Design of High-Rate Irregular Non-binary LDPC Codes Using Algorithmic Stopping-Set Cancellation

Sudarsan V. S. Ranganathan
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California 90095
Email: sudarsanvsr@ucla.edu

Dariush Divsalar
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109
Dariush.Divsalar@jpl.nasa.gov

Kasra Vakilinia, Richard D. Wesel
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California 90095
vakiliniak@ucla.edu, wesel@ee.ucla.edu

***Abstract*—Following Poulliat et al.'s design of $(2, d_c)$ non-binary LDPC (NB-LDPC) codes, this paper designs high-rate irregular NB-LDPC codes by addressing the problem of minimum symbol distance. The design procedure first identifies all stopping sets up to weight five in an LDPC code and enumerates them via a message passing algorithm. For each identified stopping set, careful labeling forces its corresponding parity-check sub-matrix to be full rank, thereby preventing the stopping set from being a sub-code and ensuring a minimum distance of at least six symbols. Simulation results for codes designed through this procedure show a significant improvement in the error-floor region over randomized labeling.**

I. Introduction

Low-density parity-check (LDPC) codes are a class of linear block codes with sparse parity-check matrices that were discovered by Gallager [1] and later shown to approach capacity [2]. LDPC code design techniques can be categorized as being *algebraic* or *non-algebraic*. Algebraic techniques design the code analytically as in [3] and others. Non-algebraic techniques, which are the focus of this paper, construct codes by optimizing some of the properties of the code's underlying graph such as *girth, degree distribution, cycle extrinsic message degree* etc. [4], [5]. Non-zero labels for the parity-check matrix are then chosen from a finite Galois field GF(q) with q > 2 yielding non-binary LDPC (NB-LDPC) codes [6].

Proper selection of non-zero labels for an NB-LDPC code can lead to a better performance than randomized labeling in both the waterfall and the error-floor regions. In this regard, optimum labeling schemes for $(2, d_c)$ codes were identified through optimization in [7], where *cycle cancellation* and *stopping-set mitigation* were employed to avoid low-weight sub-codes. For high-rate codes ($r \geq 0.8$) with a moderate block-length of around $n = 2000$ bits that are the focus of this paper, the techniques of [7] cannot be applied to improve the minimum row distance beyond two bits for low values of q because of check nodes with high degrees.

This paper demonstrates a code design technique that improves the error floor of high-rate, irregular NB-LDPC codes over random labeling. Our contributions include an exhaustive listing of stopping-set structures up to weight five and an algorithm, based on [8], to enumerate them. We extend cycle cancellation [7] to *stopping-set cancellation (SSC)* to prevent these stopping sets from being sub-codes. We demonstrate these steps by designing a protograph-based code with a low iterative-decoding-threshold [9] degree distribution. Significant improvement is seen in the error-floor region as a result of increase in minimum symbol distance of the NB-LDPC code.

The paper is organized as follows: Section II presents background material and notation. Section III lists all stopping-set structures up to weight five of LDPC codes. Section IV presents the enumeration algorithm using two particular structures as examples. Section V presents an example design of a protograph-based NB-LDPC code through SSC and follows up with simulation results. Section VI concludes the paper.

II. Background and Notation

An LDPC code is defined by a sparse parity-check matrix H containing m rows and n columns, whose entries belong to a finite field GF(q). The rate of the code is given by $r \geq \frac{n-m}{n}$, with equality if H is full rank. A bipartite undirected Tanner graph $G(V, C, E)$ is an alternative representation for H. The columns of H form the variable-node set V and the rows of H form the check-node set C. The edge set E has elements $e_{ij} : H_{ji} > 0$ which connect elements from V with elements from C. The number of edges connected to a node is its degree. The girth g of a graph is the length of its shortest cycle.

