsymbol{x}^*)]^2 - \hat{P}_{\text{IS}}^2}{N_{\text{IS}}}. \quad (4)$$

In order to evaluate the quality of the estimators, the normalized error of MC and IS simulators are calculated, respectively, as [4]

$$\phi_{\text{MC}} = \frac{\sqrt{\text{var}(\hat{P}_{\text{MC}})}}{\hat{P}_{\text{MC}}} \approx \frac{1}{\sqrt{\hat{P}_{\text{MC}} N_{\text{MC}}}}, \quad (5)$$

and

$$\phi_{\text{IS}} = \frac{\sqrt{\text{var}(\hat{P}_{\text{IS}})}}{\hat{P}_{\text{IS}}} = \sqrt{\frac{1}{N_{\text{IS}}^2 \hat{P}_{\text{IS}}^2} \sum_i^{N_{\text{IS}}} [1_E(\boldsymbol{x}^*)\omega(\boldsymbol{x}^*)]^2 - \frac{1}{N_{\text{IS}}}}. \quad (6)$$

To ensure that the IS simulator provides a good estimation of P_E, the IS simulation should be continued until $\phi_{\text{IS}}^2 \leq \phi_{\text{MC}}^2$ is satisfied [4], i.e.,

$$\frac{\sum_1^{N_{\text{IS}}} [1_E(\boldsymbol{x}^*)\omega(\boldsymbol{x}^*)]^2}{\{\sum_1^{N_{\text{IS}}} [1_E(\boldsymbol{x}^*)\omega(\boldsymbol{x}^*)]\}^2} - \frac{1}{N_{\text{IS}}} \leq \frac{1}{\hat{P}_{\text{MC}} N_{\text{MC}}}. \quad (7)$$

Note that $\hat{P}_{\text{MC}} N_{\text{MC}}$ is actually the number of error events collected in the MC simulator. Moreover, a speed up gain of an IS simulator relative to MC can be calculated using [4] $\gamma_s = \frac{N_{\text{MC}}}{N_{\text{IS}}}|_{\phi_{\text{IS}}=\phi_{\text{MC}}}$.

B. IS scheme for regular LDPC codes

Assume an all-zero codeword with a block length N is transmitted over a binary-input additive white Gaussian noise channel (AWGNC) with mean zero and variance (noise power) σ^2. Denote the ith code bit ($i = 1, 2, ..., N$) by $c_i \in \{0, 1\}$. The transmitted signal corresponding to this code bit equals $(-1)^{(c_i+1)}$. With an all-zero codeword transmitted, the received vector $\boldsymbol{x} = [x_1, x_2, ..., x_i, ..., x_N]$ is given by $\boldsymbol{x} = -1 + \boldsymbol{z}$, where $\boldsymbol{z} = [z_1, z_2, ..., z_i, ..., z_N]$ is an independent and identically distributed sequence with probability density function (pdf) $\frac{1}{(2\pi)^{N/2}\sigma^N} \exp(-\frac{1}{2\sigma^2} \sum_{i=1}^N z_i^2)$ for $i = 1, 2, ..., N$.

Suppose a message-passing algorithm [5] is used in the iterative decoder to decode the LDPC codes with input $\boldsymbol{x}$. To apply IS to LDPC codes at high SNR region, the error region is divided into independent sub-regions with respect to TSs. For a *regular* code, TSs with the same label $[w; u]$ are considered equivalent and categorized into one class.

The simulated error probability of a representative randomly picked from each class is obtained by employing the IS simulator. For each sub-region associated with a $[w; u]$ TS, the mean of the original density function is biased to $[\mu_1, \mu_2, ..., \mu_i, ..., \mu_N]$ based on MT. Moreover, to ensure that the mean of the biased density is on the boundary of the error region, the value of μ_i will be set as follows. If the ith bit is in the TS, $\mu_i = \mu$; otherwise $\mu_i = 0$.

Finally, the cumulative BLER of each class is obtained by multiplying the simulated error probability of its representative with the total number of elements in the class, and the overall BLER of the LDPC code is calculated by adding the cumulative BLERs of all classes.

