

# Absorbing Sets of LDPC Codes Under a Gallager B Decoding Variant

Renaya Dhingra, Gretchen L. Matthews\*, Collin McDevitt, and Emily McMillon†

Department of Mathematics, Virginia Tech, USA

## Abstract

Absorbing sets are graphical structures that cause iterative decoders to fail. The sets depend on the particular iterative decoding algorithm. In this paper, we consider a variant of the well-studied Gallager B decoding algorithm for binary low-density parity-check codes. Here, a bit node updates its value according to either the number of or fraction of unsatisfied neighboring check nodes. The traditional Gallager B algorithm is the special case in which half (meaning a fraction of  $\frac{1}{2}$ ) of the check nodes being unsatisfied is required for a bit to change its value. We study absorbing sets in these settings along with the received words which give decoder failure. We determine how changing the update rules pick up new absorbing sets. This allows for fine-tuning the standard algorithm according to the particular channel or error probability. We also connect these notions to Boolean functions.

**Keywords:** Coding theory, iterative decoding, LDPC codes, Tanner graphs, absorbing sets.

**2020 Mathematics Subject Classification:** 94B05, 94B35.

## 1 Introduction

Due to their low-complexity iterative decoding algorithms and excellent performance at large block lengths, codes based on low-density parity-check (LDPC) matrices are at the forefront of coding theory research. LDPC codes were introduced by Gallager in 1962 [6, 7] but went largely unnoticed due to the lack of computing power available at the time. In the mid-1990s, they were rediscovered by Mackay and Neal [13, 15] and later demonstrated to have sequences with information rates approaching the Shannon limit [14, 16]. This work laid the foundation for numerous results and standards including those for 5G technology [10, 17].

LDPC codes are codes with sparse parity-check matrix representations that are often represented using sparse bipartite graphs called Tanner graphs [20]. The sparse matrix representations of these codes yields easier computation, but this potential was not realized until decades after their introduction when improvements in computational power made LDPC codes practical for longer code lengths. Tanner graphs [20] are used to visualize iterative decoders which are often

---

\*Matthews is supported in part by NSF DMS-2201075 and the Commonwealth Cyber Initiative.

†McMillon is supported by NSF DMS-2303380.

described as message passing algorithms where the vertices perform computations and send information along the edges of the Tanner graph. Unfortunately, while these decoders have excellent performance in general, they are suboptimal with respect to decoder accuracy, and certain substructures in a code's Tanner graph can cause the algorithms to fail to converge to a codeword. Substructures in a code's Tanner graph representation have been shown to characterize decoder failure over certain channels and decoders. These substructures include stopping sets, trapping sets, absorbing sets, near-codewords, and pseudocodewords (e.g. [1, 3, 4, 11]).

In this paper, we focus on absorbing sets, which are combinatorially defined graph substructures, first identified in [3], that cause iterative decoder failure even at relatively high signal to noise ratios over a variety of channel models. The classical definition of absorbing sets is based on iterative decoders with update rules based on majority rule. In the same paper where Gallager introduced LDPC codes [6], he also introduced two relatively easy to understand iterative decoding algorithms, today referred to as Gallager A and Gallager B. In rough terms, Gallager A uses a check node update rule where all adjacent nodes must be in agreement in order for a variable node to change, while Gallager B loosens this update rule to require only some adjacent nodes to be in agreement. Hence, Gallager A can be viewed as a special case of Gallager B.

Here, we explore absorbing sets in LDPC codes under a variant of the Gallager B iterative decoding algorithm. In particular, we define absorbing sets in these variations of the Gallager B algorithm. We then provide results on how absorbing sets relate to each other under different update rules. Gallager's decoding algorithm was proposed for codes with bi-regular Tanner graphs, so we also characterize absorbing sets in a variation on Gallager B that requires a fraction of adjacent nodes to be in agreement instead of a specific number. These results may also be considered to fill in the spectrum of rules between Gallager A and B (and beyond). They offer the option to tailor decoding rules to particular levels of sensitivity.

This study is motivated by practical implications and strategies for error correction. First, superior error mitigation can be achieved by improving the decoding algorithm to better handle messages within absorbing sets. It can help reduce error rates but may involve using more sophisticated update rules that consider the broader graph structure. Next, in terms of code design, minimizing the formation of absorbing sets can enhance decoding performance. Optimizing the Tanner graph structure, such as by adjusting degree distributions, can reduce the prevalence and impact of these sets. Finally, absorbing sets are a significant obstacle in iterative decoding of LDPC codes, especially under integer update conditions. These sets can trap the decoding process in an incorrect state, leading to failures. Addressing this issue involves understanding the structure and behavior of absorbing sets and developing strategies to escape them. By doing so, it is possible to create more robust LDPC codes and decoding algorithms that ensure reliable communication.

The paper is structured as follows. In Section 2, we give the definitions and background on codes, iterative decoding, and absorbing sets needed in the remainder of the paper. Section 3 contains the theoretical results of the paper. It is divided into three subsections: the first describes the standard setting while the latter two consider modified update conditions. Section 4 and Section 5 concludes the paper.

## 2 Preliminaries

In this section, we will review the foundation on which our results will be built and set the notation to be used later. We will use the standard notation and definitions from coding theory. For more background on coding theory, one excellent source is [9].

### 2.1 Code representations

We begin with some definitions. Throughout, the finite field of cardinality  $q$  where  $q$  is a power of a prime number is denoted by  $\mathbb{F}_q$ . We use  $\mathbb{F}_q^{m \times n}$  to denote the set of  $m \times n$  matrices with entries in  $\mathbb{F}_q$  and  $\mathbb{F}_q^n := \mathbb{F}_q^{1 \times n}$ . For a matrix  $A \in \mathbb{F}_q^{m \times n}$ , we may write  $A = (A_{ij})$  to mean that  $A_{ij} \in \mathbb{F}_q$  is the entry in Row  $i$  and Column  $j$  of  $A$ .

A linear code  $C$  over  $\mathbb{F}_q$  is a subspace of the vector space  $\mathbb{F}_q^n$  for a positive integer  $n$ . An element of the code  $C$  is called a *codeword*. A codeword  $\mathbf{w} \in C$  is written  $\mathbf{w} = (w_1, \dots, w_n)$ . Important code parameters include the length  $n$ , the dimension  $k := \dim_{\mathbb{F}_q} C$ , and the minimum distance  $d := \min \{wt(c) : c \in C \setminus \{0\}\}$  where  $wt(x) := |\{i \in n : x_i \neq 0\}|$  is the Hamming weight of a vector  $x \in \mathbb{F}_q^n$ . A code over  $\mathbb{F}_q$  with length  $n$ , dimension  $k$ , and minimum distance  $d$  is called an  $[n, k, d]_q$  code. The rate of an  $[n, k, d]_q$  code is  $\frac{k}{n}$ , which gives a measure of its efficiency, and its error correcting capability is measured by  $d$  since it can correct any  $\lfloor \frac{d-1}{2} \rfloor$  errors. We often write  $[n, k]$  code if the alphabet is clear from the context and the minimum distance is not being discussed.

Linear codes can be represented by matrices. A generator matrix for an  $[n, k, d]_q$  code  $C$  is any matrix  $G \in \mathbb{F}_q^{k \times n}$  with row span equal to  $C$ . For the purposes of this paper, the most relevant matrix is a parity-check matrix, which is defined as follows.

