On the Combinatorial Substructures of LDPC Codes Causing Error Floors in the AWGN Channel

Hosung Park, Jong-Seon No
Dept. of EECS, INMC
Seoul National University
Seoul 151-744, Korea
Email: {lovepk98,jsno}@snu.ac.kr

Beomkyu Shin
Memory Division
Samsung Electronics
Hwaseong, Gyeonggi-do 445-701, Korea
Email: thechi@gmail.com

Habong Chung
Dept. of EE
Hongik University
Seoul 121-791, Korea
Email: habchung@hongik.ac.kr

***Abstract*—Finite-length low-density parity-check (LDPC) codes usually suffer from error floors in high signal-to-noise ratio (SNR) region. The error floor in the additive white Gaussian noise (AWGN) channel is known to be caused by trapping sets or absorbing sets. In this paper, we investigate combinatorial properties of trapping sets by using graph-theoretic approach. All non-isomorphic trapping sets are identified by a graph-theoretic tool and a method to distinguish the trapping sets which cannot appear in any protograph-based LDPC codes are proposed. Finally, a measure for estimating the harmfulness of trapping sets is proposed by using the linear system model of trapping sets.**

I. Introduction

Low-density parity-check (LDPC) codes [1] have become one of dominant error-correcting codes for many current communication systems due to their near capacity-approaching performance. Since low decoding complexity can be achieved by various iterative decoding algorithms, LDPC codes have been adopted in many practical applications. However, there are still remaining issues for the next-generation high-speed communication systems or data storage systems because finite-length LDPC codes usually suffer from the error floor problem [2] in high signal-to-noise ratio (SNR) region.

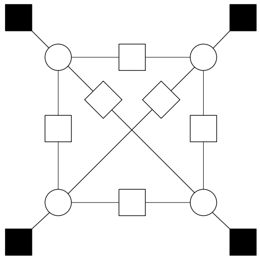
The error floor phenomenon is closely related to some combinatorial structures such as trapping sets [2] and stopping sets [3] under iterative message-passing decoding algorithms. In the binary erasure channel (BEC), stopping sets are the main reason of decoding failure under the belief-propagation (BP) decoding. Actually, it is known that every decoding failure in the BEC can be completely analyzed with stopping sets [3] regardless of the range of erasure probability. In the binary symmetric channel (BSC) and the additive white Gaussian noise (AWGN) channel, decoding failure under iterative decoding algorithms is usually caused by trapping sets, especially, in the error floor region. Trapping sets are known to hinder a corrupted codeword from converging to the correct codeword and severely contribute to the error floor. Also, absorbing sets are similarly defined as trapping sets and the error floor is further investigated with the aid of the absorbing sets [4]. However, despite the efforts of these approaches, analysis of decoding failure in the AWGN channel is not completely solved and it is still an open problem.

To solve the error floor problem in the AWGN channel, it is required to explore the combinatorial properties and the message-passing behavior of trapping sets. There have been lots of efforts to study trapping sets in this literature. In [5], a systematic method to identify trapping sets for the BSC is proposed and the critical number of a trapping set is defined as a measure for the harmfulness of the trapping set in the BSC. Also, the trapping set ontology is provided, which is a database of trapping sets that summarizes the topological relations among trapping sets. In [4], the authors characterize the absorbing sets for the class of array-based LDPC codes and provide the development of techniques to enumerate them exactly. In [6], the topology of the dominant absorbing sets for the IEEE 802.3an LDPC code is identified and tabulated using topological relationships. Also, using a linear system model for the iteration behavior of these sets, the dynamic behavior in the iterative message-passing decoding algorithm is analyzed. As an extension to [6] and [7], a refined linear model is proposed and the error floors are accurately predicted for saturating decoders in [8].

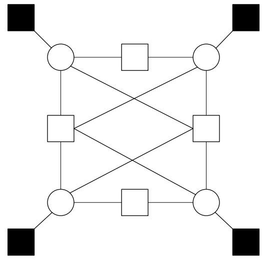
In this paper, we first propose a systematic approach to identify all non-isomorphic trapping sets by using a graph-theoretic tool. Also, among all trapping sets, we find out which types of trapping sets can appear or not in protograph-based LDPC codes regardless of lifting for a given protograph. Finally, a measure for evaluating the harmfulness of trapping sets is proposed by introducing a linear system model for trapping sets and some examples are provided to verify the proposed measure.

