On the pseudocodeword weight and parity-check matrix redundancy of linear codes

Christine A. Kelley
Department of Mathematics
The Ohio State University
Columbus, OH 43210, USA.
Email: ckelley@math.ohio-state.edu

Deepak Sridhara
Seagate Technology
1251 Waterfront Place
Pittsburgh, PA 15222, USA.
Email: {deepak.sridhara}@seagate.com

***Abstract*— The minimum pseudocodeword weight $w_{\min}$ of a linear graph-based code is more influential in determining decoding performance when decoded via iterative and linear programming decoding algorithms than the classical minimum distance $d_{\min}$ under standard maximum-likelihood decoding. Moreover, unlike the minimum distance which is unique to the code regardless of representation, the set of pseudocodewords, and therefore also the minimum pseudocodeword weight, depends on the graph representation used in decoding as well as on the communication channel. This means that a judicious choice of parity-check matrix is crucial for realizing the best potential of any graph-based code. In this paper, we introduce the notion of *pseudoweight redundancy* for the memoryless binary symmetric channel (BSC). Analogous to the stopping redundancy in the literature, this parameter gives the smallest number of rows needed for a parity-check matrix to have $d_{\min} = w_{\min}$. We provide some upper bounds on the BSC-pseudoweight redundancy and illustrate the concept with some results for Hamming codes, tree-based and finite geometry LDPC codes, Reed-Muller codes and Hadamard codes.**

I. Introduction

Finite-length analysis of iterative and linear programming (LP) decoding of low-density parity-check (LDPC) codes has received considerable interest over the last couple of years. The non-convergence of the iterative decoder or an LP decoder to a codeword in the code has been attributed to the presence of pseudocodewords of the LDPC graph representation [2], [3], [4], [5], [6]. These pseudocodewords are considered to be valid codewords in any finite cover of the base LDPC Tanner graph (including the degree-one cover which is the base graph itself) or as valid configurations on the iterative decoder's computation tree. While the former definition includes all the pseudocodewords of an LP decoder and a subset of the pseudocodewords of an iterative decoder, the latter definition includes all the pseudocodewords of an iterative decoder. An important parameter that characterizes the behavior of these decoders is the minimum pseudocodeword weight (or, the minimum pseudoweight). Since the graph representation determines the set of pseudocodewords, the minimum pseudoweight also depends on the graph representation. The pseudoweight for pseudocodewords arising on an any finite cover is well-defined for several channels such as the binary erasure channel (BEC), the binary symmetric channel (BSC), and the additive white Gaussian noise (AWGN) channel. A natural question is *"For a given code, which graph representation has the largest minimum pseudocodeword weight?"* Alternatively, since the minimum pseudoweight is at most the minimum distance of the code, *"Given a code, which graph representation, if any, has minimum pseudocodeword weight equal to the minimum distance of the code?"* With such a graph representation, one can expect the performance with LP or iterative decoding to come close to that of ML decoding, especially at high signal to noise ratios (SNRs). The above question has been partially addressed in the case of the BEC in [1] since pseudocodewords are essentially stopping sets for the BEC and the pseudocodeword weight is essentially the stopping set size. In this paper, we extend the results of [1] to the BSC case. Hence, throughout this paper, we consider pseudocodewords and pseudocodeword weights for the BSC.

A linear $[n, k, d]$ code $\mathbb{C}$ has blocklength n, dimension k, and minimum distance d, and may be defined as the null-space of a matrix H, where the rows of H span the dual code $\mathbb{C}^{\perp}$. (So, H is not unique.) This *parity-check matrix* H typically has $n-k$ linearly independent rows, though some of our recent results in [4] and those observed by Feldman in [2] indicate that the addition of redundant rows to a parity-check matrix can increase the minimum pseudocodeword weight of the corresponding graph representation of this matrix, and therefore improve iterative/linear-programming decoding performance. Thus, it is interesting to find the minimum number of redundant rows that must be added in order to make the minimum pseudoweight equal to the minimum distance of the code.

As defined in [1], let the redundancy $r(\mathbb{C})$ denote the minimum number of rows in a parity-check matrix for $\mathbb{C}$. (That is, if $\mathbb{C}$ has block length n, then $r(\mathbb{C}) = n - k$.) We define the **pseudoweight redundancy** $\phi(\mathbb{C})$ of $\mathbb{C}$ as the minimum number of rows in a parity-check matrix H such that the minimum pseudoweight of H is equal to the minimum distance $d(\mathbb{C})$ of $\mathbb{C}$. The significance of the pseudoweight redundancy is that the performance with LP decoding (and to some extent, that of iterative decoding) of a code represented by a parity-check matrix having $\phi(\mathbb{C})$ rows can converge to the performance with maximum-likelihood (ML) decoding at high channel SNRs. Note that while it is still a challenge to find a parity-check

[1]This work was primarily done when the authors were at the Fields Institute and the Institut für Mathematik at the Universität Zürich, respectively.

matrix H having $\phi(\mathbb{C})$ rows and $w_{\min}(H) = d(\mathbb{C})$, we present some results on $\phi(\mathbb{C})$ for several different linear codes.

In this paper we introduce the notion of the pseudoweight redundancy for the BSC and provide upper bounds on the pseudoweight redundancy both for general linear block codes and for specific families of linear block codes. Several of these results are analogues of the results for the stopping redundancy in [1], though the proofs for the pseudoweight redundancy are considerably more involved and are not simple extensions of the proofs for the stopping redundancy.