**Definition 2.1.** A parity check matrix for an  $[n, k]$  code  $C$  over  $\mathbb{F}_q$  is a matrix  $H \in \mathbb{F}_q^{r \times n}$  whose null space is  $C$ , meaning

$$C = \{x \in \mathbb{F}_q^n : Hx^T = 0\}.$$

Notice that  $r \geq n - k$  and the rows of  $H = (H_{ij})$  correspond to parity-check equations of  $C$ . In particular, Row  $i$  of  $H$  gives rise to the parity-check equation

$$H_{i1}x_1 + H_{i2}x_2 + \dots + H_{in}x_n = 0$$

so that the codewords of  $C$  are precisely the solutions to the system of equations

$$\begin{aligned} H_{11}x_1 + H_{12}x_2 + \dots + H_{1n}x_n &= 0 \\ H_{21}x_1 + H_{22}x_2 + \dots + H_{2n}x_n &= 0 \\ \vdots & \\ H_{r1}x_1 + H_{r2}x_2 + \dots + H_{rn}x_n &= 0 \end{aligned} \tag{1}$$

over  $\mathbb{F}_q$ . We typically take  $r = n - k$ , which is the minimum number of rows necessary in a parity-check matrix for an  $[n, k]$  code. Error detection can be accomplished by checking whether or not a received word,  $w$ , satisfies these parity-check equations given in (1)

**Remark 2.2.** According to Definition 2.1, a linear code  $C$  does not have a unique parity check matrix. If we wish to fix a parity-check matrix  $H$  for  $C$ , we may write  $C = C(H)$  to emphasize this choice.

In this paper, we restrict our attention to binary linear codes, meaning codes over  $\mathbb{F}_2 = \{0, 1\}$ . Here, codewords are strings of binary digits (0s and 1s), and operations are performed modulo 2. Next, we discuss graphical representations of these codes. A Tanner graph provides a visual representation of the relationships between variables and constraints in a code, facilitating the implementation of iterative decoding algorithms, such as belief propagation. Tanner graphs are crucial in understanding and implementing iterative decoding algorithms. They allow for visualization of how information flows between variables and constraints, enabling efficient algorithms like belief propagation to correct errors in transmitted data. The graphs considered are simple bipartite graphs given by two disjoint sets of vertices  $V_1$  and  $V_2$  and an edge set  $E$ ; such a graph  $G$  is denoted  $G = (V_1, V_2; E)$ . The neighborhood of a vertex  $v$  is denoted  $\mathcal{N}(v)$ . Given a subset of vertices  $S \subseteq V_1 \cup V_2$ , the induced subgraph of  $G$  is  $G_S = (S, \mathcal{N}(S); E_S)$  where  $E_S := \{\{u, v\} : u \in S, v \in \mathcal{N}(S) \text{ or } v \in S, u \in \mathcal{N}(S)\}$ . Recall that a biadjacency matrix of such a bipartite graph is a matrix  $M = (M_{ij}) \in \mathbb{F}_2^{|V_1| \times |V_2|}$  where  $M_{ij} = 1$  if and only if vertex  $i \in V_1$  is adjacent to vertex  $j \in V_2$ .

**Definition 2.3.** A Tanner graph of a binary  $[n, k]$  code  $C$  is a bipartite graph whose biadjacency matrix is a parity-check matrix of  $C$ .

In light of Remark 2.2, we typically consider the Tanner graph of a code  $C = C(H)$  defined as follows.

**Definition 2.4.** The Tanner graph  $T(H)$  for a binary linear code  $C = C(H)$  with  $H \in \mathbb{F}_2^{r \times n}$  is the simple bipartite graph  $(V, W; E)$  given by

- two sets of nodes (or vertices):
  - the set  $V$  of variable nodes whose elements correspond to the columns of  $H$ , equivalently, to the coordinates of codewords of  $C$
  - the set  $W$  of check nodes whose elements correspond to the rows of  $H$ , equivalently, to the parity-check constraints given by  $H$
- the set  $E$  of edges whose elements indicate which variables participate in which parity-check constraints; in particular, for  $i \in V$  and  $j \in W$ ,  $\{i, j\} \in E$  if and only if  $H_{ij} = 1$ .

We next give an example of a Tanner graph.

**Example 2.5.** Consider the binary code  $C$  of length 6 defined by the parity-check matrix

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The Tanner graph  $T(H) = (V, W; E)$  is constructed as follows and is given in Figure 1:

1. The set of variable nodes is  $V := \{v_1, v_2, v_3, v_4, v_5, v_6\}$ , where  $v_j$  corresponds to Column  $j$  of  $H$
2. The set of check nodes is  $W := \{w_1, w_2, w_3\}$ , where  $w_i$  corresponds to Row  $i$  of  $H$
3. The set  $E$  of edges is precisely those

- between  $w_1$  and each of  $v_1, v_3, v_4$ , since the first row of  $H$  has 1s in the 1st, 3rd, and 4th columns.
- between  $w_2$  and each of  $v_2, v_3, v_5$ , since the second row of  $H$  has 1s in the 2nd, 3rd, and 5th columns.
- between  $w_3$  and each of  $v_1, v_2, v_6$ , since the third row of  $H$  has 1s in the 1st, 2nd, and 6th columns.

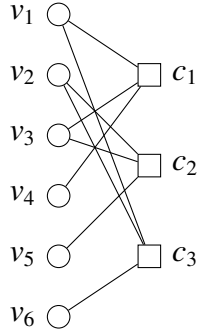


Figure 1: The Tanner graph corresponding to the parity-check matrix  $H$  given in Example 2.5.

Example 2.5 gives a basic example of a Tanner graph. In practice, these graphs are larger and more complex, especially for codes used in modern communication systems such as LDPC codes.

An important type of LDPC codes are *biregular LDPC codes* which are those in which every variable node is connected to  $d_v$  check nodes, and each check node is connected to  $d_c$  variable nodes. Such a biregular LDPC code is sometimes called a  $(d_v, d_c)$ -regular LDPC code. This regularity simplifies the design and analysis processes, making it easier to predict the performance and optimize the codes for specific applications. Biregular LDPC codes can preserve the core benefits of LDPC codes, such as their sparse parity-check matrices and the efficiency of iterative decoding algorithms, while introducing a structured consistency that can enhance performance in certain scenarios [18]. An example of a biregular LDPC code is given in Example 2.6.

**Example 2.6.** The parity check matrix below gives a  $(3, 4)$ -regular LDPC code.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Its corresponding Tanner graph is in Figure 2.

## 2.2 Relevant decoding algorithms

The steps in Gallager A/B decoding involve specific conditions for flipping the value of a variable node (VN) in response to calculations performed by its check node (CN) neighbors. Here, flip

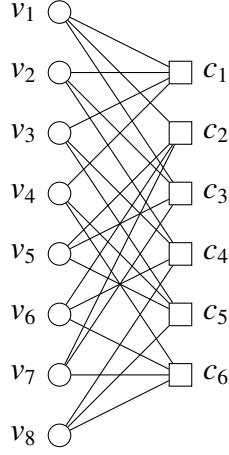


Figure 2: The (3,4)-biregular Tanner graph corresponding to the parity-check matrix  $H$  given in Example 2.6.

means a value of 0 is changed to 1 and a value of 1 is changed to 0. In Gallager A decoding, a variable node flips its value only if all its neighboring check nodes are unsatisfied. In contrast, Gallager B decoding dictates a specific number of neighboring check nodes that need to be unsatisfied in order for a variable node to flip.

The notation that we will use to explain Gallager A/B decoding may be found in Table 1.

$\mu^{(0)}(v)$	the channel information received for $v$
$\mu^{(t)}(v)$	the state of variable node $v$ following the completion of iteration $t$
$\mu_{c \rightarrow v}^{(t)}$	the information sent from check node $c$ to variable node $v$ at time iteration $t$

Table 1: Notation used in Gallager A/B decoding where  $v$  represents a variable node and  $c$  represents a check node.