A *protograph* [10] $G_p(V_p, C_p, E_p)$ is a Tanner graph with $n_p = |V_p|$ and $m_p = |C_p|$, where n_p and m_p are relatively small. The edge set E_P allows multiple edges connecting a variable-check node pair. To obtain a code $G(V, C, E)$, a protograph G_p is *lifted* by a factor N through two operations. G_p is first copied N times to produce $n = n_pN$ variable nodes, $m = m_pN$ check nodes and $|E| = |E_p|N$ edges. The variable node connections are then permuted within each set of N edge copies to obtain the graph $G(V, C, E)$. The edges are labeled with non-zero elements from GF(q) for an NB-LDPC code. Practical implementation advantages are obtained when edge permutations form circulant sub-matrices in H.

A *stopping set* S is a set of a variable nodes whose b neighboring check nodes have at least two edges connected

This research was supported by a gift from the Broadcom Foundation and National Science Foundation (NSF) grants CCF-1162501 and CCF-1161822 (JPL Task Plan 82-17473). This research was carried out in part at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with NASA.

to S. With no variable nodes of degree 1, a stopping set is always a cycle or an interconnection of cycles [5]. The sub-matrix H_S corresponding to the stopping set is of size $b \times a$, where a is the *weight* of the stopping set. A *sub-code* is a stopping set that has a non-trivial null-space associated with its sub-matrix. We categorize a stopping set as wide, square, or tall depending upon the structure of its sub-matrix as follows:

1) A *wide stopping set* has $a > b$ and is always a sub-code since a wide sub-matrix H_s always has a non-trivial null-space. These stopping sets are *"unavoidable"* sub-codes as labeling cannot prevent them from being sub-codes.
2) A stopping set is *square* if $a = b$ and *tall* if $a < b$. In an NB-LDPC code, these stopping sets are sub-codes only if their sub-matrices are rank-deficient. They can be prevented from being sub-codes through proper labeling.

For an NB-LDPC code, the weight of the smallest stopping-set sub-code in H is the minimum symbol distance. The labeling scheme introduced in [7] for $(2, d_c)$ NB-LDPC codes prevented square stopping sets (cycles) from being sub-codes and maximized the minimum distance of wide sub-codes yielding an improved performance. Empirically, for our high rate and moderate block-length, irregular NB-LDPC codes that are labeled randomly have a high error floor for degree distributions with low iterative-decoding thresholds mainly due to very low-weight tall and square stopping-set sub-codes.

III. Stopping-Set Structures up to Weight Five

This section catalogs[1] *all* stopping-set structures up to weight five in LDPC codes. The catalog facilitates the identification of stopping sets through a simple algorithm, thereafter which tall and square stopping sets can be labeled carefully for NB-LDPC codes. We denote square, wide and tall stopping sets as ss, sw and st respectively. In all figures, circles represent variable nodes and squares represent check nodes. O and T are labels specific to the algorithm of Sec. IV.

Remark 1: In this work, we consider LDPC codes with girth $g > 4$ without variable nodes of degree 1.

Lemma 1: Any stopping set of weight a has variable nodes of degrees at most $a - 1$.

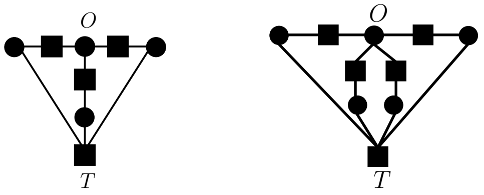
Proof: A variable node of degree a needs at least a other variable nodes to form a stopping set with girth $g > 4$. ■

A. Square Stopping-Set Structures up to Weight Five

Lemma 2: All variable nodes in weight-three structures have degree 2. There are no tall structures of weight three.

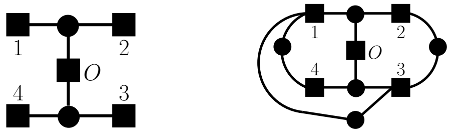
Proof: The first part of the Lemma follows from Lemma 1. The proof of the second part is described in [7] as a structural property of $(2, d_c)$ LDPC codes. As a result, every weight-three stopping set is a length-six cycle formed by variable nodes of degree 2. We call this structure $ss1$. ■

There are two square structures of weight four. The first, $ss2$, is a simple length-eight cycle similar to $ss1$. The other structure, $ss3$, is shown in Fig. 1(a).