Other related works include some improved bounds on the stopping redundancy [7] and a detailed investigation of the stopping redundancy of Reed-Muller codes [8]. In addition, stopping sets are related to trapping sets, and in [9] the tradeoff between redundancy and trapping set size with respect to error floors is investigated for LDPCs on the AWGN channel.

The paper is organized as follows. We first introduce some preliminary definitions and results in section II. In particular, we show that the parity-check matrix H^* containing all the non-zero codewords of the dual code $\mathbb{C}^\perp$ has the property that that the minimum pseudoweight of H^* equals the minimum distance $d(\mathbb{C})$. As a simple illustration, the pseudoweight redundancy is derived for the codes that have a cycle-free Tanner graph representation in Section III. In Section IV, we examine the pseudoweight redundancy of codes constructed from existing codes and for each case, compare it with the pseudoweight redundancy of the component codes. We also present some concrete results on the pseudoweight redundancy for Hamming codes. In Section V we derive some upper and lower bounds on the pseudoweight redundancy for tree-based LDPC codes and finite-geometry based LDPC codes, as originally introduced in [10] and [11]. Bounds on the pseudoweight redundancy of Reed-Muller codes obtained using the $\{u, u+v\}$ construction are given in Section VI, and subsequent bounds for Hadamard codes are obtained. Section VII outlines some future directions and concludes the paper.

II. Preliminaries and initial results

An LDPC code is defined by a sparse parity-check matrix H or equivalently, its incidence graph which is a sparse bipartite graph G (also called, a Tanner graph). The left vertices are called *variable* (or, *codebit*) nodes and the right vertices are called *check* (or, *constraint*) nodes. The set of codewords is the set of all binary assignments to the variable nodes such that at each check node, the modulo two sum of the variable node assignments connected to the check node is zero, i.e., the parity-check constraint involving the neighboring variable nodes is satisfied.

To analyze the performance of LP decoding or graph-based iterative decoding, we will first define pseudocodewords of the LDPC parity-check matrix H or its Tanner graph G. Since the set of pseudocodewords arising from finite degree graph covers has an elegant mathematical description and are tractable [4], [5], we confine our analysis in this paper to this set. A pseudocodeword arising from a graph-cover is defined below. For a definition of a finite covering graph and examples of pseudocodewords on these covers, see [4], [5].

Definition 2.1: *Let $\hat{v}_{i,1}, \hat{v}_{i,2}, \ldots, \hat{v}_{i,\ell}$ denote the variable nodes in a degree ℓ lift $\hat{G}$ that correspond to a variable node v in a base Tanner graph G. Suppose $\hat{\mathbf{c}} = (\hat{c}_{1,1}, \hat{c}_{1,2}, \ldots, \hat{c}_{1,\ell}, \hat{c}_{2,1}, \ldots, \hat{c}_{2,\ell}, \ldots)$ is a codeword in the Tanner graph $\hat{G}$ representing a degree ℓ cover of G. A* pseudocodeword $\mathbf{p}$ of G *is a vector $(p_1, p_2, \ldots, p_n)$ obtained by reducing a codeword $\hat{\mathbf{c}}$, of the code in the lift graph $\hat{G}$, in the following way:*

$$\hat{\mathbf{c}} = (\hat{c}_{1,1}, \ldots, \hat{c}_{1,\ell}, \hat{c}_{2,1}, \ldots, \hat{c}_{2,\ell}, \ldots) \rightarrow \left(\frac{\hat{c}_{1,1}+\hat{c}_{1,2}+\cdots+\hat{c}_{1,\ell}}{\ell}, \frac{\hat{c}_{2,1}+\hat{c}_{2,2}+\cdots+\hat{c}_{2,\ell}}{\ell}, \ldots\right) = (p_1, p_2, \ldots, p_n) = \mathbf{p},$$

where $p_i = \frac{\hat{c}_{i,1}+\hat{c}_{i,2}+\cdots+\hat{c}_{i,\ell}}{\ell}$.

From the above definition, it is easy to show that for an LDPC constraint graph G, a pseudocodeword $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ is a vector that satisfies the following set of inequalities:

$$0 \le p_i \le 1, \quad \text{for } i = 1, 2, \ldots, n. \tag{1}$$

and, if variable nodes i_1, i_2, ..., i_d participate in a check node of degree d, then the pseudocodeword components satisfy

$$p_{i_j} \le \sum_{k=1,2,..d, k\neq j} p_{i_k}, \text{ for } j = 1, 2, .., d. \tag{2}$$

Note that the above set of pseudocodewords can also be described elegantly by means of a polytope, called the fundamental polytope [5], [6]. In the rest of the paper, we will consider pseudocodewords and their behavior on the binary symmetric channel (BSC). The weight of a pseudocodeword $\mathbf{p}$ on the BSC is defined as follows [12].

