Graph-covers and iterative decoding of finite length codes

Ralf Koetter and Pascal O. Vontobel
Coordinated Science Laboratory
Dep. of Elect. and Comp. Eng.
University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801
E-mail: {koetter,vontobel}@uiuc.edu

Abstract: *Codewords in finite covers of a Tanner graph G are characterized. Since iterative, locally operating decoding algorithms cannot distinguish the underlying graph G from any covering graph, these codewords, dubbed pseudo-codewords are directly responible for sub-optimal behavior of iterative decoding algorithms. We give a simple characterization of pseudocodewords from finite covers and show that, for the additive, white Gaussian noise channel, their impact is captured in a finite set of "minimal" pseudo-codewords. We also show that any (j,k)-regular graph possesses asymptotically vanishing relative minimal pseudo-weight. This stands in sharp contrast to the observation that for $j > 2$ the minimum Hamming distance of a (j,k)-regular low-density parity-check code typically grows linearly with the length of the code.*

1. Introduction

While iterative, message-passing decoding algorithms have had unparalleled success, it is fair to say that their behavior for the case of finite length codes is, at present, not well understood. Nevertheless, in some cases specialized techniques give some insight into the problem. The case of iterative decoding for the erasure channel was investigated by Di et al. [8] utilizing the notion of *stopping sets*. On the other hand, the *computation tree* and *pseudo-codewords* were the basis of a finite length analysis introduced by Wiberg [3] developed in [5,6]. Finally, the idea of *near-codewords* was used by MacKay and Postol [10] to empirically characterize problematic situations for iterative decoding.

The goal of this paper is to continue the study of iterative decoding algorithms for finite length codes. It turns out that *finite* graph covers (in contrast to the universal cover) provide a powerful tool to characterize the behavior of locally operating, message-passing decoding algorithms. Not only does our analysis give a crisp and *quantifiable* design criterion for iteratively decodable codes but it also elegantly reflects and unifies the notions of stopping sets, pseudo-codewords and near-codewords.

We show that the performance of iterative decoding schemes is, even in the high SNR regime, largely dominated not by minimum distance considerations but by the notion of pseudo-weight which, loosely speaking, measures the minimum weight of an error pattern that will cause nonconvergence in the iterative decoder. This minimum pseudo-weight is shown to grow sublinearly for sequences of regular low-density parity-check (LDPC) codes, which stands in sharp contrast to the fact that their expected minimum distance grows as a linear function of the code length.

This paper is organized as follows: In Section 2 we give some basic notation relating to iterative decoding and we give an illustrative example. Sections 3 and 4 lay out the basic theory behind our analysis. Section 5 gives bounds on the effective pseudo-weight of any LDPC code. Section 6 sketches algorithmic approaches to computing the minimum pseudo-weight of a code. While many facts are stated as theorems, propositions etc. in this paper, proofs are generally omitted due to lack of space. For proofs of the claims we refer to a forthcoming paper on these issues [12].

2. Basics and an Example

Let $\mathbb{F}_2$ denote the binary field. A binary, linear code $\mathcal{C}$ of type $[n,k]$ is a k-dimensional subspace of the binary Hamming space $\mathbb{F}_2^n$. Any code of type $[n,k]$ may be specified as the nullspace of an $n \times (n-k)$ parity-check matrix $\mathbf{H}$, i.e. $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_2^n : \mathbf{H}\mathbf{c}^T = \mathbf{0}\}$.

We can associate a bipartite graph $\mathsf{G}_{\mathbf{H}}$, the so-called Tanner graph [1,2,4], with a given parity-check matrix $\mathbf{H}$ in the following way: a vertex f_i, $i = 0, 1, \ldots, n-k-1$ is created for each row in the parity-check matrix and a vertex c_j, $j = 0, 1, \ldots, n-1$ is created for each codeword position. Moreover, we create an (undirected) edge $\{f_i, c_j\}$ between f_i and c_j if and only if the entry $\mathbf{H}_{i,j}$ of the parity-check matrix is nonzero. The set of parity-check vertices f_i is denoted as $V_f = \{f_i : i = 0, 1, \ldots, n-k-1\}$ and the set of codeword position vertices c_i is denoted as $V_c = \{c_i : i = 0, 1, \ldots, n-1\}$. We will simultaneously refer to codeword position, codeword position vertices and the value of a codeword position by c_i. The edge set of $\mathsf{G}_{\mathbf{H}}$ is denoted as $E \subseteq \{\{v, u\} : v \in V_f, u \in V_c\}$. The set of neighbors of a vertex v is