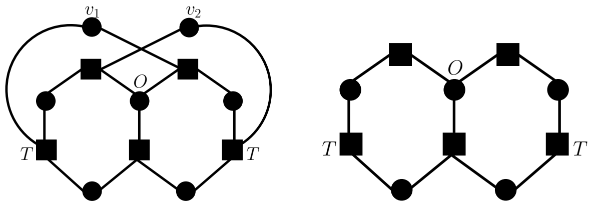
(a) $ss3$ (b) $ss5$

Fig. 1. Two non-trivial square stopping-set structures



(a) Sub-graph (b) Example of $ss6$

Fig. 2. The square stopping-set structure $ss6$ contains the sub-graph shown in (a). An example $ss6$ stopping set is shown in (b).



(a) A weight-seven sub-code (b) $ss7$

Fig. 3. A sub-code and its constituent square structure of weight five

Lemma 3: In an LDPC code with girth $g > 4$, a weight-a square structure cannot contain variable nodes of degree both $a - 2$ and $a - 1$. There can be at most two variable nodes of degree $a - 2$ or one variable node of degree $a - 1$.

Proof: Any violation of the lemma leads to a length-four cycle in an $a \times a$ sub-matrix which is a contradiction. ■

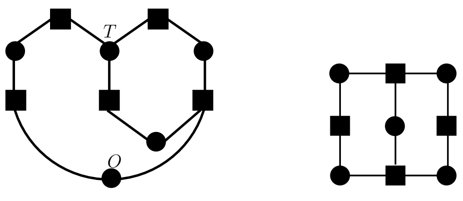
The remaining square structures all have weight $a = 5$. For all variable nodes having degree 2, the only possible configuration is $ss4$, a simple length-ten cycle. For one variable node having degree 4, the only possible structure is $ss5$ (similar to $ss3$), which is shown in Fig. 1(b).

Fig. 2(a) shows the only way two variable nodes of degree 3 can be present when $a = 5$ and $g > 4$. If check nodes 1, 2, 3 and 4 form at least three of the four possible connections through variable nodes of degree 2, we have square structure $ss6$, an example of which is shown in Fig. 2(b). Presence of all the four possible connections leads to four $ss6$ stopping sets and a weight-six sub-code.

Now we consider the weight-five square structures with one variable node of degree 3. The weight-seven sub-code shown in Fig. 3(a) has six square stopping sets of weight five of the form shown in Fig. 3(b) which is denoted $ss7$. Removing either v_1 or v_2 in Fig. 3(a) leads to a weight-six sub-code which contains only two $ss7$ structures.

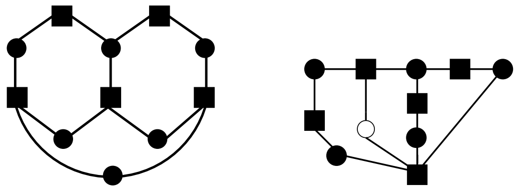
The next square structure, $ss8$, is shown in Fig. 4(a). The wide stopping set $sw1$ in Fig. 5(a) is a union of $ss1$ and $ss7$ and it contains two $ss8$ stopping sets. Stopping set $sw2$, shown in Fig. 5(b), is a union of $ss1$ and $ss3$. Removing the unfilled

[1] See http://www.seas.ucla.edu/csl/resources/manuscripts/ISIT_2014_ss_catalog.pdf for a complete catalog of these structures.



(a) $ss8$ (b) Sub-code

Fig. 4. Two weight-five stopping-set structures



(a) $sw1$ (b) $sw2$

Fig. 5. Two wide structures of weight six

variable node from $sw2$ leads to an $ss8$ stopping set.

Lemma 4: $ss7$ and $ss8$ are the only square structures of weight five with only one variable node of degree 3.