Definition 2.2: *Let $\mathbb{C}$ be a binary linear code and H a parity-check matrix for $\mathbb{C}$. Let $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ be a pseudocodeword of H. Let e be the smallest number such that the sum of the e largest components of $\mathbf{p}$ is at least the sum of the remaining components of $\mathbf{p}$. Then, the* weight *of $\mathbf{p}$ is*

$$w_{BSC}(\mathbf{p}) = \begin{cases} 2e, & \text{if } \sum_{e \text{ largest}} p_i = \sum_{\text{remaining}} p_i \\ 2e-1, & \text{if } \sum_{e \text{ largest}} p_i > \sum_{\text{remaining}} p_i \end{cases}$$

Since we only consider the weight over the BSC throughout this paper, we will drop the subscript in the above weight definition and simply refer to $w(\mathbf{p})$ as the weight of a pseudocodeword.

Definition 2.3: *Let $\mathbb{C}$ be a binary linear code and let H be a parity check matrix for $\mathbb{C}$. The* **pseudo-distance** *of H on the BSC channel is defined as the minimum pseudocodeword weight over all pseudocodewords of H and is denoted by $w_{\min}(H)$.*

Note that when a pseudocodeword $\mathbf{p}$ of H is a codeword in $\mathbb{C}$, the weight of the pseudocodeword $w(\mathbf{p})$ is equal to its Hamming weight. Thus, for any binary linear code $\mathbb{C}$ with arbitrary parity-check matrix H, $w_{\min}(H) \le d(\mathbb{C})$.

Definition 2.4: *Let $\mathbb{C}$ be a binary linear code with minimum Hamming distance $d(\mathbb{C})$. Then the* **pseudoweight redundancy**

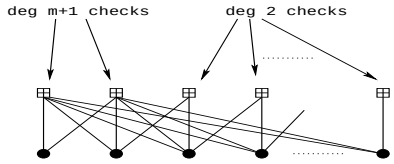

Fig. 1. A Tanner graph with $m+1$ variable nodes and $m+1$ check nodes.

of $\mathbb{C}$ is defined as the smallest integer $\phi(\mathbb{C})$ such that there exists a parity-check matrix H for $\mathbb{C}$ with $\phi(\mathbb{C})$ rows and $w_{\min}(H) = d(\mathbb{C})$.

The following theorem shows that the pseudoweight redundancy of a code $\mathbb{C}$ is, indeed, well-defined.

Theorem 2.1: *Let $\mathbb{C}$ be a binary linear code, and let H^* denote the parity-check matrix for $\mathbb{C}$ consisting of all the codewords of the dual code $\mathbb{C}^\perp$. Then $w_{\min}(H^*) = d(\mathbb{C})$.*

In [3], it is observed that H^* does not always eliminate all non-codeword pseudocodewords. We point out that this does not contradict the above theorem, as non-codeword pseudocodewords may also assume weights $\geq d(\mathbb{C})$.

Theorem 2.1 says that any binary linear code has $\phi(\mathbb{C}) \leq 2^{n-k}$. However such a parity-check matrix representation makes iterative or linear programming decoding too complex for practical use. The rest of this paper focuses on finding tighter upper bounds on the pseudoweight redundancy for various families and constructions of codes. The code in next example shows that in some cases, just one additional (redundant) check equation is sufficient to achieve a good parity-check matrix representation.

Example 2.1: Figure 1 shows the Tanner graph of a code $\mathbb{C}$ having $m+1$ variable nodes and $m+1$ check nodes. In [4], it was shown that this code has minimum distance m or $m+1$, depending on whether m is even or odd, respectively. However, the minimum BSC pseudocodeword weight is 2 for both cases. By adding just one redundant check node, namely, the row $(1, 0, 0, \ldots, 0)$ when m is even and the row $(1, 0, 0, \ldots, 0, 1)$ when m is odd, the minimum pseudocodeword weight becomes equal to the minimum distance. Hence, $\phi(\mathbb{C}) = m + 2$.

In fact, additional redundancy is not always necessary, as in the case for codes with $d(\mathbb{C}) \leq 3$.

Theorem 2.2: *Let $\mathbb{C}$ be a binary linear code with minimum distance $d(\mathbb{C}) \leq 3$. Then any parity-check matrix H for $\mathbb{C}$ satisfies $w_{\min}(H) = d(\mathbb{C})$.*

III. Tree-codes and Repetition codes

As an example, consider the pseudoweight redundancy for the family of linear codes that have a cycle-free representation (i.e., tree-codes) or a single cycle representation. Any linear code which has a cycle-free Tanner graph representation has no non-codeword pseudocodewords [4] and thus, $w_{\min} = d_{\min}$ for this representation. Furthermore, it is easy to show that the cycle-free representation corresponds to a parity-check matrix with $r(\mathbb{C})$ rows and thus, $\phi(\mathbb{C}) = r(\mathbb{C})$ for these codes.

An $[n, n-1, 2]$ single parity check (SPC) code $\mathbb{C}_{spc}$ and an $[n, 1, n]$ repetition code $\mathbb{C}_{rpc}$ are examples of linear codes with a cycle-free Tanner graph representation. For the $[n, n-1, 2]$ SPC code, the parity-check matrix $[11\ldots1]$ is a matrix with a cycle-free representation and has $\phi(\mathbb{C}_{spc}) = 1$. For the $[n, 1, n]$ repetition code, the $(n-1) \times n$ matrix H where row i has a 1 in positions one and $i+1$ for $i = 1, 2, \ldots, n-1$ is a parity-check matrix for $\mathbb{C}_{rpc}$ that has a cycle-free Tanner graph. Any pseudocodeword for H has form $\mathbf{p} = \alpha \cdot \mathbf{1}$, and the weight of any pseudocodeword $\mathbf{p}$ when all the components in its support are the same can be shown to be [4]

$$w(\mathbf{p}) = |supp(\mathbf{p})| = n.$$

(Note that the e value in Definition 2.2 is $n/2$ for such a pseudocodeword.) Hence, $\phi(\mathbb{C}_{rpc}) = n - 1$. Note that the repetition code also has a Tanner graph representation with a single cycle. This single cycle Tanner graph also has no non-codeword pseudocodewords. However, the corresponding parity-check matrix has one additional row in this representation than that for the cycle-free representation.