Unfortunately, the IS scheme proposed for regular LDPC codes is not applicable to irregular codes. The main reason is that TSs with the same label $[w; u]$ can be configured very differently in terms of their induced connected subgraphs. Because of the discrepancy in configurations, these TSs will also give rise to quite different contributions to the error floor.

III. PROPOSED IS SCHEME FOR IRREGULAR LDPC CODES

A. Search for detrimental PTS-ICSs

In the first stage of the proposed IS scheme, we aim to search as many detrimental PTS-ICSs as possible within the code graph by using a three-step search method.

1) Step one: In [6], "approximate cycle extrinsic message degree (ACE)" has been defined as a metric to measure the number of check nodes singly connected to a cycle. In particular, the ACE of a cycle with length $2l$ equals $\sum_{i=1}^{i=l}(d_i - 2)$, where d_i is the degree of the ith variable node in this cycle. Since detrimental PTS-ICSs contain a single cycle or multiple cycles formed by a combination of short cycles, identifying single, short cycles with few extrinsic edges will facilitate us searching for detrimental PTS-ICSs. In other words, we should search for (single) short cycles with small ACE values, which can be accomplished using the method in [6]. In the following, we briefly outline the procedures.

- Set the depth threshold (θ) and the ACE threshold (χ).
- Sort the variable nodes according to their degrees in non-decreasing order, i.e., $d_i \leq d_j$, if $i \leq j$.

- Initialize $i = 1$.
- A subgraph is obtained by traversing the bipartite graph of the LDPC codes breadth-first from the ith variable node to depth θ.
- Search the subgraph and list all the cycles originating from and terminated at the ith variable node with ACE less than χ.
- Set $i = i + 1$ and repeat searching for the cycles, until all the variable nodes have been considered, i.e., $i = N$.

Note that to avoid repetitive counting, the subgraph expanded from the ith variable node excludes all the variable nodes with index less than i. At the end of the procedures, we will have recorded a number of (single) short cycles with ACE values less than χ. These cycles are indeed PTS-ICSs with a single cycle.

From the results in [7], it can be observed that all of the variable-degree distributions optimized by the density evolution algorithm contain degree-2 variable nodes. However, an irregular code containing degree-2 variable nodes, if not well constructed, may contain many detrimental $[w; u; e]$ PTS-ICSs with $u < 2$. Suppose $[w; u; e]$ PTS-ICSs with $w > \psi$ are not so harmful to the decoder because the chance that errors happen to all the variable nodes in PTS-ICSs with $w > \psi$ is low. Then, in order to find out the most detrimental PTS-ICSs, we need to set the depth threshold (θ) to be large enough such that $[w; 0; e]$ and $[w; 1; e]$ single- or multiple-cycle PTS-ICSs with $w \leq \psi$ can be found in future steps. Furthermore, χ should be set to strike a balance between the computation time and accuracy in heuristic manner. (Details not shown due to shortage of space.)

2) Step two: Suppose W single cycles have been discovered in the previous step. To identify all the possible detrimental PTS-ICSs with multiple cycles, one way is to consider the single cycles under all kinds of combinations, which will result in a total of $\binom{W}{2} + \binom{W}{3} + ... \binom{W}{W}$ possibilities. However, this is a prohibitive number due to the very large W. Instead of using the above computation-intensive method, we apply a multi-bit error impulse technique which creates impulse errors to all the variable nodes in a single cycle. Details of the technique are described as follows.

Consider one of the single cycles. We apply some unnatural noises, called "error impulses" with amplitude ε, to all the bit positions in the cycle. We also scale the remaining bits in the codeword by a relatively smaller parameter α [8]. Without loss of generality, we assume that the first few bits of the codeword are involved in the single cycle. Then, when an all-zero codeword is transmitted, the deterministic input to the decoder can be represented by

$$\boldsymbol{x}_e = [\underbrace{-1+\varepsilon, -1+\varepsilon, \ldots, -1+\varepsilon}_{\text{bits in the single cycle}}, \underbrace{-\alpha, -\alpha, \ldots, -\alpha, -\alpha}_{\text{bits not in the single cycle}}]. \quad (8)$$

Then, we run the decoder with the deterministic input $\boldsymbol{x}_e$. For cases where the decoder reaches the maximum number of iteration and fails to find the valid codeword, we record the hard decision vector $\hat{\boldsymbol{x}}_I$ at each iteration where $I = 1, 2, \ldots, I_{\max}$ and $I_{\max}$ denotes the maximum iteration number.