Proof: A weight-five square structure with only one variable node of degree 3 contains exactly two cycles and the possible cycle lengths are six and eight, where at least one cycle must be of length six. More cycles or longer cycles force a higher-weight structure. Two length-six cycles lead to $ss7$ and one cycle each of length six and eight lead to $ss8$. ■

B. The Only Wide Weight-Five Stopping-Set Structure

A 3×4 wide structure cannot be present in an LDPC code with girth $g > 4$. Moreover, a 4×5 stopping set has only variable nodes of degree 2 in order to satisfy $g > 4$. For a code with girth $g > 4$, the lowest possible weight of a wide structure with variable nodes of degree 2 is 5 and is achieved by the structure shown in Fig. 4(b). This structure should be avoided while constructing the code in order to achieve a minimum distance of six symbols since any labeling leads to a sub-code.

C. Tall Stopping-Set Structures up to Weight Five

Recall that Lemma 2 precludes weight-three tall structures and hence we begin the discussion with weight-four tall structures. We will examine the possible variable node degrees in a tall structure, which we identify by a list of degrees. For example the list 3322 denotes a structure with two variable nodes of degree 3 and two variable nodes of degree 2.

Lemma 5: A stopping-set structure with only one variable node of degree larger than 2 is either square or wide.

Proof: Since structures with only variable nodes of degree 2 are either square or wide [7], addition of another variable node of a higher degree can only lead to a square or a wide structure since no new check nodes are introduced in this process. A square structure is formed if the variable nodes of degree 2 do not contain any stopping set to begin with. ■

We start with the degree list 3322. The following Lemma, referred to as the *sharing constraint*, will be used in this section whenever required.

Lemma 6: Two variable nodes of degree 3 in a weight-four tall structure share exactly one check node.

Proof: If these two variable nodes do not share exactly one check node, two additional variable nodes are not sufficient to complete the structure with girth $g > 4$. ■

$$\begin{bmatrix}1 & 0\\ 1 & 0\\ 1 & 1\\ 0 & 1\\ 0 & 1\end{bmatrix} \overset{\text{completion}}{\Longrightarrow} \begin{bmatrix}1 & 0 & 1 & 0\\ 1 & 0 & 0 & 1\\ 1 & 1 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 1 & 0 & 1\end{bmatrix} \qquad (st1)$$

Consider the two variable nodes of degree 3 that share a check node. The above sub-matrix (on the left), upon completion, leads to a unique stopping-set structure for 3322 and this structure is denoted $st1$.

Similar to the above argument, it can be shown by constructing a sub-matrix that 3332 is not a valid configuration. The 3333 stopping set, denoted $st2$, is a unique structure that is a obtained from $st1$ by *"promoting"* both variable nodes of degree 2 to degree 3 through one additional check node.

Now we consider the weight-five tall stopping-set structures.

- 33222: We consider first the case where the two variable nodes of degree 3 do not share a check node. As shown below, a unique completion leads to the tall structure $st3$.

$$\begin{bmatrix}1 & 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0 & 1\\ 0 & 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 1\end{bmatrix} (st3), \quad \begin{bmatrix}1 & 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 0\\ 1 & 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 1\\ 0 & 0 & 1 & 0 & 1\end{bmatrix} (st4)$$

  For the case where the two variable nodes of degree 3 share one check node, $st4$, shown above and in Fig. 6(b), is the only unique completion.
- 33322: There are two non-isomorphic structures $st5$ and $st6$ with the same number of check nodes. The set of three variable nodes of degree 3 has to satisfy the *sharing constraint* in such a way that they share exactly one check node between each pair. This leads to the two possible structures which are not described here and can be seen through sub-matrices similar to the above ones.
- $33332, 33333$: 33332 has two non-isomorphic structures which essentially differ in the number of check nodes that are present (unlike $st5, st6$). The first, $st7$, is a simple extension of $st2$. The second structure, denoted $st8$, has one check node more than $st7$. The structure corresponding to 33333, denoted $st9$, is shown in Fig. 6(a).