IV. Constructions of codes from other codes

In this section, we consider codes constructed from smaller component codes and relate the pseudoweight redundancy of the constructed codes with that of the component codes.

Theorem 4.1: *Let $\mathbb{C}_1$, $\mathbb{C}_2$ be (n_1, k_1, d_1), (n_2, k_2, d_2) binary linear codes, respectively. Then $\mathbb{C}_3 = \{(u, v) : u \in \mathbb{C}_1, v \in \mathbb{C}_2\}$ is an $(n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\})$ code with*

$$\phi(\mathbb{C}_3) \leq \phi(\mathbb{C}_1) + \phi(\mathbb{C}_2) \tag{3}$$

Proof: Suppose H_1 and H_2 are parity-check matrices for $\mathbb{C}_1$ and $\mathbb{C}_2$, having $\phi(\mathbb{C}_1)$ and $\phi(\mathbb{C}_2)$ rows, respectively, such that $w_{\min}(H_1) = d(\mathbb{C}_1)$ and $w_{\min}(H_2) = d(\mathbb{C}_2)$. Then

$$H = \begin{bmatrix} H_1 & \mathbf{0} \\ \mathbf{0} & H_2 \end{bmatrix}.$$

is a parity check matrix for $\mathbb{C}$ with $\phi(\mathbb{C}_1) + \phi(\mathbb{C}_2)$ rows. Let $P = (p_1, p_2, \ldots, p_{n_1}, q_1, q_2, \ldots, q_{n_2})$ be a pseudocodeword in H. Then, $\mathbf{p} = (p_1, \ldots, p_{n_1})$ is a pseudocodeword in H_1 and $\mathbf{q} = (q_1, \ldots, q_{n_2})$ is a pseudocodeword in H_2. Let e_1 be the smallest integer such that the sum of the e_1 largest components in $\mathbf{p}$ is at least the sum of the remaining components in $\mathbf{p}$. Define e_2 for $\mathbf{q}$ analogously. Suppose e is the smallest number such that the sum of the e largest components of P is at least the sum of the remaining components in P. Then clearly, $\min\{e_1, e_2\} \leq e \leq e_1 + e_2$. Hence, the weight of P is at least $2e - 1 \geq \min\{d(\mathbb{C}_1), d(\mathbb{C}_2)\} = d(\mathbb{C})$. Thus, $w_{\min}(H) = d(\mathbb{C})$ and hence $\phi(\mathbb{C}) \leq \phi(\mathbb{C}_1) + \phi(\mathbb{C}_2)$. ∎

Theorem 4.2: *Let $\mathbb{C}_1$ be an (n, k, d) binary linear code. Then the code $\mathbb{C}_2 = \{(u, u) : u \in \mathbb{C}_1\}$ is an $(2n, k, 2d)$ code with*

$$\phi(\mathbb{C}_2) \leq \phi(\mathbb{C}_1) + n \tag{4}$$

Proof: Suppose H_1 is a parity-check matrix for $\mathbb{C}_1$ having $\phi(\mathbb{C}_1)$ rows and the property that $w_{\min}(H_1) = d(\mathbb{C}_1) = d$. Then

$$H_2 = \begin{bmatrix} H_1 & \mathbf{0} \\ \mathbf{I}_n & \mathbf{I}_n \end{bmatrix}.$$

is a parity check matrix for $\mathbb{C}_2$ with $\phi(\mathbb{C}_1) + n$ rows. Any pseudocodeword of H_2 has the form $P = (p_1, p_2, \ldots, p_n, p_1, p_2, \ldots, p_n)$ since the last n rows of H_2 impose the constraint that the 1st component of P equals the $(n+1)^{th}$ component, the second component equals the $(n+2)^{th}$ component and so on. Furthermore, $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ is a pseudocodeword of H_1. Let e be the smallest integer such that the sum of the e largest components of $\mathbf{p}$ is at least the sum of the remaining components of $\mathbf{p}$. Then, the weight of $\mathbf{p}$ is $2e - 1 \geq d$. Suppose now, e_2 is the smallest number such that the sum of e_2 components of P is at least the sum of the remaining components of P, then clearly $e_2 = 2e$. and furthermore, the weight of P is $2e_2 \geq 2d = d(\mathbb{C}_2)$. Thus, $\phi(\mathbb{C}_2) \leq \phi(\mathbb{C}_1) + n$. ∎

Theorem 4.3: *Let $\mathbb{C}$ be an $(n, k, 3)$ binary linear code. Then the extended code $\mathbb{C}'$ is an $(n+1, k, 4)$ code with*

$$\phi(\mathbb{C}') \leq 2\phi(\mathbb{C}) \tag{5}$$

Proof: Let H be a parity-check matrix for $\mathbb{C}$ with $\phi(\mathbb{C})$ rows and $w_{\min}(H) = d(\mathbb{C})$. Let $\bar{H}$ be the binary complement of H. Then

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

is a parity-check matrix for $\mathbb{C}'$, where $\mathbf{0}$ and $\mathbf{1}$ are the all-zeros and all-ones column vectors, respectively. Let $\mathbf{p}' = (p_1, p_2, \ldots, p_n, p_{n+1})$ be a pseudocodeword of H'. Then clearly, $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ is a pseudocodeword of H.