3) Step three: In the previous step, we have recorded sequences of hard decision vectors when the decoder fails to converge. Consider a particular sequence. For each vector $\hat{\boldsymbol{x}}_I$ ($I = 1, 2, \ldots, I_{\max}$) in the sequence, a PTS or PTS-ICS can be observed by considering the set of variable nodes decoded as "1". However, to locate the PTS or PTS-ICS most detrimental to the decoder, not only the final state of the decoder but also the sequence of hard decision vectors should be taken into account. In [8], the authors have selected the target TS as the set of variable nodes corresponding to bits decoded as "1" in $\tilde{\boldsymbol{x}}$, where

$$\tilde{\boldsymbol{x}} = \min_{I} \omega_H(\hat{\boldsymbol{x}}_I \boldsymbol{H}^T), \quad (9)$$

$\boldsymbol{H}$ represents the code matrix and $\omega_H(\cdot)$ denotes the hamming weight of a vector. But then, in this manner, some redundant TSs or TS-ICSs may be recorded. In our proposed method, to ensure that redundant PTSs or PTS-ICSs will not be counted, we select PTS-ICSs as follows. First, we select $\tilde{\boldsymbol{x}}$ using (9). Then, we form the PTS-ICS from the set of variable nodes decoded as "1" in $\tilde{\boldsymbol{x}}$. Next, we modify the PTS-ICS by removing all the variable nodes not in the cycles and their associated edges. Finally, if the modified PTS-ICS does not exist in our detrimental-PTS-ICS list, it is added to the list. By the end of this stage, we will have compiled a list of detrimental PTS-ICSs.

B. Classification of detrimental PTS-ICSs

Assume that a total of Ω detrimental (both single-cycle and multiple-cycle) PTS-ICSs have been found in the previous stage. We then classify the PTS-ICSs into the same group if they have the same label $[w; u; e]$. Suppose a total of D PTS-ICS groups are formed. We can label these groups with $[w_m; u_m; e_m]$ where $m = 1, 2, \ldots, D$. Note also that there is no guarantee that PTS-ICSs with the same label $[w_m; u_m; e_m]$ (i.e., in the same PTS-ICS group) are configured equivalently under the graph of irregular codes. Further delicate classifications of the PTS-ICSs may result a better accuracy in error-floor prediction, but also give rise to higher computation complexity.

C. Implementation of IS on PTS-ICSs

From our simulation results (not shown due to space limitation), we find that error probabilities of PTS-ICSs in the same group, though may not be the same, lie within the same order. Based on the observation, we propose a two step approach in providing a better estimation of the error-floor without delicate classifications of PTS-ICSs.

1) Rough estimation: For each group of PTS-ICS, we randomly select one representative and use an IS simulator to estimate its BLER. Hence, we obtain a total of D different error probabilities, denoted by $\hat{P}_{\text{IS}}(m), m = 1, 2, \ldots, D$.

2) Fine estimation: Suppose the largest value among the BLERs found in the previous step ($\hat{P}_{\text{IS}}(m), m = 1, 2, \ldots, D$) is of the order 10^{-J}, where J is a positive integer. We then further consider groups with representatives giving rise to error probabilities larger than $10^{-(J+1)}$. Denote the set of the indices of such groups by $\mathbb{S}_{\text{fine}}$. For Group m' where

$m' \in \mathbb{S}_{\text{fine}}$, we will apply IS simulator to estimate the error probabilities of all its PTS-ICS elements. Denoting the error probability of the kth element in Group m' by $\hat{P}_{\text{IS}}(m', k)$, the overall BLER, $\bar{P}_{\text{IS}}$, of the code can then be estimated using

$$\bar{P}_{\text{IS}} = \sum_{m' \in \mathbb{S}_{\text{fine}}} \sum_{k} \hat{P}_{\text{IS}}(m', k) + \sum_{m=1, m \notin \mathbb{S}_{\text{fine}}}^{D} \mathbb{M}_m \hat{P}_{\text{IS}}(m) \quad (10)$$

where $\mathbb{M}_m$ represents the multiplicities of Group m.