For tall structures of weight five with variable node(s) of degree 4, the *sharing constraint* is that a variable node of degree 4 shares exactly one check node with every other variable node in the structure. Because of this constraint, the following sub-matrix, without loss of generality, will always be present in these structures:

$$\begin{bmatrix}1 & 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0 & 1\\ 1 & 1 & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\end{bmatrix}. \qquad (1)$$

The resulting tall structures with variable nodes of degree 4 are as follows: 43222-$st10$, 43322-$st11$, 43332-$st12$, 43333-$st13$, 44222-$st14$, 44332-$st15$, 44333-$st16$, 44433-$sr17$, 44444-$st18$. There are no structures for 44322, 44422 and 44432.

Theorem 1: The eight square structures $ss1, ss2, \ldots, ss8$, the one wide structure and the eighteen tall structures $st1, st2, \ldots, st18$ are the only stopping-set structures of weight five or less that can be present in any LDPC code.

Proof: The section's discussion proves the theorem. ■

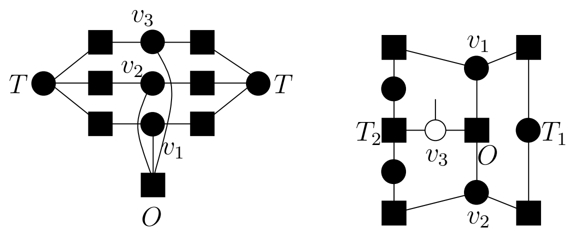
IV. Stopping-Set-Enumeration (SSE) Algorithm

This section presents the stopping-set-enumeration algorithm and provides two examples. The approach generalizes to most structures described in the previous section. The algorithm, inspired by [8], is based on message passing.

For a specified structure, the algorithm runs a message-passing procedure from each potential *origin* node O as denoted in the figures. Depending upon the structure considered, certain types of nodes can be removed from the graph during initialization and the rest of the nodes are *active*. The first step is to pass the index of the potential origin node O, after which a fixed number of message-passing iterations is performed. At each iteration, any active node that receives a message vector appends its own index to the vector and then it passes the modified message along all its edges except the one from which it received the original message. That node is then removed from the graph. The messages at the last nodes (if any) are checked to see if the nodes form valid *termination nodes* T leading to instances of the structure considered.

Example 1: Consider the tall structure $st9$ shown in Fig. 6(a). To enumerate this, the algorithm begins by removing all variable nodes except those with degree 3 from the Tanner graph. Then a message-passing procedure is run once for each check node as a potential origin node O. Three message-passing iterations are needed to traverse from the potential origin O to the possible termination nodes T. If $st9$ is present for that potential O, in the third iteration the two termination nodes T each receive messages along all 3 of their edges that identify the same three variable nodes (v_1, v_2, v_3 in Fig. 6(a)) to form the stopping set.

Example 2: Fig. 6(b) shows tall structure $st4$, which requires a more complicated approach. The possible connectivity between the origin check node O and the termination node T_2 through v_3, which is of degree 3 and *not* part of the stopping set, complicates the message passing. A message passed via $O \rightarrow v_3 \rightarrow T_2$ will cause T_2 to be removed from the graph before the messages from v_1 and v_2 can reach T_2. To overcome this problem, the message-passing procedure must be performed multiple times from each potential origin check node, once for each pair of variable nodes of degree 3 neighboring the check node. By activating one such pair of variable nodes for each message-passing procedure, a check node can discover all $st4$ structures for which it is the origin O. Four iterations are required for each procedure. T_1 will terminate in the third iteration while T_2 terminates in the fourth iteration when an $st4$ structure is found.



(a) Tall structure $st9$ (b) Tall structure $st4$

Fig. 6. Examples of stopping-set enumeration (SSE)

TABLE I
Iterative-Decoding Thresholds for Non-binary Protograph Ensembles with $m_p = 1, n_p = 6$ Using EXIT Charts [9]

Degree Distribution	**16-QAM Threshold** (E_s/N_0 dB)	**BIAWGNC Threshold** (E_b/N_0 dB)
222222	11.19	2.81
322222	11.03	2.73
332222	10.98	2.56
333222	11.02	2.60
333322	11.03	2.69
333332	11.05	2.70
333333	11.18	2.75