With the notation set, we are prepared to explain Gallager A/B decoding for a code  $C$  with variable nodes  $V = \{v_1, \dots, v_n\}$  and check nodes  $W = \{c_1, \dots, c_m\}$ . To begin, for each  $v \in V$ , fix a value

$$b_v \leq \min \{|\mathcal{N}(v)| - 1 : v \in V\}.$$

### Gallager A/B Decoding Algorithm

1. Initialize each variable node  $v$  with values  $\mu^{(0)}(v)$  given by the channel. One may think of  $(\mu^{(0)}(v_1), \dots, \mu^{(0)}(v_n))$  as the received word. Begin with iteration  $t = 0$ .
2. At each check node  $c$ , for each of its variable node neighbors  $v \in \mathcal{N}(c)$ , compute

$$\mu_{c \rightarrow v}^{(t)} := \sum_{v' \in \mathcal{N}(c) \setminus \{v\}} \mu^{(t)}(v') \pmod{2}.$$

3. At each variable node  $v$ , set

$$\mu^{(t+1)}(v) := \begin{cases} \mu_{c' \rightarrow v}^{(t)} & \text{if at least } b_v \text{ of the } c' \in \mathcal{N}(v) \text{ are equal to } \mu_{c' \rightarrow v}^{(t)} \\ \mu^{(t)}(v) & \text{otherwise.} \end{cases}$$

Increment  $t$  by 1.

4. Repeat Steps 2 and 3 until either  $(\mu^{(t)}(v_1), \dots, \mu^{(t)}(v_n)) \in C$  or after a certain number of iterations have passed.

The difference between Gallager A and B is in the choice of  $b_v$ . In Gallager A,

$$b_v = |\mathcal{N}(v)| - 1,$$

meaning that in Gallager B,  $b_v$  is a fixed smaller number. Typically, if  $\deg(v) = \deg(v')$  for all vertices  $v, v' \in V$ , then values are chosen so that  $b_v = b_{v'}$  and we simply write  $b$  rather than  $b_v$ . In particular, the class of LDPC codes introduced by Gallager were all biregular. Hence, a fixed choice of  $b$  made sense, because every variable node had the same number of neighbors. In the case of LDPC codes which are not biregular, a slight variation on Gallager B decoding can be used, in which the following step replaces Step 3 in the traditional decoding algorithm.

- 3b. Fix  $b' \in \mathbb{Z}$ . At each variable node  $v$ , let  $S'_v = \{c \in \mathcal{N}(v) \mid \mu_{c \rightarrow v}^{(t)} = \mu^{(t)}(v)\}$  be the set of satisfied check node neighbors of  $v$  at iteration  $t$  and  $U'_v = \mathcal{N}(v) \setminus S'_v$  be the set of unsatisfied check node neighbors of  $v$  at iteration  $t$ . Then set

$$\mu^{(t+1)}(v) = \begin{cases} \mu^{(t)}(v) + 1 & \text{if } |U'_v| \geq |S'_v| + b' \\ \mu^{(t)}(v) & \text{otherwise.} \end{cases}$$

Note that this phrasing uses satisfied and unsatisfied check node neighbors instead of messages  $\mu_{c \rightarrow v}^{(t)}$ . To see that this formulation is equivalent, notice that:

- If  $\mu^{(t)}(v) = 0$ , then in the set  $S'_v$ , we are looking at the set of check nodes  $c \in \mathcal{N}(v)$  such that  $\mu_{c \rightarrow v}^{(t)} = 0$ . Because  $\mu_{c \rightarrow v}^{(t)} = \sum_{v' \in \mathcal{N}(c) \setminus v} \mu_{v' \rightarrow c}^{(t)} \pmod 2$ ,  $\mu^{(t)}(c) = \sum_{v \in \mathcal{N}(c)} \mu^{(t)}(v) = \mu_{c \rightarrow v}^{(t)} + \mu^{(t)}(v) = 0 + 0 = 0$ , so  $S'_v$  is exactly the set of satisfied check node neighbors.
- If  $\mu^{(t)}(v) = 1$ , then in the set  $S'_v$ , we are looking at the set of check nodes  $c \in \mathcal{N}(v)$  such that  $\mu_{c \rightarrow v}^{(t)} = 1$ . Because  $\mu_{c \rightarrow v}^{(t)} = \sum_{v' \in \mathcal{N}(c) \setminus v} \mu_{v' \rightarrow c}^{(t)} \pmod 2$ ,  $\mu^{(t)}(c) = \sum_{v \in \mathcal{N}(c)} \mu^{(t)}(v) = \mu_{c \rightarrow v}^{(t)} + \mu^{(t)}(v) = 1 + 1 = 0$ , so  $S'_v$  is exactly the set of satisfied check node neighbors.

Further note that Step 3b is equivalent to the original Step 3 in the case where  $C$  is biregular by taking  $b' = 2b - |\mathcal{N}(v)|$ .

Throughout the remainder of the paper, we will use the following version of the Gallager B algorithm, which we will call the  $(u \mid s + b)$  variant.

### $(u \mid s + b)$ Variant of Gallager B Decoding Algorithm

1. Initialize each variable node  $v$  with values  $\mu^{(0)}(v)$  given by the channel. One may think of  $(\mu^{(0)}(v_1), \dots, \mu^{(0)}(v_n))$  as the received word. Begin with iteration  $t = 0$ .
2. At each check node  $c \in W$ , calculate  $\mu^{(t)}(c) = \sum_{v \in \mathcal{N}(c)} \mu^{(t)}(v) \pmod 2$ . Define  $S = \{c \in W \mid \sum_{v \in \mathcal{N}(c)} \mu^{(t)}(v) = 0\}$  and  $U = W \setminus S$ .

3. At each  $v \in V$ , let  $S_v = S \cap \mathcal{N}(v)$  and  $U_v = U \cap \mathcal{N}(v)$ . Then set

$$\mu^{(t+1)}(v) = \begin{cases} \mu^{(t)}(v) + 1 & \text{if } |U_v| \geq |S_v| + b \\ \mu^{(t)}(v) & \text{otherwise.} \end{cases}$$

Increment  $t$  by 1.

4. Repeat Steps 2 and 3 until either  $(\mu^{(t)}(v_1), \dots, \mu^{(t)}(v_n)) \in C$  or after a certain number of iterations have passed.

**Remark 2.7.** We will say that the  $(u \mid s + b)$  variant uses the  $u \geq s + b$  update rule. The standard definition of an absorbing set occurs under the case when the modified Gallager B algorithm given in Section 2 has  $b = 0$ , i.e., under the  $(u \mid s)$  decoder. We consider the effect of modifying  $b$  in Sections 3.2 and 3.3.

We now give an example illustrating this variant of Gallager B decoding.

**Example 2.8.** Consider the code  $C(H)$  in Example 2.5 and corresponding Tanner graph in Figure 1. Suppose the vector we receive from the channel is  $\mathbf{y} = (0, 0, 0, 0, 0, 1)$ . Then we initialize:

$$\mu^{(0)}(v_1) = 0, \quad \mu^{(0)}(v_2) = 0, \quad \mu^{(0)}(v_3) = 0, \quad \mu^{(0)}(v_4) = 0, \quad \mu^{(0)}(v_5) = 0, \quad \mu^{(0)}(v_6) = 1$$

We'll first consider the case where we set  $b = 0$ . We first calculate:

$$\mu^{(0)}(c_1) = \mu^{(0)}(v_1) + \mu^{(0)}(v_3) + \mu^{(0)}(v_4) = 0$$

$$\mu^{(0)}(c_2) = \mu^{(0)}(v_2) + \mu^{(0)}(v_3) + \mu^{(0)}(v_5) = 0$$

$$\mu^{(0)}(c_3) = \mu^{(0)}(v_1) + \mu^{(0)}(v_2) + \mu^{(0)}(v_6) = 1$$

so  $S = \{c_1, c_2\}$  and  $U = \{c_3\}$ . We then calculate the sets  $S_v$  and  $U_v$  for each  $v \in V$ :

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$
$S_v$	$\{c_1\}$	$\{c_2\}$	$\{c_1, c_2\}$	$\{c_1\}$	$\{c_2\}$	$\emptyset$
$U_v$	$\{c_3\}$	$\{c_3\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{c_3\}$