IV. Results and discussion

We apply the proposed PTS-ICS search method to three irregular LDPC codes of block length 1008 and code rate 0.5. The degree distributions of the codes follow those optimized by density evolution [7]. Specifically, the first two codes follow the variable-node degree distribution

$$\lambda_1(x) = 0.23802x + 0.20907x^2 + 0.03492x^3 + 0.12015x^4 + 0.01587x^6 + 0.00480x^{13} + 0.37627x^{14}, \quad (11)$$

and the third one follows the variable-node degree distribution

$$\lambda_2(x) = 0.27165x + 0.25105x^2 + 0.30938x^3 + 0.00104x^4 + 0.43853x^9, \quad (12)$$

which have maximum variable-node degrees 15 and 10, respectively. Further, we abbreviate the distributions by "DE15" and "DE10", respectively The first code used in our study, which follows "DE15" variable-node-degree distribution, is actually the "PEGirReg504x1008" code selected from the database at the MacKay website [9]. Here, we rename it as "PEG-only-DE15" to emphasize that it has been constructed using the progressive-edge-growth (PEG) algorithm [10]. The second code, with the abbreviation "PEG-ACSE-DE15", follows "DE15" variable-node-degree distribution and is built using our proposed PEG-ACSE construction method in [2]. Finally, the code "PEG-only-DE10" is constructed with the PEG algorithm with "DE10" variable-node-degree distribution.

First, we run the MC simulations on each of the codes at SNR=2.8 dB until we have collected 100 decoding failures, which is defined as the decoder failing to converge within 50 iterations. We observe that most of the failures are due to single- or multiple-cycle $[w; u; e]$ PTS-ICSs with $w < 17$. Moreover, among the multiple-cycle PTS-ICSs related to the failures, most are formed by combinations of single cycles with ACE values not larger than 6. Therefore, during the search for single cycles in the codes, we set $\psi = 16$ and $\chi = 6$. In Table I, we show some results about the PTS-ICSs found. Next, we select ε and α with an aim to maximizing the number of detrimental multiple-cycle PTS-ICSs found. In Table II, we present the parameters used and the total number of detrimental PTS-ICSs (Ω) and the number of PTS-ICS groups (D) found for the three irregular codes. Fig. 1 presents the multiplicities of multiple-cycle ($e > 0$) $[w; u; e]$ PTS-ICSs with $w < 17$ and $u = 0, 1$ found in the codes.