II. Identification of Trapping Sets

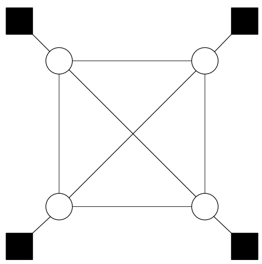
Let H denote a parity-check matrix of LDPC codes. The *Tanner graph* of an LDPC code is a bipartite graph whose incidence matrix is H. A class of nodes in the Tanner graph which correspond to the columns of H are called *variable nodes* and the other class of nodes which correspond to the rows of H are called *check nodes*. Note that the weight of a column (row) in H is equal to the degree of the corresponding variable (check) node. An LDPC code is called *variable-regular* when all variable nodes have the same degree and called *regular* when it is variable-regular and all check nodes



(a) A $(4,4)$ trapping set.



(b) A $(4,4)$ trapping set.



(c) A simplified representation of the $(4,4)$ trapping set.

Fig. 1. Graphical representation of $(4,4)$ trapping sets for $d_v = 4$

also have the same degree. Let d_v denote the degree of variable nodes for a (variable-) regular LDPC code. The *girth* of an LDPC code is defined as the length of the shortest cycle in the Tanner graph.

A trapping set is defined in [2] as the set of variable nodes that are not eventually correct for a given received data from the AWGN channel and it is denoted by (a,b) trapping set or $\mathcal{T}(a,b)$ when there are a variable nodes and b odd degree check nodes neighbors in the sub-graph induced by the trapping set. Later, trapping sets are combinatorially defined in [9] and this paper will follow this definition of trapping sets.

Definition 1: A *trapping set* $\mathcal{T}(a,b)$ is a configuration of a variable nodes, for which the induced subgraph contains $b > 0$ odd-degree check nodes. $\mathcal{T}(a,b)$ is called an *elementary trapping set* if all check nodes in the induced subgraph have either degree one or two, and there are exactly b degree-one check nodes.

Elementary trapping sets are convenient to be analyzed and it is observed that most of trapping sets contributing the error floors are elementary [9], [10]. In this section, all elementary trapping sets will be identified under the assumption that LDPC codes are variable-regular and the girth is at least 6.

As an example for a graphical representation of trapping sets, Fig. 1(a) represents a $(4,4)$ trapping set for $d_v = 4$, where a white circle represents a variable node in the trapping set, and white and black squares represent degree-one and -two check nodes, respectively. The $(4,4)$ trapping set in Fig. 1(b) is not of elementary form while the $(4,4)$ trapping set in Fig. 1(a) is elementary. Since an (a,b) elementary trapping set has always degree-two check nodes except for b degree-one check nodes, there is no confusion if white squares are omitted

TABLE I
ENUMERATION OF SMALL-SIZE NON-ISOMORPHIC ELEMENTARY TRAPPING SETS

$d_v = 3$		$d_v = 4$		$d_v = 5$	
(a,b)	Num	(a,b)	Num	(a,b)	Num
(3,3)	1	(3,6)	1	(3,9)	1
(4,4)	1	(4,8)	1	(4,12)	1
(4,2)	1	(4,6)	1	(4,10)	1
(5,5)	1	(4,4)	1	(4,8)	1
(5,3)	2	(5,10)	1	(5,15)	1
(5,1)	1	(5,8)	3	(5,13)	3
(6,6)	1	(5,6)	3	(5,11)	3
(6,4)	4	(5,4)	2	(5,9)	2
(6,2)	4	(5,2)	1	(5,7)	1
(7,7)	1	(6,12)	1	(5,5)	1
(7,5)	6	(6,10)	5	(6,18)	1
(7,3)	10	(6,8)	10	(6,16)	5
(7,1)	4	(6,6)	11	(6,14)	11
(8,8)	1	(6,4)	7	(6,12)	15
(8,6)	10	(6,2)	3	(6,10)	12
(8,4)	25	-	-	(6,8)	8
(8,2)	19	-	-	(6,6)	5
-	-	-	-	(6,4)	2
-	-	-	-	(6,2)	1

in the graphical representation of an elementary trapping set. Accordingly, the $(4,4)$ trapping set in Fig. 1(a) can be simply represented as Fig. 1(c) and this kind of representation will be called the *simplified representation* of a trapping set.

An (a,b) elementary trapping set does not always exist for any a and b when d_v is given. For a given d_v, Theorem 1 provides some necessary conditions on a and b for the existence of an (a,b) elementary trapping set.