Hence,  $v_1, v_2$ , and  $v_6$  are such that  $|U_v| \geq |S_v|$ . So the next iteration gives:

$$\mu^{(1)}(v_1) = 1, \quad \mu^{(1)}(v_2) = 1, \quad \mu^{(1)}(v_3) = 0, \quad \mu^{(1)}(v_4) = 0, \quad \mu^{(1)}(v_5) = 0, \quad \mu^{(1)}(v_6) = 0$$

We then calculate:

$$\mu^{(1)}(c_1) = 1, \quad \mu^{(1)}(c_2) = 1, \quad \mu^{(1)}(c_3) = 0$$

so  $S = \{c_3\}$  and  $U = \{c_1, c_2\}$ . We then calculate the sets  $S_v$  and  $U_v$  for each  $v \in V$ :

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$
$S_v$	$\{c_3\}$	$\{c_3\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{c_3\}$
$U_v$	$\{c_1\}$	$\{c_2\}$	$\{c_1, c_2\}$	$\{c_1\}$	$\{c_2\}$	$\emptyset$

So all  $v$  except  $v_6$  are such that  $|U_v| \geq |S_v|$ . The next iteration gives:

$$\mu^{(2)}(v_1) = 0, \quad \mu^{(2)}(v_2) = 0, \quad \mu^{(2)}(v_3) = 1, \quad \mu^{(2)}(v_4) = 1, \quad \mu^{(2)}(v_5) = 1, \quad \mu^{(2)}(v_6) = 0$$

We then calculate:

$$\mu^{(2)}(c_1) = 0, \quad \mu^{(2)}(c_2) = 0, \quad \mu^{(2)}(c_3) = 0$$

which shows that this is a codeword, and so we are done.

Now consider the case where  $b = 1$ , still with  $\mathbf{y} = (0, 0, 0, 0, 0, 1)$ . We first calculate:

$$\mu^{(0)}(c_1) = 0, \quad \mu^{(0)}(c_2) = 0, \quad \mu^{(0)}(c_3) = 1$$

so  $S = \{c_1, c_2\}$  and  $U = \{c_3\}$ . We then calculate the sets  $S_v$  and  $U_v$  for each  $v \in V$ :

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$
$S_v$	$\{c_1\}$	$\{c_2\}$	$\{c_1, c_2\}$	$\{c_1\}$	$\{c_2\}$	$\emptyset$
$U_v$	$\{c_3\}$	$\{c_3\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{c_3\}$

We see that only  $v_6$  is such that  $|U_v| \geq |S_v| + b$ . So the next iteration gives:

$$\mu^{(1)}(v_1) = 0, \quad \mu^{(1)}(v_2) = 0, \quad \mu^{(1)}(v_3) = 0, \quad \mu^{(1)}(v_4) = 0, \quad \mu^{(1)}(v_5) = 0, \quad \mu^{(1)}(v_6) = 0$$

which is a codeword, and so we are done.

In the next section, we will consider these decoding algorithms and structures in a Tanner graph that can interfere with correct decoding.

### 3 Absorbing sets under various update protocols

Absorbing sets have been studied as a way to understand iterative decoder failure; see for instance, [2, 3, 8, 12]. In this section, we consider absorbing sets relevant to variants of the Gallager B algorithm introduced in Section 2. We first review the notion of absorbing sets in the literature and then in the following subsections present new results.

#### 3.1 Absorbing sets in the standard setting

Consider a code  $C(H)$  and its Tanner graph  $G := T(H)$  with sets of variable and check nodes  $V$  and  $W$ . For a variable node  $v \in V$ , let its number of odd degree check node neighbors be denoted by

$$D_o(v) = |\{c \in \mathcal{N}(v) : |\mathcal{N}(c)| \equiv 1 \pmod{2}\}|$$

and its number of even degree check node neighbors be denoted by

$$D_e(v) = |\{c \in \mathcal{N}(v) : |\mathcal{N}(c)| \equiv 0 \pmod{2}\}|.$$

Note that the dependence of the Tanner graph  $T(H) = G$  on  $H$  is crucial—just as there are many different parity-check matrices to represent a single code, there are also many different Tanner graphs. While codewords do not change with these different representations, graph substructures such as absorbing sets are representation dependent.

For convenience, given a Tanner graph  $G = (V, W; E)$  and some  $D \subseteq V$ , the set of odd degree (resp., even) check node neighbors in the induced subgraph  $G_D$  of  $G$  is  $O(D)$  (resp.,  $E(D)$ ). With these ideas, we can now define the standard type of absorbing set found in the literature [2].

**Definition 3.1.** Given integers  $a \leq |V|$  and  $b \leq |W|$ , an  $(a, b)$ -absorbing set is a subset  $D \subseteq V$  of variable nodes with  $|D| = a$ ,  $|O(D)| = b$ , and for each  $v \in D$ ,

$$|\mathcal{N}(v) \cap O(D)| < |\mathcal{N}(v) \cap E(D)|.$$

Throughout, in a slight abuse of notation, we will use three representations of an absorbing set interchangeably:

- the set  $D \subseteq V$  of variable nodes corresponding to the absorbing set
- the graph  $G_D$  induced by the absorbing set,
- the vector  $\mathbf{y}_D \in \mathbb{F}_2^n$ , where

$$(\mathbf{y}_D)_i = \begin{cases} 1 & \text{if } v_i \in D \\ 0 & \text{otherwise.} \end{cases}$$

The vector  $\mathbf{y}_D$  can be thought of as the *support vector* or *incidence vector* of  $D$ .

We now give an example to illustrate the preceding concepts.

**Example 3.2.** Let  $H$  be as given in Example 2.6 and let  $G = T(H)$  be its associated Tanner graph shown in Figure 2. Consider the set  $D = \{v_5, v_6, v_8\}$ . Notice that  $\mathcal{N}(D) = \{c_2, c_3, c_4, c_5, c_6\}$ . The induced graph  $G_D$  contains all these variable nodes as well as all edges between them in the original graph  $G$  and is shown in Figure 3.

This visual representation can be helpful when determining odd versus even parities of degrees in  $G_D$ . Using the graph, we can see that

$$O(D) = \{c_3\} \quad \text{and} \quad E(D) = \{c_2, c_4, c_5, c_6\},$$

and we can check that for each  $v \in D$ ,  $|\mathcal{N}(v) \cap O(D)| < |\mathcal{N}(v) \cap E(D)|$ . Because  $|D| = 3$  and  $|O(D)| = 1$ , this is a  $(3, 1)$ -absorbing set. We may refer to this absorbing set as its set of variable nodes,  $D$ , as its induced subgraph  $G_D$ , or as its support vector  $\mathbf{y}_D = (0, 0, 0, 0, 1, 1, 0, 1)$ .

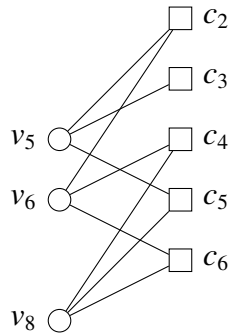


Figure 3: The induced graph  $G_D$  given by  $D = \{v_5, v_6, v_8\}$  where  $G$  is as in Figure 2. Here,  $G_D$  is a  $(3, 1)$ -absorbing set under the standard definition.

In Example 3.2, we considered the induced subgraph of an absorbing set. This is a subgraph of the original Tanner graph. It is sufficient to consider the induced subgraphs of absorbing sets in

order to consider absorbing sets, as we could imagine such an induced subgraph as isomorphic to subgraphs of many different Tanner graphs. The results in Subsections 3.2 and 3.3 and in Section 4 thus consider graphical representations without specific overall codes or Tanner graphs. It is important to make this note, as many absorbing sets have degree 1 check nodes (as was the case in Example 3.2), and no practical code would ever have a degree 1 check node in the overall code.