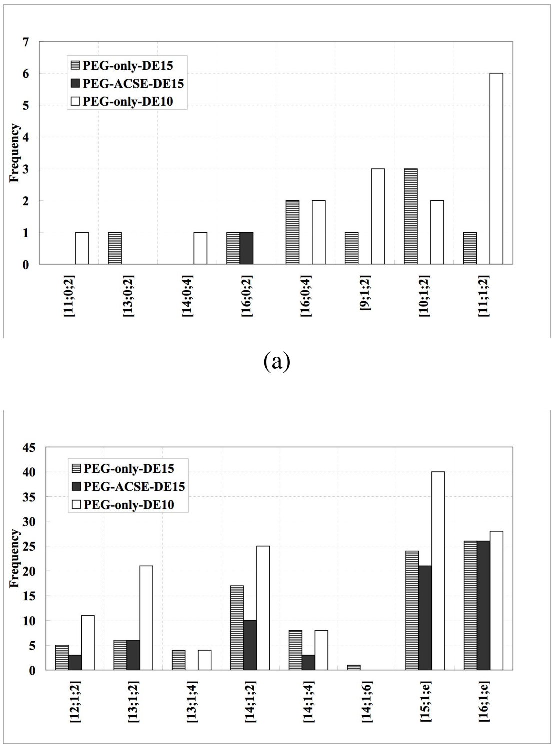
For each code, we then apply the IS simulation to a representative PTS-ICS from each of the PTS-ICS groups. We assume an all-zero codeword being transmitted. Also, the ith input is from shifted $-1 + z_i$ to $-1 + \mu_i + z_i$ ($i = 1, 2, ..., N$), where z_i is a random AWGN noise sample. To ensure that the mean of the biased density is on the boundary of the error region, the value of μ_i will be set as follows. If the ith bit is in the PTS-ICS, $\mu_i = \mu$; otherwise $\mu_i = 0$. The decoder will iterate a maximum of 50 times for each input vector. Further, it has been reported that for binary AWGN channel, the optimal biased-density center should reside close to the boundary between the error region and the region that can be decoded successfully [11]. Hence, we set $\mu = 1$ for $[w; 0; e]$ detrimental PTS-ICSs. For $[w; u; e]$ detrimental PTS-ICSs with $u > 0$, μ has to be bigger than one and a bisection technique is further applied to search for the optimal μ in (μ_l, μ_u) where μ_l and μ_u are set to 1 and 3.2, respectively.

In Fig. 2, we present the BLERs of the three irregular codes estimated by both the standard MC technique and our proposed IS technique. It is clear that the Code 'PEG-ACSE-DE15" provides the best performance, while "PEG-only-DE10" is

TABLE II

Parameters Used and Results in the Search of detrimental PTS-ICSs in the irregular codes.

Code	PEG-only-DE15	PEG-ACSE-DE15	PEG-only-DE10
ε	1.7	1.5	1.7
α	0.50	0.35	0.60
Ω	43,071	43,468	61,765
D	155	153	138



(a)

(b)

Fig. 1. Multiplicities of multiple-cycle $[w; u; e]$ PTS-ICSs with $w < 17$ and $u = 0, 1$ ($e > 0$) in Codes "PEG-only-DE15", "PEG-ACSE-DE15" and "PEG-only-DE10".

TABLE I

MULTIPLICITIES OF SINGLE-CYCLE ($[w;u;0]$) PTS-ICSS WITH $w < 11$ AND $u = 1,2$ IN CODES "PEG-ONLY-DE15", "PEG-ACSE-DE15" AND "PEG-ONLY-DE10".

Code	W	Number of Groups	$[7;1;0]$	$[8;1;0]$	$[9;1;0]$	$[10;1;0]$	$[6;2;0]$	$[7;2;0]$	$[8;2;0]$	$[9;2;0]$	$[10;2;0]$
PEG-only-DE15	39,221	26	3	5	1	13	0	89	126	63	132
PEG-ACSE-DE15	39,845	24	0	6	3	19	0	0	116	58	121
PEG-only-DE10	57,738	26	9	2	2	9	57	207	163	105	103

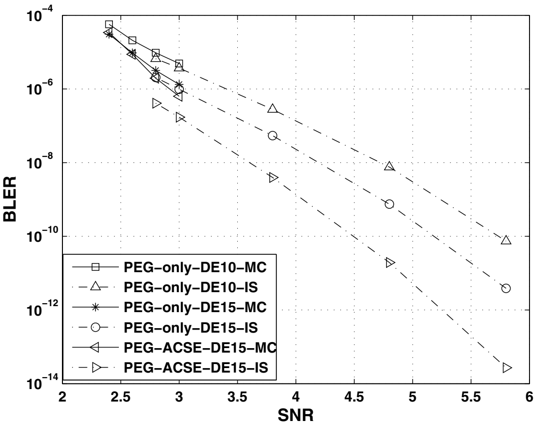


Fig. 2. Block error rates (BLERs) obtained by the standard MC technique and our proposed IS technique for Codes "PEG-only-DE15", "PEG-ACSE-DE15" and "PEG-only-DE10".