Suppose e is the smallest number such that the sum of the e largest components of $\mathbf{p}$ is at least the sum of the remaining components, then we know that $e \geq 3$ since $w_{\min}(H) = 3 = d(\mathbb{C})$. Therefore, any $p_i < \sum_{j \neq i, j=1}^{n} p_j$, for $i = 1, 2, \ldots, n$. Furthermore any two columns in H will always contain either a 01 among its rows or a 10 or both since any two columns in H cannot be identical as $d(\mathbb{C}) = 3$. Therefore, any two columns of $\bar{H}$ will either contain a 01, a 10, or both among its rows.

Suppose e' is the smallest number such that the sum of the e' largest components of $\mathbf{p}'$ is at least the sum of the remaining components in $\mathbf{p}'$. Then we need to show that $e' \geq 2$ if the sum of the e' largest components equals the sum of the remaining components and that $e' > 2$ if the sum of e' largest components exceeds the sum of the remaining components. Suppose p_1 and p_{n+1} are the largest components in $\mathbf{p}'$. Then, applying the inequality of equation (2) at a row c among the first n rows of H' that contains a 1 in the first column yields $p_1 \leq \sum_{j \in N_c} p_j$, where N_c represents all the neighbors of check node c excluding variable node 1. Applying the inequality of equation (2) in row $n + c$ yields $p_{n+1} \leq \sum_{j \in N_{c+n}} p_j$, where N_{c+n} is the set of neighbors of check node $n+c$ excluding variable node $n+1$. Notice that the two sets N_c and N_{c+n} are non-intersecting since the row $n+c$ is the binary complement of row c. Combining the above two inequalities, we get $p_1 + p_{n+1} \leq \sum_{j \neq 1, n+1} p_j$. Thus, either $e' \geq 2$ if $p_1 + p_{n+1} = \sum_{j \neq 1, n+1} p_j$ or $e' > 2$. Suppose p_1 and p_2 were the dominant components of $\mathbf{p}'$. Then, there is a row among the first n rows that contains a 10 or a 01 among its first two columns. Let us suppose row c contains a 10 among its first two columns. Then applying equation (2) at row c, we get $p_1 \leq \sum_{j \in N_c} p_j$ where N_c is the set of neighbors of check node c excluding variable node 1. Further, $2 \notin N_c, n+1 \notin N_c$. Applying the pseudocodeword inequality at row $n + c$, we get $p_2 \leq \sum_{j \in N_{c+n}} p_j$ where, N_{n+c} is the set of neighbors of check node $n + c$ excluding variable node 2. Again, N_{c+n} and N_c are non-intersecting. And by the previous argument, $e' \geq 2$ if $p_1 + p_2 = \sum_{j=3}^{n+1} p_j$ and $e' > 2$ if $p_1 + p_2 > \sum_{j=3}^{n+1} p_j$. Thus, the weight of $\mathbf{p}'$ is at least 4, thereby proving that $w_{\min}(H') = d(\mathbb{C}') = 4$. This shows that $\phi(\mathbb{C}') \leq 2\phi(\mathbb{C})$. ∎

A. Hamming codes and extended Hamming codes

For the class of binary Hamming codes, we can derive some concrete results on the pseudoweight redundancy based on the results obtained so far.

Corollary 4.4: Any $[2^m - 1, 2^m - 1 - m, 3]$ binary Hamming code $\mathbb{C}$ has $\phi(\mathbb{C}) = m$.

The proof follows from Theorem 2.2 and the fact that $r(\mathbb{C}) = m$.

Corollary 4.5: Any $[2^m, 2^m - m - 1, 4]$ binary extended Hamming code $\mathbb{C}$ has $\phi(\mathbb{C}) \leq 2m$.

The proof follows by combining the previous result with Theorem 4.3.

V. Structured constructions of LDPC codes

We examine the pseudoweight redundancy for the tree-based LDPC codes that we introduced in [10]. The tree-based LDPC code constructions are notable for their relatively small gaps between minimum distance and minimum pseudocodeword weight of the corresponding tree-based parity-check representation. The two-dimensional finite-geometry (FG) based LDPC codes occur as a special case of the tree-based construction.