**Remark 3.3.** *It has been shown that the probability that a decoder fails is independent of the transmitted codeword [5]. Hence, we can always assume that the transmitted vector in our analyses was the all zero codeword. Therefore, for an absorbing set  $D$ , we can think of its support vector  $\mathbf{y}_D$  as being the received error vector.*

## 3.2 Absorbing sets under integer update conditions

Keeping in mind Remark 3.3, in defining absorbing sets for the  $(u | s + b)$  Variant of Gallager B, we consider the all zero word as the sent codeword. This motivates the following definition.

**Definition 3.4.** *A  $(u | s + b)$  absorbing set is a set of variable nodes that, under the  $u \geq s + b$  update rule, the set of variable nodes is not eventually correct.*

**Definition 3.5.** Given  $b \in \mathbb{Z}$  and a code  $C(H)$  of length  $n$  with Tanner graph  $T(H) = G$  and set  $V$  of variable nodes,

$$(u | s + b)_1 = \{\mathbf{y}_D \in \mathbb{F}_2^n \mid D \subseteq V \text{ and } \mu^{(0)}(v) = \mu^{(t)}(v) \forall v \in D \text{ and } \forall t \in \mathbb{N}\}$$

is the set of vectors such that the state of variable nodes never changes under the  $u \geq s + b$  update rule and

$$(u | s + b)_2 = \{\mathbf{y}_D \in \mathbb{F}_2^n \mid D \subseteq V \text{ such that } \exists i, j \in \mathbb{N} \text{ with } i \neq j \text{ and } \mu^{(i)}(v) = \mu^{(j)}(v) \forall v \in D\}$$

is the set of vectors whose states are exactly those obtained in a prior iteration of the decoding algorithm under the  $u \geq s + b$  update rule. We will refer to the sets  $D \subseteq V$  with  $\mathbf{y}_D \in (u | s + b)_1$  (resp.,  $\mathbf{y}_D \in (u | s + b)_2$ ) as  $(u | s + b)_1$  or Type 1 (resp.,  $(u | s + b)_2$  or Type 2) absorbing sets.

In either type of absorbing set, the set of variable nodes is not eventually correct under the  $u \geq s + b$  update rule. Because the state of the variable nodes remains unchanged when decoding all elements of  $(u | s + b)_1$ , we see that the conditions to be in  $(u | s + b)_1$  are a special case of the requirements to be in  $(u | s + b)_2$ . Hence, we have the following observation.

**Remark 3.6.** *For any integer  $b$ ,*

$$(u | s + b)_1 \subseteq (u | s + b)_2.$$

We now give an example of a Type 1 absorbing set.

**Example 3.7.** *Consider the Tanner graph structure given in Figure 4. Suppose that the vector that is received from the channel is  $\mathbf{y} = (1, 1)$ . Then we have:  $\mu^{(0)}(v_1) = 1$  and  $\mu^{(0)}(v_2) = 1$ . We calculate  $\mu^{(0)}(c_1) = \mu^{(0)}(c_2) = 0$  and  $\mu^{(0)}(c_3) = 1$ . Because neither  $v_1$  nor  $v_2$  has as many or more unsatisfied than satisfied neighbors,  $\mu^{(1)}(v_1) = \mu^{(1)}(v_2) = 1$ . It is clear that the state of the variable nodes  $\mu^{(t)}(v_i)$  will be the same regardless of  $t \in \mathbb{N}$ . Hence, we get  $\mathbf{y} \in (u | s)_1$ .*

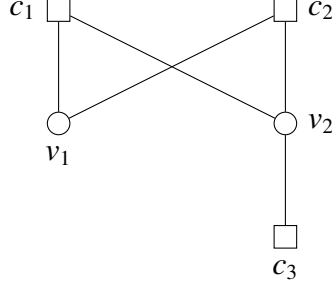


Figure 4: The Tanner graph of a  $(u | s)_1$  absorbing set.

We can nicely classify Type 1 absorbing sets. In fact, any  $(u | s)_1$  absorbing set is also a  $(u | s + 1)_1$  absorbing set. This is captured in the following theorem.

**Theorem 3.8.** *All  $(u | s)_1$ -absorbing sets are  $(u | s + 1)_1$ -absorbing sets.*

*Proof.* Let  $V$  denote the set of variable nodes. Assume  $S \subseteq V$  is an arbitrary  $(u | s)_1$  absorbing set, meaning for all  $v \in S$ ,  $D_e(v) > D_o(v)$  holds. Initializing all variable nodes in  $S$  with 1, we have  $D_e(v) + 1 > D_o(v)$  for all  $v \in S$ . Therefore,  $u \geq s + 1$  is never satisfied, implying that  $S$  is also a  $(u | s + 1)_1$  absorbing set.  $\square$

In fact, we can prove something stronger.

**Corollary 3.9.** *If  $(u | s)_1$  is an absorbing set, then, for any  $k \in \mathbb{N} \cup \{0\}$ ,  $(u | s + k)_1$  is an absorbing set.*

*Proof.* We will prove this using mathematical induction. Let  $k = 0$ . By assumption,  $S$  is a  $(u | s)_1$  absorbing set, thus the base case is true.

Now assume that for some  $k \in \mathbb{N}$ ,  $S$  is a  $(u | s + k)_1$  absorbing set. We need to prove that  $S$  is also a  $(u | s + (k + 1))_1$  absorbing set. Suppose, for the sake of contradiction, that  $S$  is not a  $(u | s + (k + 1))_1$  absorbing set. According to the well-ordering principle, there must be a smallest  $k' \in \mathbb{N}$  such that  $S$  is not a  $(u | s + k')_1$  absorbing set. Given our inductive hypothesis, this  $k'$  must be  $k + 1$ . However, since  $k$  is the smallest value where  $S$  is not a  $(u | s + (k + 1))_1$  absorbing set, and by Theorem 3.8, if  $S$  is a  $(u | s + k)_1$  absorbing set, it must also be a  $(u | s + (k + 1))_1$  absorbing set. This leads to a contradiction. Therefore,  $S$  must be a  $(u | s + (k + 1))_1$  absorbing set.

By the principle of mathematical induction,  $S$  is a  $(u | s + k)_1$  absorbing set for all  $k \in \mathbb{N}$ .  $\square$

Interestingly, the results of Theorem 3.8 and Corollary 3.9 do not analogously hold for Type 2 absorbing sets. This is demonstrated in the example that follows.

**Example 3.10.** *In this example, we will demonstrate that  $(u | s)_2 \not\subseteq (u | s + 1)_2$ . Consider the subgraph of an LDPC code with the Tanner graph from Figure 5. We have that the vector  $(1, 1, 0, 0) \in (u | s)_2$ , since check nodes  $c_1, c_2, c_3, c_4$  would indicate to flip and  $c_5, c_6, c_7$  would indicate correct states. Under the  $u \geq s$  update rule, this would cause  $v_1, v_2, v_3, c_4$  to all flip; using similar reasoning, we have that they would all flip again. Hence,  $(1, 1, 0, 0) \in (u | s)_2$ . Now under the  $u \geq s + 1$  update rule, we instead would have only the variable nodes  $v_1, v_2$  flip as  $c_6, c_7$  would be in a correct state, but the inequality  $D_o(v_3) \geq D_e(v_3) + 1$  would not be true. The same is true for  $v_4$ . Hence, we would get  $v_1 = v_2 = v_3 = v_4 = 0$  so the vector would be successfully decoded so  $(1, 1, 0, 0) \notin (u | s + 1)_2$ .*

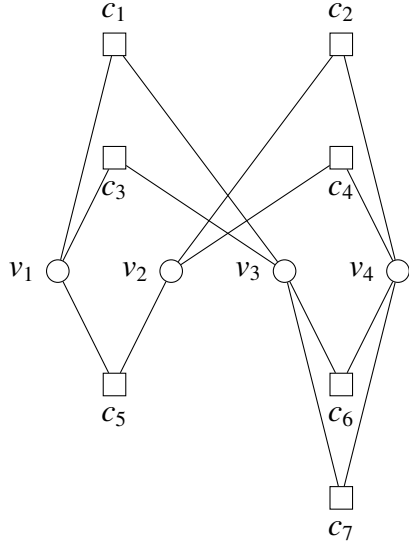


Figure 5: The Tanner graph of a  $(u \mid s)_2$  absorbing set that is not a  $(u \mid s + 1)_2$  absorbing set.