Theorem 1: Consider (a,b) elementary trapping sets with d_v. Then, a and b satisfy that $a \geq 3$, $(d_v+1)a - a^2 \leq b \leq a(d_v-2)$, b has the same parity with a if d_v is odd, and b is even if d_v is even.

Proof: Consider the simplified representation of an (a,b) elementary trapping set. Except for the b edges connected to degree-one check nodes, there are $(ad_v - b)/2$ edges because every edge meets two variable nodes at its both ends. The numerator $ad_v - b$ should be a multiple of 2 and thus b has the same parity with a if d_v is odd and b is even if d_v is even. Since the number of edges except the b edges cannot be larger than that of the complete graph of a nodes, we have $(ad_v - b)/2 \leq \binom{a}{2}$ which yields $(d_v+1)a - a^2 \leq b$. Also, since the number of edges except the b edges cannot be less than a for a connected graph, we have $a \leq (ad_v - b)/2$ which yields $b \leq a(d_v-2)$. Combining with the previous inequality, we have $a \geq 3$. ■

To identify the elementary trapping sets in Theorem 1, a package program "nauty" [11] for graph theory can be exploited, which can generate a specified class of all non-isomorphic simple graphs. It is firstly proposed in [12] to use

TABLE II
ENUMERATION OF NON-ISOMORPHIC ELEMENTARY TRAPPING SETS UNDER GIRTH CONSTRAINTS

a	$g \geq 6$			$g \geq 8$			$g \geq 10$		
	d_v			d_v			d_v		
	3	4	5	3	4	5	3	4	5
3	1	0	0	0	0	0	0	0	0
4	2	1	0	1	0	0	0	0	0
5	4	2	1	2	0	0	1	0	0
6	9	7	3	4	1	0	1	0	0
7	21	22	10	8	1	0	2	0	0
8	55	111	67	22	6	1	5	0	0
9	148	615	634	51	12	1	10	0	0
10	439	4355	9577	153	83	6	29	1	0

"nauty" for identifying elementary trapping sets and then other works also use it [5], [8].

All simplified graphs of elementary trapping sets are generated by setting the number of nodes to a, the number of edges to $(ad_v - b)/2$, and the range of node degrees to $[2, d_v]$. The result for the number of small size of all non-isomorphic elementary trapping sets is shown in Table I. Also, the effect of girth to the number of elementary trapping sets can be seen in Table II under the assumption that every variable node can have at most one degree-one check node.

III. TRAPPING SETS OF PROTOGRAPH-BASED LDPC CODES

LDPC codes are usually adopted in many practical applications by taking a form of protograph-based LDPC codes due to the good efficiency of implementation. Therefore, in this section, we focus on elementary trapping sets which can or cannot appear in protograph-based LDPC codes. We first define protograph-based LDPC codes as follows.

A *protograph* [13] is a bipartite graph for constructing protograph-based LDPC codes. In this section, we consider only simple protographs. The Tanner graph of a protograph-based LDPC code is constructed by copying the protograph a certain number z times and permuting the same z edges. Such copy-and-permute operation is called *lifting* and z is called the *lift size* of the protograph-based LDPC code. Note that quasi-cyclic (QC) LDPC codes are a class of protograph-based LDPC codes using only cyclic permutations.

Definition 2 ([4]): An (a, b) *absorbing set* of the Tanner graph is a set of a variable nodes which induce a subgraph with exactly b odd degree check nodes where all variable nodes neighbor strictly more even-degree check nodes than odd-degree check nodes within the subgraph. Moreover, in a *fully absorbing set*, every variable node outside of the induced subgraph neighbors strictly fewer odd-degree check nodes of the subgraph than other check nodes of the Tanner graph.

In [4], a detailed theoretical analysis of (fully) absorbing sets for array-based LDPC codes are provided. More specifically, all minimal (fully) absorbing sets for the array-based LDPC codes are characterized and the techniques to exactly enumerate them are developed. In this section, by taking different approaches, we are interested in figuring out what kind of elementary trapping sets cannot appear in protograph-based LDPC codes regardless of the lift size and the permutations.