TABLE III

NUMBER OF SIMULATION RUNS FOR THE STANDARD MC TECHNIQUE AND THE PROPOSED IS TECHNIQUE FOR DIFFERENT CODES. (N_{MC} CALCULATED FROM $100/\hat{P}_{\mathrm{IS}}$ IS ATTACHED WITH THE TAG "$*$")

Code	**SNR (dB)**	N_{MC}	N_{IS}
PEG-only-DE15	2.8	2.84×10^7	8.69×10^5
	3.0	6.50×10^7	7.58×10^5
	3.8	$1.83 \times 10^9*$	8.07×10^5
	4.8	$1.34 \times 10^{11}*$	5.97×10^5
	5.8	$2.58 \times 10^{13}*$	4.00×10^5
PEG-ACSE-DE15	2.8	5.00×10^7	1.19×10^6
	3.0	1.55×10^8	1.11×10^6
	3.8	$2.35 \times 10^{10}*$	1.03×10^6
	4.8	$4.95 \times 10^{12}*$	8.92×10^5
	5.8	$3.51 \times 10^{15}*$	9.07×10^5
PEG-only-DE10	2.8	1.01×10^7	5.49×10^5
	3.0	2.07×10^7	5.56×10^5
	3.8	$3.59 \times 10^8*$	5.38×10^5
	4.8	$1.31 \times 10^{10}*$	2.01×10^5
	5.8	$1.35 \times 10^{12}*$	1.68×10^5

the worst. Table III shows the number of simulation runs (up to 2 decimal places) for the standard MC technique and the proposed IS technique for the three codes at different SNR values. The standard MC simulator terminates until 100 decoding failures are collected. But for extremely low BLERs, instead of running the MC directly, the number of MC simulation runs (N_{MC}) is estimated using $100/\hat{P}_{\mathrm{IS}}$. The results show that the proposed IS technique can estimate the BLER with much less simulation runs. In particular, a speed-up gain of 3.92×10^9 (= $3.51 \times 10^{15}/9.07 \times 10^5$) is achieved at SNR=5.8 dB for Code "PEG-ACSE-DE15".

V. CONCLUSION

Given a particular *irregular* low-density parity-check (LDPC) code. We have developed in this paper a three-step method to search for as many detrimental primary trapping set-induced connected subgraphs (PTS-ICSs) as possible in this code. Then we classify the PTS-ICSs based on the label $[w;u;e]$ (w denotes the number of variable nodes in the TS; u represents the number of check nodes with odd number of connections to the TS; and e is the cycle indicator). Furthermore, we propose a two-step IS simulator to estimate the BLER of the code at the high signal-to-noise ratio (SNR) region. Results show that our proposed scheme can achieve a good accuracy of BLER estimation compared with the standard MC technique. In addition, speed-up gains of up to 3.92×10^9 times can be achieved.

REFERENCES

[1] T. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annu. Allerton Conf. on Communication, Control, and Computing*, pp. 1426–1435, Urbana-Champaign, USA, 2003.

[2] X. Zheng, F. Lau, and C. Tse, "Categorization of trapping sets based on their induced connected subgraphs," *IEEE Communications Letters*, submitted.

[3] P. Smith and H. Gao, "Quick simulation: a review of importance sampling techniques in communications systems," *IEEE Journal on Selected Areas in Communication*, vol. 15, no. 4, pp. 597–613, 1997.

[4] E. Cavus, C. Haymes, and B. Daneshrad, "An IS simulation technique for very low BER performance evaluation of LDPC code," in *Proc. IEEE Int. Conf. Commun.*, pp. 1095–1100, Istanbul, Turkey, 2006.

[5] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[6] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, 2004.

[7] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-checkcodes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[8] C. Cole, S. Wilson, E. Hall, and T. Giallorenzi, "A general method for finding low error rates of LDPC codes," *Arxiv preprint cs/0605051*, 2006.

[9] D. Mackay, "Encyclopedia of sparse graph codes," 2005.

[10] X. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[11] B. Xia and W. Ryan, "On importance sampling for linear block codes," in *Proc. IEEE Int. Conf. Commun.*, pp. 2904–2908, Anchorage, USA, 2003.