Given specific Tanner graph structure requirements, we can prove that a set of variable nodes is always a Type 1 absorbing set in certain circumstances. One of these circumstances is where all variable nodes in a subset  $S \subseteq V$  have exactly some number  $t$  more even degree than odd degree neighbors in  $G_S$ . We begin with an example to demonstrate this situation, then follow with Theorem 3.12.

**Example 3.11.** Consider the absorbing set defined by the graph in Figure 6. Notice that in this graph, each variable node has exactly 2 more even degree than odd degree neighbors.

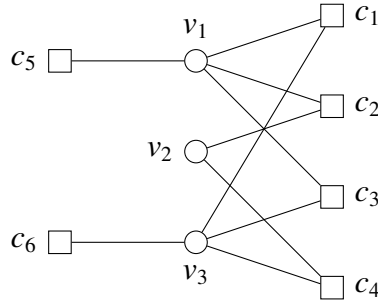


Figure 6: The Tanner graph of an absorbing set in which all variable nodes have exactly 2 more even degree than odd degree neighbors.

It can be verified that this graph gives a  $(u \mid s - 1)_1$  absorbing set but is not a  $(u \mid s - 2)_1$  absorbing set. This observation is generalized in Theorem 3.12.

**Theorem 3.12.** If every  $v \in S$  is such that  $v$  has exactly  $t \in \mathbb{Z}$  more even degree than odd degree neighbors in  $G_S$ , then  $S$  is not an  $(u \mid s - t)_1$  absorbing set. However, in this situation,  $S$  is a  $(u \mid s - k)_1$  absorbing set for all  $k < t$ .

*Proof.* Assuming that for all  $v \in S$ , the equality  $D_e(v) = D_o(v) + t$  holds, where  $t \in \mathbb{Z}$ . If every variable node in  $S$  is initialized with 1, then  $D_e(v) = s$  and  $D_o(v) = u$ . Under the update rule  $u \geq s - t$ , this would not be an absorbing set, as  $D_o(v) = D_e(v) - t$  would cause every variable node to switch on the first iteration of the algorithm. Under the update rule  $u \geq s - k$  where  $k < t$ , this would be an absorbing set, as  $u \geq u + t - k$  would never be satisfied. Hence  $S$  is not a  $(u | s - t)_1$  absorbing set but is a  $(u | s - k)_1$  absorbing set for any  $k < t$ .  $\square$

**Theorem 3.13.** *A subset  $S \subseteq V$  is a  $(u | s - t + 1)_1$  absorbing set if and only if for all  $v \in S$ ,*

$$D_o(v) \leq D_e(v) + t.$$

*Proof.* For the forward implication, let  $S \subseteq V$  be a  $(u | s - t + 1)_1$  absorbing set. Initializing each variable node with 1, we get for each  $v \in S$  that  $|U_v| < |S_v| + t$  (otherwise one of the variable nodes would change in the second iteration). Given that  $D_o(v) = |U_v|$  and  $D_e(v) = |S_v|$ , we obtain the following inequality:  $D_o(v) < D_e(v) + t$ .

For the converse, assume that for all  $v \in S$ ,  $v$  has at least  $t$  more even degree than odd degree neighbors. Then the inequality  $D_e(v) \geq D_o(v) + t$  holds. If every variable node is initialized with 1, then  $D_e(v) = s$  and  $D_o(v) = u$ , so we have the inequality  $s - t \geq u$ . The update rule  $u \geq s - t + 1$  is never satisfied, as  $s - t \geq u \geq s - t + 1$  has no solutions. So none of the variable nodes change, and  $S$  is a  $(u | s - t + 1)_1$  absorbing set.  $\square$

Notice that Theorem 3.13 is a complete classification of Type 1 absorbing sets. Therefore, we can make this a definition similar to the standard absorbing set definition given in Section 3.1.

**Definition 3.14.** *A  $(u | s + b)_1$  absorbing set is a subset  $D \subseteq V$  of variable nodes such that for each  $v \in D$ ,  $D_o(v) < D_e(v) + b$ .*

Under the standard definition of absorbing sets and the  $u \geq s$  update rule, all codewords are absorbing sets. We conclude this section by showing that a similar fact holds for the  $u \geq s + b$  update rule.

**Theorem 3.15.** *For  $b \geq 0$ , all codewords are  $(u | s + b)_1$  and  $(u | s + b)_2$  absorbing sets.*

*Proof.* Assume that  $\mathbf{c} \in C = C(H)$  and  $(V, W; E) = T(H)$  for some parity-check matrix  $H$ . Since  $\mathbf{c} \in C$ , we have  $|U_v| = 0$  for each variable node  $v \in V$ . Consequently, we get  $|S_v| = \mathcal{N}(v)$ . Therefore, the inequality  $|U_v| < |S_v| + b$  always holds. Thus,  $\mu^{(0)}(v) = \mu^{(t)}(v)$  for all  $t \in \mathbb{N}$ . By Remark 3.6, we conclude that  $\mathbf{c} \in (u | s + b)_1 \subseteq (u | s + b)_2$ .  $\square$

### 3.3 Absorbing sets under fractional update conditions

The update rule we considered in Section 3.2 updates when a variable node has a set number more (or less, depending on the sign of  $b$ ) unsatisfied than satisfied neighboring check nodes. When the variable nodes in the Tanner graph have the same (or relatively close to the same) degree, this mimics the situation where we update when a set fraction of neighboring nodes are unsatisfied. However, when variable nodes have wildly different degrees, this can result in unstable decoder behavior. This can be rectified by considering a decoder variant that requires that some fraction of check node neighbors are unsatisfied instead of a specific number.

In the standard setting, this fraction is  $\frac{1}{2}$ , i.e. if  $u$  is the number of unsatisfied check node neighbors of a variable node  $v$  and  $s$  is the number of satisfied check node neighbors of  $v$ ,  $v$  changes if  $\frac{u}{u+s} \geq \frac{1}{2}$ . Building upon the framework built in Section 3.2, we can generalize the definition of an absorbing set under general fractional update rules so that we update when  $\frac{u}{u+s} \geq \frac{a}{b}$ , where both  $a, b \in \mathbb{N}$  and  $a < b$ .

This iterative decoder framework can be viewed as a slight modification of the  $(u \mid s + b)$  Gallager B Decoder Variant given in Section 2 by replacing step 3 with the following:

3b.' At each  $v \in V$ , let  $U_v = U \cap \mathcal{N}(v)$ . Then set

$$\mu^{(t+1)}(v) = \begin{cases} \mu^{(t)}(v) + 1 & \text{if } \frac{|U_v|}{|\mathcal{N}(v)|} \geq \frac{a}{b} \\ \mu^{(t)}(v) & \text{otherwise.} \end{cases} \quad (2)$$

Increment  $t$  by 1.

We will call this fractional iterative decoding variant  $\left(\frac{a}{b}\right)$  decoding. We define absorbing sets in this setting in a similar way as in Section 3.2.