Assume a fully connected simple protograph whose incidence matrix is the $J \times L$ all-1 matrix where $J < L$. It is known that any cycle in a a protograph-based LDPC code can be always mapped into a tailless non-reversing closed (TNC) walk whose permutation shift is the identity permutation in its protograph [14]. Since a TNC walk traversing a node cannot return back to the node within two steps in a simple protograph, two variable (check) nodes apart from each other by 2 in any cycle occurring in protograph-based LDPC codes do not have the same mapping image in the protograph. Equivalently, in the simplified representation of a cycle, no two nodes (edges) sharing the same edge (node) have the same image in the protograph. Note that an edge can be interpreted as a check node in the simplified representation and that, in this section, we will consider the simplified representation related to elementary trapping sets with no mention. Since trapping sets consist of some cycles, any two nodes (edges) sharing the same edge cannot have the same color in the simplified representation of an elementary trapping set if nodes (edges) having the same image in the protograph are regarded as having the same color.

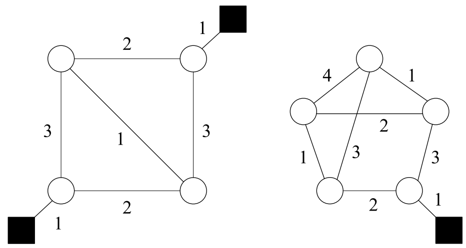
Definition 3: A *vertex coloring* of a graph is a labeling of the graph's vertices with colors such that no two vertices sharing the same edge have the same color. The smallest number of colors needed to color a graph G is called the *chromatic number* denoted by $\chi(G)$. A graph G whose vertices can be assigned k colors is *k-colorable* and it is *k-chromatic* if $\chi(G) = k$. An *edge coloring* of a graph is a proper coloring of the edges, meaning an assignment of colors to edges so that no vertex is incident to two edges of the same color. The smallest number of colors needed for an edge coloring of a graph G is the *edge chromatic number* denoted by $\chi'(G)$. A graph G whose edges can be assigned k colors is *k-edge-colorable* and it is *k-edge-chromatic* if $\chi'(G) = k$.

Clearly, elementary trapping sets in protograph-based LDPC codes should be L'-colorable and J'-edge-colorable with $L' \leq L$ and $J' \leq J$. From Theorems 2 and 3, we can find out which elementary trapping sets can appear in protograph-based LDPC codes with proper permutations and lift size.

Theorem 2 ([15]): Let $\Delta(G)$ denote the maximum degree of vertices in a graph G. For a simple graph G, $\chi'(G)$ is either $\Delta(G)$ or $\Delta(G) + 1$.

Proof: It is obvious that $\chi'(G) \geq \Delta(G)$ for edge coloring. From Vizing's theorem, we have $\chi'(G) \leq \Delta(G) + \mu(G)$ for a multigraph G, where $\mu(G)$ is the multiplicity of G. For a simple graph G, we obtain $\chi'(G) \leq \Delta(G) + 1$. ■

Theorem 3: Consider a fully connected simple protograph with J horizontal nodes and L vertical nodes where $J < L$. If an elementary trapping set $\mathcal{T}$ is edge-colored with $\chi'(\mathcal{T}) = J$



(a) A $(4, 2)$ trapping set. (b) A $(5, 1)$ trapping set.

Fig. 2. The simplified representation of two elementary trapping sets for $d_v = 3$

colors, there exist proper permutations and lift size such that the elementary trapping set $\mathcal{T}$ can occur in protograph-based LDPC codes lifted from the protograph.

Proof: According to Theorem 2, $\chi'(\mathcal{T})$ is either J or $J+1$ because the maximum degree $\Delta(\mathcal{T})$ is J. Since we actually have J colors, $\chi'(\mathcal{T})$ should be J so that the elementary trapping set $\mathcal{T}$ can occur in any protograph-based LDPC codes. Note that any vertex coloring of $\mathcal{T}$ does not need to be considered because $\chi(G) \leq \Delta(G) + 1$ for any simple graph G [15] and $\Delta(\mathcal{T}) + 1 \leq L$. ■

Definition 4: A graph G is said to be *class 1* when $\chi'(G) = \Delta(G)$. Otherwise, G is said to be *class 2*.