Theorem 5.1: *Let $\mathbb{C}$ be a binary Type II, $\ell = 3$ tree-based LDPC code with the tree-based parity-check representation H having $n = p^{2s} + p^s + 1$ rows and columns and row and column weight equal to $p^s + 1$. Then, if $p = 2$, $\phi(\mathbb{C}) \leq n$. If $p > 2$, then $\mathbb{C}$ is a repetition code and $\phi(\mathbb{C}) = n - 1$.*

Proof: We prove in [10] that $w_{\min}(H) = d(\mathbb{C})$ for $p = 2$ and therefore, $\phi(C) \leq n$ by construction. The second statement follows from the result in Section III. ∎

The Type II, $\ell = 3$ tree-based LDPC codes are described by parity-check matrices that are $p^s + 1$ regular having $n =$

$p^{2s}+p^s+1$ rows and columns each and are equivalent to the 2-dimensional projective geometry (PG) over $GF(p^s)$ - binary LDPC codes. We prove this equivalence in [10] and show that for the tree-based parity-check matrix in the Type II $\ell = 3$ construction, $w_{\min}(H) = d(\mathbb{C}) = p^s + 2$ when $p = 2$ and the codes are repetition codes with $d(\mathbb{C}) = n$ when $p > 2$. Thus, the two-dimensional PG LDPC codes have $\phi(\mathbb{C}) \leq n$ for $p = 2$ and $\phi(\mathbb{C}) = n-1$ for $p > 2$ (from results in Section III). We can also show that for the 2-dimensional Euclidean geometry (EG) LDPC codes (formed by excluding the origin point in the corresponding geometry), the block length is $n = p^{2s}-1$ and a corresponding tree-based graph representation in [10] has n variable nodes and n check nodes and $w_{\min}(H) = d(\mathbb{C}) = p^s + 1$ when $p = 2$. Thus $\phi(\mathbb{C}) \leq n = 2^{2s} - 1$ for the two-dimensional EG LDPC codes constructed from the geometry over $GF(2^s)$. We are investigating the pseudoweight redundancy for the finite geometry LDPC codes obtained from higher dimensional Euclidean and projective geometries as well as the Type-I LDPC codes from the tree-based construction.

The Type II, $\ell = 4$ tree-based LDPC codes are equivalent to those obtained from finite generalized quadrangles.

Conjecture 5.2: *Let $\mathbb{C}$ be Type II, $\ell = 4$ tree-based LDPC code with the tree-based parity-check representation H having $n = p^{3s}+p^{2s}+p^s+1$ rows and columns and row and column weights equal to $p^s + 1$. Then, if $p = 2$, $\phi(\mathbb{C}) \leq n$.*

We believe that the tree-based parity-check representation in [10] achieves $w_{\min}(H) = d(\mathbb{C}) = 2+2p^s+p^{2s}$ for $p = 2$ and hence $\phi(\mathbb{C}) \leq n$ for these codes. Furthermore, we conjecture that the LDPC codes from generalized polygons [4], [15] with parity check matrices that are 2^s+1 regular, have $w_{\min} = d_{\min}$ and $\phi(\mathbb{C}) \leq n$, the block length of the codes.

VI. Reed-Muller Codes

We assume the reader is familiar with the $\{u, u+v\}$ construction of Reed-Muller codes (see e.g., [13]). Let $R(r,m)$ denote the binary Reed-Muller code of order r. $R(r,m)$ has length 2^m, dimension $k = \sum_{i=0}^{r} \binom{m}{i}$, and distance 2^{m-r}. We show that for the recursive parity-check matrix representation of these codes introduced in [1], the minimum pseudocodeword weight equals the minimum distance.

Let $H(r,m)$ denote the parity-check matrix for $R(m-r-1,m) = R(r,m)^{\perp}$. (Thus, the distance of the code $R(m-r-1,m)$ is 2^{r+1}.) Then the recursive construction in [1] shows that for all positive integers m and for all $r = 0, 1, \ldots, m-1$,

$$H(r,m) = \begin{pmatrix} H(r,m-1) & H(r,m-1) \\ \mathbf{0} & H(r-1,m-1) \\ H(r-1,m-1) & \mathbf{0} \end{pmatrix}$$

Furthermore, $H(m,m) = I_{2^m}$ and $H(0,m) = (111\ldots1)$. Note that $R(m-1,m)$ is the single parity check code of distance 2 and block length 2^m. By the result in Section III, the pseudo-distance $w_{\min}(H(0,m)) = d(R(m-1,m)) = 2$ and its pseudoweight redundancy is 1.

Lemma 6.1: *The pseudo-distance of $H(m-1,m)$ is 2^m. (That is, $w_{\min}(H(m-1,m)) = 2^m$.)*

Proof: We will use induction on m. It is easy to check that $H(0,m)$ has a pseudo-distance equal to 2. Observe that

$$H(m-1,m) = \begin{bmatrix} I_{2^{m-1}} & I_{2^{m-1}} \\ \mathbf{0} & H(m-2,m-1) \end{bmatrix}.$$

By the induction hypothesis, the pseudo-distance $w_{\min}$ of $H(m-2,m-1)$ is 2^{m-1}. Just as in the proof of theorem 4.2, any pseudocodeword of $H(m-1,m)$ has the form $\mathbf{p} = (p_1, p_2, \ldots, p_{2^{m-1}}, p_1, p_2, \ldots, p_{2^{m-1}})$. Furthermore, $\mathbf{p}' = (p_1, p_2 \ldots, p_{2^{m-1}})$ is a pseudocodeword of $H(m-2,m-1)$. From the above argument, suppose e' is the smallest number such that the sum of e' largest components of $\mathbf{p}'$ is at least the sum of the remaining components of $\mathbf{p}'$, then $e' \geq w_{\min}(H(m-2,m-1))/2 = 2^{m-2}$. Suppose e is the smallest number such that the sum of the e largest components of $\mathbf{p}$ is at least that of the remaining components, then $e = 2e'$. Thus, the weight of $\mathbf{p}$ is $2e \geq 2^m$. This proves that the pseudo-distance of $H(m-1,m)$ is $w_{\min}(H(m-1,m)) = 2^m$. ∎

Theorem 6.1: *The pseudo-distance of $H(r,m)$ is 2^{r+1} for all positive integers m and for all $r = 0, 1, \ldots, m-1$.*

We will only give a sketch of the proof due to lack of space.