**Definition 3.16.** Given  $b \in \mathbb{Z}$  and a code  $C(H)$  of length  $n$  with Tanner graph  $T(H) = G$  and set  $V$  of variable nodes,

$$\left(\frac{a}{b}\right)_1 = \{\mathbf{y}_D \in \mathbb{F}_2^n \mid D \subseteq V \text{ and } \mu^{(0)}(v) = \mu^{(t)}(v) \forall v \in D \text{ and } \forall t \in \mathbb{N}\}$$

is the set of vectors such that the state of variable nodes never changes under the  $\frac{|U_v|}{|\mathcal{N}(v)|} \geq \frac{a}{b}$  update rule and

$$\left(\frac{a}{b}\right)_2 = \{\mathbf{y}_D \in \mathbb{F}_2^n \mid D \subseteq V \text{ such that } \exists i, j \in \mathbb{N} \text{ with } i \neq j \text{ and } \mu^{(i)}(v) = \mu^{(j)}(v) \forall v \in D\}$$

is the set of vectors whose states are exactly those obtained in a prior iteration of the decoding algorithm under the  $\frac{|U_v|}{|\mathcal{N}(v)|} \geq \frac{a}{b}$  update rule. We will refer to the sets  $D \subseteq V$  with  $\mathbf{y}_D \in \left(\frac{a}{b}\right)_1$  (resp.,  $\mathbf{y}_D \in \left(\frac{a}{b}\right)_2$ ) as  $\left(\frac{a}{b}\right)_1$  or Type 1 (resp.,  $\left(\frac{a}{b}\right)_2$  or Type 2) absorbing sets.

As in Section 3.2, we observe the following.

**Remark 3.17.** For any  $a, b \in \mathbb{N}$  with  $a < b$ ,

$$\left(\frac{a}{b}\right)_1 \subseteq \left(\frac{a}{b}\right)_2$$

We now give an example of an  $\left(\frac{a}{b}\right)_1$  absorbing set in the specific case where  $a = 2$  and  $b = 3$ . This example both serves to explain the  $\left(\frac{a}{b}\right)$  decoder and to demonstrate Definition 3.16.

**Example 3.18.** We will show that the Tanner graph in Figure 7 corresponds to a  $\left(\frac{2}{3}\right)_1$  absorbing set.

Initialize  $\mu^{(0)}(v_1) = \mu^{(0)}(v_2) = \mu^{(0)}(v_3) = \mu^{(0)}(v_4) = 1$ . Then we calculate that  $\mu^{(0)}(c_1) = \mu^{(0)}(c_2) = \mu^{(0)}(c_3) = 0$  and  $\mu^{(0)}(c_4) = \mu^{(0)}(c_5) = \mu^{(0)}(c_6) = \mu^{(0)}(c_7) = 1$ . We notice that  $\frac{|U_{v_1}|}{|\mathcal{N}(v_1)|} = 0 < \frac{2}{3}$ ,  $\frac{|U_{v_2}|}{|\mathcal{N}(v_2)|} = \frac{2}{4} = \frac{1}{2} < \frac{2}{3}$ ,  $\frac{|U_{v_3}|}{|\mathcal{N}(v_3)|} = \frac{1}{3} < \frac{2}{3}$ , and  $\frac{|U_{v_4}|}{|\mathcal{N}(v_4)|} = \frac{3}{5} < \frac{2}{3}$ . So none of the variable nodes update, and  $\mu^{(0)}(v_i) = \mu^{(k)}(v_i)$  for all  $k \geq 0$  and  $i \in [4]$ . This shows that the Tanner graph in Figure 7 corresponds to a  $\left(\frac{2}{3}\right)_1$  absorbing set.

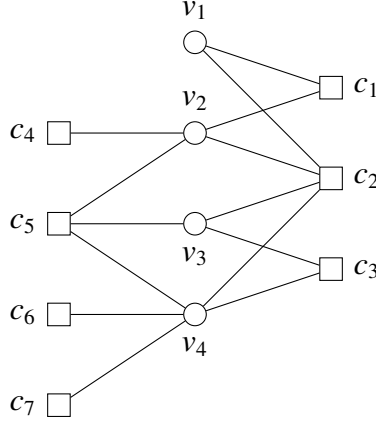


Figure 7: The Tanner graph of a  $\left(\frac{2}{3}\right)_1$  absorbing set.

Using the techniques and results built up in Section 3.2, we jump to completely classifying Type 1 absorbing sets in the  $\left(\frac{a}{b}\right)$  update setting.

**Theorem 3.19.** *A subset  $S \subseteq V$  is a  $\left(\frac{a}{b}\right)_1$  absorbing set if and only if for all  $v \in S$ ,*

$$\left(\frac{b-a}{a}\right) D_o(v) < D_e(v).$$

*Proof.* Assume  $S$  is an  $\left(\frac{a}{b}\right)_1$  absorbing set. By definition, this means that in the context of an LDPC code using the Gallager B decoding algorithm variant, the nodes in  $S$  never change under our update rule. Specifically, for  $S$  to be an  $\left(\frac{a}{b}\right)_1$  absorbing set, for each  $v \in S$ ,  $\frac{|U_v|}{|\mathcal{N}(v)|} < \frac{a}{b}$ . Note that in the induced subgraph  $G_S$ ,  $|U_v| = D_o(v)$  and  $|\mathcal{N}(v)| = |U_v| + |S_v| = D_o(v) + D_e(v)$ . So we have the condition  $\frac{D_o(v)}{D_o(v)+D_e(v)} < \frac{a}{b}$  for each  $v \in S$ . Rearranging this inequality, we obtain  $\left(\frac{b-a}{a}\right) D_o(v) < D_e(v)$ .

Conversely, assume that for all  $v \in S$ , the inequality  $\left(\frac{b-a}{a}\right) D_o(v) < D_e(v)$  holds. We need to show that this condition is sufficient for  $S$  to be an  $\left(\frac{a}{b}\right)_1$  absorbing set. In the induced subgraph  $G_S$ ,  $D_o(v) = |U_v|$ ,  $D_e(v) = |S_v|$ , and  $D_o(v) + D_e(v) = |\mathcal{N}(v)|$  for all  $v \in S$ . Rearranging this inequality, we obtain  $\frac{D_o(v)}{D_o(v)+D_e(v)} < \frac{a}{b}$ . So we have  $\frac{|U_v|}{|\mathcal{N}(v)|} < \frac{a}{b}$ . For any  $v \in S$ , the decoder will change only when  $\frac{|U_v|}{|\mathcal{N}(v)|} \geq \frac{a}{b}$ , which never happens. So  $S$  is a  $\left(\frac{a}{b}\right)_1$  absorbing set.  $\square$

**Definition 3.20.** *Under  $\left(\frac{a}{b}\right)$  decoding, a Type 1 absorbing set is a subset  $D \subseteq V$  of variable nodes such that for each  $v \in D$ ,  $\left(\frac{b-a}{a}\right) D_o(v) < D_e(v)$ .*

## 4 Connection to Boolean functions

In this section, we consider the connection between parity-check codes and Boolean functions. We use the standard notation  $\vee$  for logical or and  $\oplus$  for exclusive or. For a vector  $y \in \mathbb{F}_2^n$ ,  $\bar{y} := y + \mathbb{1} \in \mathbb{F}_2^n$ .

Notice that given an arbitrary parity-check matrix  $H$  and the Tanner graph  $T = (V, W; E)$  associated with  $H$ , we may consider the function  $g : \{0, 1\}^{|V|} \rightarrow \{0, 1\}$  defined by

$$g(\mathbf{y}) = \bigvee_{c \in W} \bigoplus_{v_i \in \mathcal{N}(c)} y_i. \quad (3)$$

As the next result shows, another way to determine whether a vector is in the code is use the Boolean function in (3) associated with the corresponding Tanner graph.

**Theorem 4.1.** *For a parity-check matrix  $H$  with the Tanner graph  $T = (V, W; E)$  and the associated Boolean function in (3),  $g(\mathbf{y}) = 1$  if and only if  $\mathbf{y} \notin C(H)$ .*

*Proof.* Given an arbitrary Tanner graph  $T = (V, W; E)$ , assume that  $\mathbf{y} \in C$  where  $C$  is the corresponding code. As  $\mathbf{y} \in C$ , for all  $c \in W$ , we have  $\mu^{(0)}(c) = 0 \pmod 2$ . Hence, we get  $\bigoplus_{v_i \in N(c)} y_i = 0$ . Since this is true for all check nodes, we get  $\bigvee_{c \in W} \bigoplus_{v_i \in N(c)} y_i = 0$ .