Finally, what we need to find out which elementary sets cannot appear in protograph-based LDPC codes from a fully connected simple protograph is determining whether each elementary trapping set is class 1 or 2. It is known that determining the class of a graph is an NP complete problem [16]. However, since small size of trapping sets are usually of interest, it is not difficult to determine the class of an elementary trapping set. As an example, the simplified representations of a $(4, 2)$ and a $(5, 1)$ elementary trapping sets for $d_v = 3$ are shown in Fig. 2. The $(4, 2)$ trapping set can be edge-colored with 3 colors which are illustrated as numbers instead as shown in Fig. 2(a) so that it is class 1. It means that this $(4, 2)$ trapping set can occur in some protograph-based LDPC codes lifted from a $3 \times L$ fully connected simple protograph with proper permutations and lift size. On the other hand, the $(5, 1)$ trapping set is class 1 as shown in Fig. 2(b) so that it cannot appear in any protograph-based LDPC codes lifted from the $3 \times L$ protograph.

IV. A New Measure for the Harmfulness of Trapping Sets

All possible elementary trapping sets are identified via a systematic search by a graph-theoretic tool in Section II. If those elementary trapping sets appear in a specific LDPC code, each trapping set may have a different amount of effects on the error correcting performance degradation in the error floor region, even under the same a and b. In this section, a new measure for estimating the harmfulness of elementary trapping sets is proposed.

The dynamic behavior of trapping sets in the BP decoding over the AWGN channel can be approximately explained by linear system models [6]-[8]. A linear system model of trapping sets will be briefly introduced by following the notations of [8]. Consider an elementary trapping set $\mathcal{T}(a, b)$ where the number of internal edges is $m := ad_v - b$. Let $\mathbf{x}_l$ denote a message column vector of length m whose i-th element means the message passed through the i-th edge in $\mathcal{T}$ from the variable node to the degree-two check node at the l-th iteration. Let λ denote the channel input column vector of length a and $\lambda_l^{(ex)}$ denote the extrinsic message column vector of length b flowing into $\mathcal{T}$ through b degree-one check nodes at the l-th iteration. Note that $\lambda_l^{(ex)}$ can be computed by using the density evolution with Gaussian approximation [17]. At last, let $\tilde{\mathbf{t}}_l$ denote the output column vector of length a after l iterations. Then, the following equations can be obtained via the linear system model.

$$\mathbf{x}_0 = \mathbf{B}\lambda \tag{1}$$

$$\mathbf{x}_l = \bar{g}'_l \mathbf{A}\mathbf{x}_{l-1} + \mathbf{B}\lambda + \mathbf{B}_{ex}\lambda_l^{(ex)} \tag{2}$$

$$\tilde{\mathbf{t}}_l = \bar{g}'_l \mathbf{C}\mathbf{x}_{l-1} + \lambda + \mathbf{D}_{ex}\lambda_l^{(ex)} \tag{3}$$

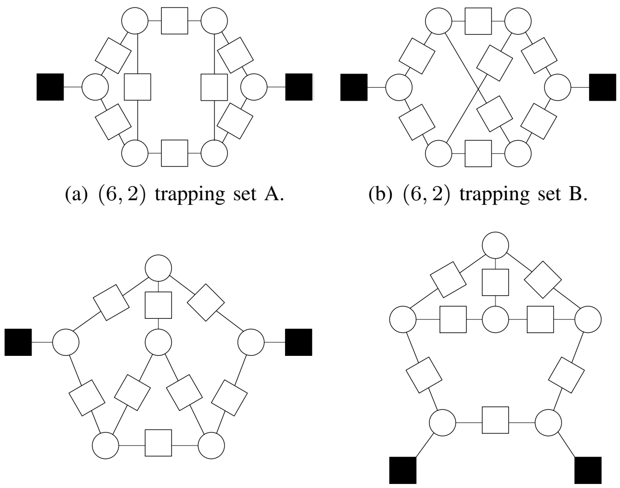
In (2) and (3), $\bar{g}'_l$ represents the mean gain generated by check node operations at the l-th iteration as seen in the modified min-sum decoding [18]. Since $\bar{g}'_l$ does not have a large effect on estimating the harmfulness of trapping sets, we will continue this argument without $\bar{g}'_l$.

The matrices in (1), (2), and (3) are determined by the structure of $\mathcal{T}$. The $m \times a$ matrix $\mathbf{B}$ and the $m \times b$ matrix $\mathbf{B}_{ex}$ are used to map λ and $\lambda_l^{(ex)}$ to the proper entries of the message vector to match the set of variable node update equations, respectively. The $m \times m$ matrix $\mathbf{A}$ describes the relation of the message update upon the previous message vector. Each nonzero entry $[\mathbf{A}]_{ij} = 1$ means the j-th edge of the previous iteration is involved with the variable node update of the i-th edge. The $a \times m$ matrix $\mathbf{C}$ and the $a \times b$ matrix $\mathbf{D}_{ex}$ are used to map $\mathbf{x}_{l-1}$ and $\lambda_l^{(ex)}$ to the corresponding entry of the soft output decision vector $\tilde{\mathbf{t}}_l$.