V. Code Design Using Stopping-Set Cancellation

Our code design starts with a protograph $G_p(V_p, C_p, E_p)$: $|V_p| = n_p, |C_p| = m_p$ with multiple edges between a variable-check node pair. We lift the protograph through two steps by factors f_1 and f_2 respectively, using circulant permutations, to obtain a code of block-length $n = n_p \times f_1 \times f_2$. The two-stage lifting is due to observations in [11] that indicate a possibility of low minimum distance if circulant permutations are assigned via single-stage lifting (we also observed this empirically). If the largest number of parallel edges in G_p is z, then f_1 should satisfy $f_1 \geq z$ to obtain a protograph G_{p_1} without multiple edges between any variable-check node pair. However, f_1 should be kept as small as possible since larger values of f_2 allow more parallelism in the implementation of the decoder. Thus we choose $m_p = 1$ resulting in the smallest possible n_p and a large f_2.

We consider protographs with variable-node degrees not greater than 3 and select a degree distribution with a low iterative-decoding threshold computed using EXIT charts for protograph-based NB-LDPC codes [9] as shown in Table I. The chosen protograph is of size $m_p = 1, n_p = 6$ and distribution 333222. With the lifting factors chosen as $f_1 = 3$ and $f_2 = 29$, two-stage lifting is performed using the circulant-PEG algorithm avoiding the structure in Fig. 4(b). We thus obtained a parity-check matrix of size 87×522 ($r = 0.833$) with girth $g = 6$. The choice of this protograph over the 332222 which has a better threshold was due to the following:

1) The number of cycles of length 12 with only variable nodes of degree 2 in the lifted code with same parameters for the distribution 332222 was 10382 compared to 1421 of our chosen distribution.
2) Empirically, the code constructed from the distribution 332222 had a high error-floor with randomized labeling.

TABLE II
STOPPING SETS UP TO WEIGHT FIVE OR LESS FOR THE 333222 CODE

ID	$ss2$	$ss4$	$ss7$	$ss8$	$st1$	$st3$	$st4$	$st5/st6$
#	29	377	58	29	29	145	232	29

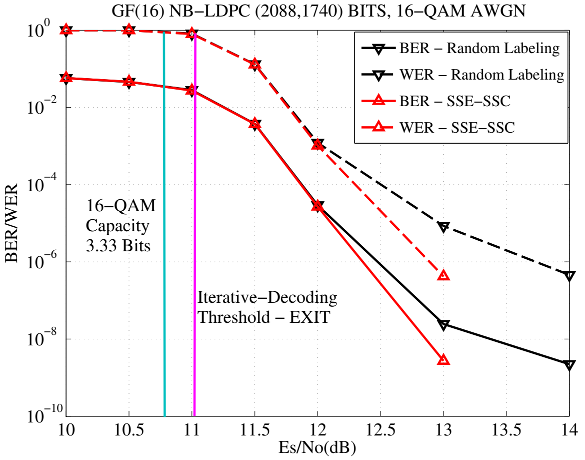


Fig. 7. BER/WER vs. SNR of the protograph-based NB-LDPC code designed through SSE and SSC simulated using 16-QAM over AWGN channel

Table II shows the 928 stopping sets of weight five or less identified using the stopping-set-enumeration algorithm to enumerate each of the structures for our lifted 333222 code. The method of using error impulses [12] to enumerate stopping sets gave the same result, but without a guarantee of completeness and with extraordinarily long running times.

Let $\mathcal{T}$ denote the set of enumerated stopping sets. The sub-matrix H_{S_i} of each stopping set S_i with weight a_i in $\mathcal{T}$ is assigned non-zero elements from GF(16) to satisfy the following full-rank condition (FRC):

$$\text{FRC} : \text{rank}(H_{S_i}) = a_i, \forall i \in \{1, 2, \ldots, |\mathcal{T}|\}. \tag{2}$$

Thus, the minimum distance obtained through this construction is at least six symbols provided all stopping sets in $\mathcal{T}$ are cancelled and the structure of Fig. 4(b) is avoided. The variable nodes not involved in any $S_i \in \mathcal{T}$ are labeled randomly.