Proof sketch: Let $\mathbf{p} = (p_1, p_2, \ldots, p_{2^{m+1}})$ be a pseudocodeword of $H(r,m+1)$. Using the recursive construction above, $H(r,m+1)$ can be written as

$$H(r,m+1) = \begin{pmatrix} H(r,m) & H(r,m) \\ \mathbf{0} & H(r-1,m) \\ H(r-1,m) & \mathbf{0} \end{pmatrix}.$$

Repeating the recursion for $m-r+1$ times, we obtain

$$H(r,m+1) = \begin{pmatrix} H(r,r) & H(r,r) & \ldots & H(r,r) \\ \vdots & \vdots & \ldots & \vdots \end{pmatrix}.$$

Since $H(r,r) = I_{2^r}$, we consider the first 2^r rows in the above. The first row involves columns $1, 2^r+1, 2^{2r}+1, \ldots, 2^{m+1}-2^r+1$. The second row involves columns $2, 2^r+2, 2^{2r}+2, \ldots, 2^{m+1}-2^r+2$, and so on. Suppose e is the smallest number such that the sum of e largest values of $\mathbf{p}$ is at least the sum of the remaining components.

Let us consider two cases: Case 1: Assume that the first 2^r largest components in $\mathbf{p}$ occur one each from the sets $\{p_1, p_{2^r+1}, p_{2^{2r}+1}, \ldots, p_{2^{m+1}-2^r+1}\}$, $\{p_2, p_{2^r+2}, p_{2^{2r}+2}, \ldots, p_{2^{m+1}-2^r+2}\}$, $\ldots$, $\{p_{2^r}, p_{2^r+2^r}, \ldots, p_{2^{m+1}}\}$ and without loss of generality, assume the first 2^r components of $\mathbf{p}$ are $p_1, p_2 \ldots, p_{2^r}$. Then, applying the pseudocodeword inequality in equation (2) at the first 2^r rows of $H(r,m+1)$, we obtain

$$\begin{aligned} p_1 &\leq p_{2^r+1} + p_{2^{2r}+1} + \cdots + p_{2^{m+1}-2^r+1} \\ p_2 &\leq p_{2^r+2} + p_{2^{2r}+2} + \cdots + p_{2^{m+1}-2^r+2} \\ &\vdots \\ p_{2^r} &\leq p_{2^{2r}} + p_{2^{3r}} + \cdots + p_{2^{m+1}} \end{aligned}$$

Summing all the above inequalities, we have $p_1 + p_2 + \cdots + p_{2^r} \leq \sum_{i=2^r+1}^{2^{m+1}} p_i$. Thus $e \geq 2^r$ and the weight of $\mathbf{p}$ is $2e \geq 2^{r+1}$.

Note that upon recursively expanding $H(r, m+1)$ $m-r+1$ times, the sub-matrix of $H(r, m+1)$ excluding the first 2^r rows of $H(r, m+1)$ are of the form

$$H(r, m+1) = \begin{pmatrix} \vdots & \ddots & \ddots & \ddots & \vdots \\ H(0, m-r+1) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H(0, m-r+1) & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & H(0, m-r+1) \end{pmatrix}$$

Note that $H(0, m-r+1)$ is an all ones vector of length $m-r+1$, i.e., $H(0, m-r+1) = (11\ldots1)$.

Case 2: Assume that the first 2^r largest components of $\mathbf{p}$ occur one each from the sets $\{p_1, p_2 \ldots, p_{2^{m-r+1}}\}$, $\{p_{2^{m-r+1}+1}, \ldots, p_{2^{m-r+2}}\}$, $\ldots$, $\{p_{2^m+1}, \ldots, p_{2^{m+1}}\}$. (Without loss of generality, assume the 2^r largest components in $\mathbf{p}$ are $p_1, p_{2^{m-r+1}+1}, p_{2^{m-r+2}+1}, \ldots, p_{2^m+1}$. Then, applying the pseudocodeword inequality in equation (2) at all the rows excluding the first 2^r rows of $H(r, m+1)$, we obtain

$$p_1 \leq p_2 + p_3 + \cdots + p_{2^{m-r+1}}$$
$$p_{2^{m-r+1}+1} \leq p_{2^{m-r+1}+2} + p_{2^{m-r+1}+3} + \cdots + p_{2^{m-r+2}}$$
$$\vdots$$
$$p_{2^m+1} \leq p_{2^m+2} + p_{2^m+3} + \cdots + p_{2^{m+1}}$$

Summing all the above inequalities, we have $p_1 + p_{2^{m-r+1}+1} + \cdots + p_{2^m+1} \leq \sum_{rest} p_i$. Thus $e \geq 2^r$ and $w(\mathbf{p}) = 2e \geq 2^{r+1}$.