Let $\mu_k \in \mathbb{C}$ be an eigenvalue of the nonnegative matrix $\mathbf{A}$ and let $\mathbf{w}_k$ be the left eigenvector associated with μ_k such that $\mathbf{w}_k^* \mathbf{A} = \mu_k \mathbf{w}_k^*$. Let r denote the eigenvalue which has the largest magnitude and let $\mathbf{w}_1$ denote the left eigenvector associated with r. Note that r is real and $\mathbf{w}_1$ is positive under our assumptions [8]. Then, the trapping set error indicator β_l is defined as belows, which means that a trapping set is in an erroneous direction if β_l is negative and it is in a correct direction, otherwise.

$$\beta_l = \mathbf{w}_1^T \mathbf{B}\lambda + \sum_{i=1}^{l} \frac{\mathbf{w}_1^T \left(\mathbf{B}\lambda + \mathbf{B}_{ex}\lambda_i^{(ex)} \right)}{r^i} \tag{4}$$

In [6], r is called *gain* of trapping sets and it is asymptotically equal to $d_v - 1 - b/a$. The gain of a trapping set can represent the harmfulness of the trapping set, but it is not sufficient to be an exact measure because all non-isomorphic (a, b) trapping sets have the same gain. Therefore, we propose



(a) $(6,2)$ trapping set A. (b) $(6,2)$ trapping set B.

(c) $(6,2)$ trapping set C. (d) $(6,2)$ trapping set D.

Fig. 3. Four $(6,2)$ elementary trapping sets for $d_v = 3$

a measure for estimating the harmfulness of trapping sets as Definition 5.

Definition 5: The *trappability* of a trapping set $\mathcal{T}$, denoted by $t(\mathcal{T})$, is a measure of the occurrence of trapping errors. It is quantitatively defined as

$$t(\mathcal{T}) = r\frac{\sum_{i=1}^{m} |w_{1i}|}{\sum_{i\in\mathcal{I}_{ex}} |w_{1i}|} \tag{5}$$

where w_{1i} is the i-th element of $\mathbf{w}_1$ and $\mathcal{I}_{ex}$ is the index set whose elements correspond to the edges directly affected by the extrinsic information.

The definition of the trappability in (5) is explained as follows. The gain r can represent the erroneous behavior of a trapping set caused by message-passing inside the trapping set. However, the trappability of a trapping set clearly depends on the inter-connection with extrinsic information as well as the intra-connection of the trapping set. The equation (4) is rewritten as

$$\beta_l = \mathbf{w}_1^T \mathbf{B}\lambda \left(1 + \sum_{i=1}^{l} \frac{1}{r^i}\right) + \sum_{i=1}^{l} \frac{\mathbf{w}_1^T \mathbf{B}_{ex} \lambda_i^{(ex)}}{r^i}. \tag{6}$$

While the first term in (6) is only dominated by the channel input and the intra-connection, the second term in (6) reflects the extrinsic information. When a trapping set receives many erroneous inputs from the channel, the first term tends to be negative and the second term goes positively. Since the trappability of a trapping set is desired to be independent of the channel input, the outside structure of the trapping set, and the number of iterations, we make the trappability involve the sensitivity with respect to the change of λ and $\lambda_i^{(ex)}$ when $l \to \infty$. For simplicity, assume that $\lambda_i^{(ex)}$ is constant with respect to l. Then, the sums of the derivatives with respect to the elements of λ and $\lambda_i^{(ex)}$ becomes $\left(r\sum_{i=1}^{m} |w_{1i}|\right)/(r-1)$ and $\left(\sum_{i\in\mathcal{I}_{ex}} |w_{1i}|\right)/(r-1)$, respectively. By defining the ratio of those sums as the trappability, the equation (5) is obtained.

Note that the trappability provides fair comparision of trapping sets under the same SNR and degree distribution.