Figs. 7 and 8 present word-error rate and bit-error rate performance of the designed code. Simulations were performed using 16-QAM and BPSK over the AWGN channel with a full-precision FFT-QSPA decoder along with a maximum of 100 decoding iterations. Randomized labeling of H that lead to a minimum distance of four symbols (the lifted graph had no $ss1$ structures) is compared against labeling satisfying (2) that lead to a minimum distance of six symbols. We observed error events corresponding to these stated values.

Fig. 7 shows the improvement in the error-floor region for the designed protograph-based NB-LDPC code that has a minimum distance of six symbols. Fig. 8 compares the designed code against an RS-based binary LDPC code [3]. The error floor seen after stopping-set cancellation is still largely due to convergence of decoder to low-weight codewords (weights 6 and 7) indicating possibilities for further improvement.

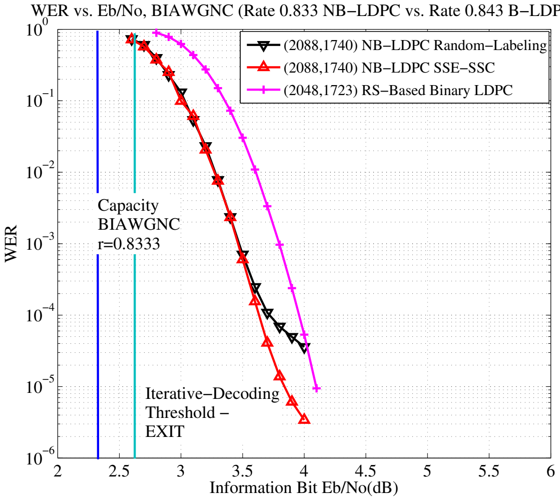


Fig. 8. Comparison with RS-Based Binary-LDPC code over BIAWGNC

VI. CONCLUSION

This paper identifies all stopping-set structures up to weight five of LDPC codes. This facilitates enumeration and subsequent cancellation of all such stopping sets, thereby improving the error floor of a high-rate, irregular protograph-based NB-LDPC code by a factor of ten. Cancellation of all such stopping sets yields a minimum distance of at least six symbols. The stopping-set-enumeration algorithm presented here might be extended to identify larger stopping sets without knowing their structures a-priori and to handle non-codewords.

REFERENCES

[1] R. G. Gallager, "Low-Density Parity-Check Codes," 1963.

[2] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[3] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "A Class of Low-Density Parity-Check Codes Constructed Based on Reed-Solomon Codes with Two Information Symbols," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 317–319, Jul. 2003.

[4] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[5] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Construction of Irregular LDPC Codes with Low Error Floors," in *Proc. IEEE Int. Conf. on Commun.*, vol. 5, May 2003, pp. 3125–3129.

[6] M. Davey and D. MacKay, "Low-Density Parity Check Codes over $GF(q)$," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, Jun. 1998.

[7] C. Poulliat, M. Fossorier, and D. Declercq, "Design of Regular $(2, d_c)$-LDPC Codes over GF(q) Using Their Binary Images," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1626–1635, Oct. 2008.

[8] M. Karimi and A. Banihashemi, "Message-Passing Algorithms for Counting Short Cycles in a Graph," *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 485–495, Feb. 2013.

[9] B.-Y. Chang, L. Dolecek, and D. Divsalar, "Exit Chart Analysis and Design of Non-Binary Protograph-Based LDPC Codes," in *Proc. MILITARY COMMUNICATIONS CONFERENCE*, Nov. 2011, pp. 566–571.

[10] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes Constructed from Protographs," JPL, IPN-PR 42-154, Aug. 2003.

[11] B. Butler and P. Siegel, "Bounds on the Minimum Distance of Punctured Quasi-Cyclic LDPC Codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4584–4597, Jul. 2013.

[12] G. Richter, "Finding Small Stopping Sets in the Tanner Graphs of LDPC Codes," in *Proc. 4th Int. Symp. on Turbo Codes & Related Topics*, Munich, Apr. 2006, pp. 1–5.