It is easy to combine the two cases above when the largest 2^r largest components of $\mathbf{p}$ are distributed in any other manner than those described above. In such a case, we apply the pseudocodeword inequality at some of the rows in the first 2^r rows of $H(r, m+1)$ and at some of the rows in the remaining rows of $H(r, m+1)$ and arrive at the conclusion that $e \geq 2^r$. This proves that the weight of $\mathbf{p}$ is $2e \geq 2^{r+1}$. □

Schwartz and Vardy prove that the number of rows in $H(r, m)$ is

$$g(r, m) = \sum_{i=0}^{r} \binom{m-r-1+i}{i} 2^i$$

for all $r = 0, \ldots, m-1$. From the previous result, the pseudo-distance of $H(r, m)$ is equal to the distance of the code $R(m-1-r, m)$. Thus, we have

Theorem 6.2: *The pseudoweight redundancy of $R(m-1-r, m)$ is at most $g(r, m)$.*

One type of Hadamard codes obtained from Hadamard matrices are a special case of the Reed-Muller codes. They correspond to Reed-Muller codes of $R(1, m)$ and have block length 2^m, dimension $m+1$ and distance 2^{m-1}. Thus, as a corollary of Theorem 6.2, we have

Corollary 6.3: *The pseudoweight redundancy of the $[2^{m+1}, m+1, 2^{m-1}]$ Hadamard code from the construction in [13] is at most $g(m-2, m) = (m-2)2^{m-1} + 1$.*

VII. Conclusions and future work

This paper introduced the notion of pseudoweight redundancy which provides insight as to which graph representations of a code may best realize its decoding potential under iterative or LP decoding. Upper bounds on the pseudoweight redundancy were shown for general linear codes, and the families of tree-based LDPC codes, finite-geometry based LDPC codes, and Reed-Muller codes were analyzed with respect to this parameter for the BSC. More sophisticated bounding techniques as well as extensions to other codes (such as BCH codes) and channels is in progress. Several of the results presented in the paper are easily extended when the BSC-pseudoweight is replaced by the max-fractional weight defined in [3]. We showed in [4] that the pseudoweight on the AWGN and BSC channels are lower bounded by the max-fractional weight. Thus, the results that extend to the max-fractional weight also extend to the AWGN pseudocodeword weight. These results will be presented in a future paper. A natural question is, what is the optimal level of redundancy: although the decoding performance is best when the minimum pseudocodeword weight is equal to the minimum distance, a high level of redundancy increases the complexity of the decoder. Furthermore, even if a parity-check matrix H with $\phi(\mathbb{C})$ rows and $w_{\min}(H) = d(\mathbb{C})$ is chosen for decoding, the parity-check matrix may still contain higher weight pseudocodewords that may affect the decoding performance at low to moderate SNRs.

References

[1] M. Schwartz and A. Vardy. On the stopping distance and the stopping redundancy of codes. In *Proceedings of the IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005.

[2] J. Feldman, M. J. Wainwright, D. R. Karger, Using Linear Programming to Decode Binary Linear Codes. IEEE Transactions on Information Theory, 51(3), 954 – 972, Mar. 2005.

[3] J. Feldman, *Decoding Error-Correcting Codes via Linear Programming Decoding*, Ph.D. Thesis, M.I.T., Cambridge, MA, 2003.

[4] C. Kelley and D. Sridhara. Pseudocodewords of Tanner graphs. Submitted to IEEE Trans. on Information Theory, June 2005.

[5] R. Koetter and P. 0. Vontobel. Graph-covers and iterative decoding of finite length codes. In *Proceedings of the IEEE International Symposium on Turbo Codes and Applications*, Brest, France, September 2003.

[6] R. Koetter, W.-C. W. Li, P.O. Vontobel, and J.L. Walker. Characterizations of pseudo-codewords of LDPC codes. Submitted to Journal on Advances in Mathematics of Communications, Aug. 2006.

[7] J. Han and P. H. Siegel. Improved upper bounds on stopping redundancy. *IEEE Trans. Inform. Theory*, IT-53(1):90–104, Jan. 2007.

[8] T. Etzion. On the stopping redundancy of Reed-Muller codes. *IEEE Trans. Inform. Theory*, IT-52(11):4867–4879, Nov. 2006.

[9] S. Laendner, T. Hehn, O. Milenkovic, J. B. Huber. When does one redundant parity-check equation matter? In *Proc. of GLOBECOM*, 2006.

[10] C. Kelley, D. Sridhara, and J. Rosenthal. Tree-based construction of LDPC codes having good pseudocodeword weights. IEEE Transactions on Information Theory, 53(4), pp. 1460 – 1478, April 2007.

[11] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar. Codes on finite geometries. *IEEE Trans. Inform. Theory*, IT-51(2):572–596, 2005.

[12] G. D. Forney, Jr., R. Koetter, F. Kschischang, and A. Reznik. On the effective weights of pseudocodewords for codes defined on graphs with cycles. In B. Marcus and J. Rosenthal, editors, *Codes, Systems and Graphical Models*, IMA Vol. 123, pages 101–112. Springer-Verlag, 2001.

[13] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Amsterdam, North-Holland, 1978.

[14] R. M. Tanner, D. Sridhara, A. Sridharan, D. Costello, Jr., and T. E. Fuja. LDPC block and convolutional codes based on circulant matrices. *IEEE Trans. Inform. Theory*, IT-50(12):2966–2984, 2004.

[15] Z. Liu and D. A. Pados. LDPC codes from generalized polygons. *IEEE Trans. Inform. Theory*, IT-51(11):3890 - 3898, 2005.