As an example, consider all non-isomorphic $(6,2)$ elementary trapping sets for $d_v = 3$ as shown in Fig. 3. Let $\mathcal{T}_A$, $\mathcal{T}_B$, $\mathcal{T}_C$, and $\mathcal{T}_D$ denote the trapping sets in Fig. 3(a), Fig. 3(b), Fig. 3(c), and Fig. 3(d), respectively. For $\mathcal{T}_A$, the 16×16 matrix $\mathbf{A}$ is represented as

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

and $\mathcal{I}_{ex} = \{4, 5, 10, 11\}$. Then, we obtain $r = 1.6956$,

$$\mathbf{w}_1 = [0.85\ 0.85\ 0.59\ 1\ 1\ 0.59\ 0.85\ 0.85 \\ 0.59\ 1\ 1\ 0.59\ 0.85\ 0.85\ 0.85\ 0.85]^T,$$

and $t(\mathcal{T}_A) = 5.5707$. In the same way, for $\mathcal{T}_B$, we obtain $r = 1.6956$ and $t(\mathcal{T}_B) = 5.5707$, for $\mathcal{T}_C$, $r = 1.7$ and $t(\mathcal{T}_B) = 5.5666$, and for $\mathcal{T}_D$, $r = 1.7293$ and $t(\mathcal{T}_B) = 6.3886$. We can see that $\mathcal{T}_D$ is the most harmful under the same SNR, and $\mathcal{T}_A$ and $\mathcal{T}_B$ have the same trappability.

V. Conclusions

In this paper, a systematic approach to identify all non-isomorphic elementary trapping sets is proposed by using a graph-theoretic tool "nauty." It is shown that among all elementary trapping sets, the trapping sets which cannot appear in protograph-based LDPC codes lifted from a fully connected simple protograph regardless of lifting can be identified by using edge coloring. Finally, the trappability of a trapping set is proposed as a measure for evaluating the harmfulness of trapping sets by introducing the linear system model of trapping sets.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No.2012-0000186) and also supported by the KCC (Korea Communications Commission) under the R&D program supervised by the KCA (Korea Communications Agency) (KCA-2012-08-911-04-003).

REFERENCES

[1] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[2] T. J. Richardson, "Error floors of LDPC codes," in *Proc. Allerton Conf. Commun., Control, Comput.*, UIUC, IL, Oct. 2003, pp. 1426-1435.

[3] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570-1579, Jun. 2002.

[4] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "Analysis of absorbing sets for array-based LDPC codes," in *Proc. IEEE Int. Conf. Commun.*, Glasgow, Jun. 2007, pp. 6261-6268.

[5] B. Vasic, S. K. Chilappagari, D. V. Nguyen, and S. K. Planjery, "Trapping set ontology," in *Proc. Allerton Conf. Commun., Control, Comput.*, UIUC, IL, Sep. 2009, pp. 1-7.

[6] C. Schlegel and S. Zhang, "On the dynamics of the error floor behavior in (regular) LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3248-3264, Jul. 2010.

[7] J. Sun, "Studies on graph-based coding systems," Ph.D. dissertation, Ohio State Univ., Columbus, OH, 2004.

[8] B. K. Butler and P. H. Siegel, "Error floor approximation for LDPC codes in the AWGN channel," in *Proc. Allerton Conf. Commun., Control, Comput.*, UIUC, IL, Sep. 2011, pp. 204-211.

[9] S. Landner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Proc. Int. Conf. Wireless Netw. Commun. Mobile Comput.*, Maui, Jun. 2005, pp. 630-635.

[10] Y. Han and W. E. Ryan, "Low-floor decoders for LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1163-1173, Jun. 2009.

[11] B. D. McKay. (2009) nauty users guide (version 2.4). [Online]. Available: http://cs.anu.edu.au/~bdm/nauty/

[12] H. Park, "LDPC decoding schemes with post-processing for lowering error floors," M.S. thesis, Seoul National Univ., Seoul, Korea, Feb. 2009.

[13] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protograph," *IPN Progress Report 42-154, JPL*, Aug. 2003.

[14] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, "Lowering the error floor of LDPC codes using cyclic liftings," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2213-2224, Apr. 2011.

[15] T. R. Jensen and B. Toft, *Graph Coloring Problems*. New York, NY: Wiley-Interscience, 1995.

[16] I. Holyer, "The NP-completeness of edge-coloring," *SIAM J. Comput.*, vol. 10, no. 4, pp. 718-720, 1981.

[17] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657-670, Feb. 2001.

[18] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406-414, Mar. 2002.