Conversely, assume that  $\mathbf{y} \notin C$ , where  $C$  is the corresponding code. As  $\mathbf{y} \notin C$ , some check node  $c \in W$  is not satisfied, hence we get  $\mu^{(0)}(c) = 1 \pmod 2$ . This implies  $\bigoplus_{v_i \in N(c)} y_i = 1$ .  $\square$

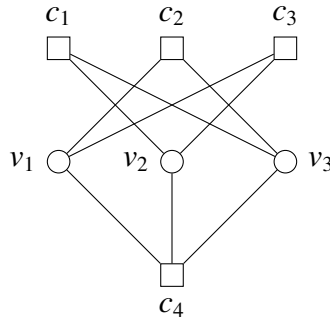


Figure 8: The Tanner graph whose Boolean function is calculated in Example 4.2.

**Example 4.2.** *Calculating the Boolean function from the graph in Figure 8, we have*

$$g(\mathbf{y}) = y_2 \oplus y_3 + y_1 \oplus y_3 + y_1 \oplus y_2 + y_1 \oplus y_2 \oplus y_3$$

*Using the equality  $x \oplus y = \bar{x} \cdot y + x \cdot \bar{y}$  and De Morgan's law we obtain*

$$g(\mathbf{y}) = \bar{y}_2 \cdot y_3 + y_2 \cdot \bar{y}_3 + \bar{y}_1 \cdot y_3 + y_1 \cdot \bar{y}_3 + \bar{y}_1 \cdot y_2 + y_1 \cdot \bar{y}_2 + y_1 \cdot y_2 \cdot y_3 + \bar{y}_1 \cdot \bar{y}_2 \cdot y_3 + \bar{y}_1 \cdot y_2 \cdot \bar{y}_3 + y_1 \cdot \bar{y}_2 \cdot \bar{y}_3.$$

*We have that the code defined by Figure 8 is just  $C = \{(0, 0, 0)\}$ . Then we have*

$$\begin{aligned} g(0, 0, 0) &= 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 0 \cdot 0 + 1 \cdot 1 \cdot 0 + 1 \cdot 0 \cdot 1 + 0 \cdot 1 \cdot 1 = 0 \\ g(0, 0, 1) &= 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 0 \cdot 1 + 1 \cdot 1 \cdot 1 + 1 \cdot 0 \cdot 0 + 0 \cdot 1 \cdot 0 = 1 \\ g(0, 1, 0) &= 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 \cdot 0 + 1 \cdot 0 \cdot 0 + 1 \cdot 1 \cdot 1 + 0 \cdot 0 \cdot 1 = 1 \\ g(0, 1, 1) &= 0 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 \cdot 1 + 1 \cdot 0 \cdot 1 + 1 \cdot 1 \cdot 0 + 0 \cdot 0 \cdot 0 = 1 \\ g(1, 1, 1) &= 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 \cdot 1 + 0 \cdot 0 \cdot 1 + 0 \cdot 1 \cdot 0 + 1 \cdot 0 \cdot 0 = 1 \end{aligned}$$

*The remaining vectors result in an output of 1 as well, and this can either be directly calculated or be observed by the symmetry of the function.*

In Example 4.2 it can be seen that the Boolean function that is constructed is equivalent to the Boolean function  $g'(\mathbf{y}) = y_1 + y_2 + y_3$  and hence it should be observed that this construction does not generally produce a minimum length Boolean expression. It is possible to create a minimum-length Boolean expression using a tool such as a Karnaugh map. However, it should be noted that Karnaugh maps are only practical for up to four variables [19]. Even so, they are useful for visualizing the code. For example, in the function  $g'(\mathbf{y}) = y_1 + y_2 + y_3$ , it can be seen that the code represents just the zero vector.

## 5 Conclusion

This research has explored absorbing sets of LDPC codes when using a variant of the Gallager B decoding algorithm. In particular, we introduced definitions that classify absorbing sets of two Gallager B decoding variants. We also introduced a connection between codes, graphical code representations, and Boolean functions. The results presented here have the potential to lead to algorithmic enhancements to overcome existing challenges in coding theory, which will ultimately contribute to more robust and efficient communication systems. A potential future research direction stemming from this work is to examine the prevalence of absorbing set influence in decoder failure. Future research may consider hybrid decoding approaches that leverage the strengths of multiple algorithms. Additionally, deeper theoretical investigations into the behavior of absorbing sets under various decoding methods will be essential for the development of more advanced error-correction techniques.

## Acknowledgements

The authors thank Andrea Escalante, Julian Jackson, Elizabeth Yen, and Tunchanok Yutitumsatit, whose discussions helped inspire Theorem 3.8 and Corollary 3.9. Support for this project was provided by the Commonwealth Cyber Initiative, the Virginia Tech Department of Mathematics and Office of Undergraduate Research, and the MAA Tensor Foundation.

## References

- [1] Changyan Di, David Proietti, I. Emre Telatar, Thomas J. Richardson, and Rüdiger L. Urbanke. Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Transactions on Information Theory*, 48(6):1570–1579, 2002.
- [2] Lara Dolecek. On absorbing sets of structured sparse graph codes. In *2010 Information Theory and Applications Workshop (ITA)*, pages 1–5, 2010.
- [3] Lara Dolecek, Zhengya Zhang, Venkat Anantharam, Martin J. Wainwright, and Borivoje Nikolic. Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes. *IEEE Transactions on Information Theory*, 56(1):181–201, 2009.
- [4] Jon Feldman, Martin J. Wainwright, and David R. Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51(3):954–972, 2005.

- [5] Jon Feldman, Martin J Wainwright, and David R Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51(3):954–972, 2005.
- [6] Robert Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.
- [7] Robert Gallager. Low-density parity-check codes, 1963.
- [8] Homayoon Hatami, David G. M. Mitchell, Daniel J. Costello, and Thomas Fuja. Performance bounds for quantized LDPC decoders based on absorbing sets. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 2539–2543, 2016.
- [9] W. Cary Huffman and Vera Pless. *Fundamentals of error-correcting codes*. Cambridge University Press, 2010.
- [10] Dennis Hui, Sara Sandberg, Yufei Blankenship, Mattias Andersson, and Leefke Grosjean. Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE. *IEEE Vehicular Technology Magazine*, 13(4):60–69, 2018.
- [11] Christine A. Kelley and Deepak Sridhara. Pseudocodewords of Tanner graphs. *IEEE Transactions on Information Theory*, 53(11):4013–4038, 2007.
- [12] Gyu Bum Kyung and Chih-Chun Wang. Finding the exhaustive list of small fully absorbing sets and designing the corresponding low error-floor decoder. *IEEE Transactions on Communications*, 60(6):1487–1498, 2012.
- [13] David J. C. MacKay. Near Shannon limit performance of low density parity check codes. *Electronics Letters*, 32:1645–1646(1), August 1996.
- [14] David J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999.
- [15] David J. C. MacKay and Radford M. Neal. Good codes based on very sparse matrices. In Colin Boyd, editor, *Cryptography and Coding*, pages 100–111, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.
- [16] Thomas J. Richardson, , M. Amin Shokrollahi, and Rüdiger L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, 2001.
- [17] Thomas J. Richardson and Shrinivas Kudekar. Design of low-density parity check codes for 5G new radio. *IEEE Communications Magazine*, 56(3):28–34, 2018.
- [18] Thomas J. Richardson and Rüdiger Urbanke. *Modern coding theory*. Cambridge university press, 2008.
- [19] K.H. Rosen. *Discrete Mathematics and Its Applications*. McGraw-Hill, 2019.
- [20] R. Michael Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.