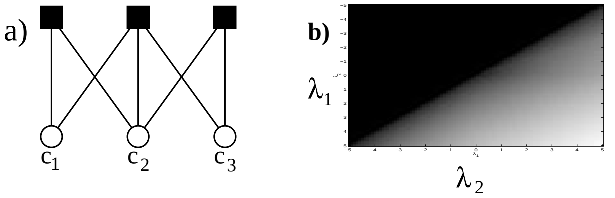

Fig. 1 **a)** Tanner graph of a trivial code of length 3 consisting of only the zero codeword. **b)** Convergence regions for the code illustrated in a). The value of λ_0 is fixed to 0.013.

defined as $\Gamma(v) = \{u : \{v, u\} \in E\}$ and the degree $\delta(v)$ of a vertex v is defined as $\delta(v) = |\Gamma(v)|$.

Let a codword $\mathbf{c} \in \mathcal{C}$ be transmitted over a noisy, memoryless channel and let a vector $\mathbf{y}$ be received. We can summarize $\mathbf{y}$ in form of a vector $\boldsymbol{\lambda} = (\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$ of log-likelihood ratios $\lambda_i = \ln \frac{\Pr(c_i=0|y_i)}{\Pr(c_i=1|y_i)}$. The decoding problem consists of finding the most likely codeword $\mathbf{c}$ given the vector $\boldsymbol{\lambda}$.

The Tanner graph of a code is the appropriate framework to describe message-passing decoding algorithms. By now, a variety of such algorithms is known, all of which may be seen as instances of the same underlying principle [2,3,9]. Most of the development in subsequent sections applies to *any locally operating* algorithm and is, thus, independent of the particular choice of message-passing algorithm. However, whenever we give experimental results we will usually use the so-called *min-sum algorithm* [3]. The difference between this algorithm and the more common sum-product algorithm is relatively small and the min-sum algorithm is more amenable to analysis.

Before we develop the theory of our approach in the next section, the following example sheds some light on some basic concepts involved:

Example 1 *We consider a trivial code $\mathcal{C}$ of length $n = 3$ and dimension $k = 0$ with parity-check matrix*

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

and Tanner graph depicted in Figure 1a. While it, at first, may seem strange to consider a zero-rate code, it is indeed an ideal candidate to investigate problematic behavior of iterative decoding. Under an optimal decision rule the decoding algorithm must output the all-zero word independently of the received log-likelihood vector $\boldsymbol{\lambda}$. On the other hand, a simple experiment reveals that the behavior of the decoding algorithm is dependent of the received vector $\boldsymbol{\lambda}$. Figure 1b depicts the convergence behavior of an iterative decoding algorithm for a fixed value of $\lambda_0 = 0.013$ as both λ_1 and λ_2 range from -5 to 5. The algorithm fails to converge after 100 iterations in the black region of the image while it converges to the zero codeword in the gray colored areas. The speed of convergence is indicated by the shade of gray. Moreover, we note that this behavior is independent of the algorithm in question (min-sum, sum-product, etc.) and it is found for virtually any locally *operating decoding method. A closer study shows that the region of convergence to the zero word is empirically well described (up to numerical accuracy) by the condition $\lambda_0 + \lambda_1 + \lambda_2 \geqslant 0$. In other words, a message-passing algorithm realizes the decoding region of a repetition code of length 3. In order to understand this behavior we consider the graph in Figure 2.*

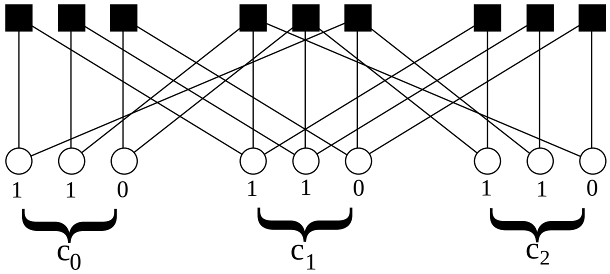

Fig. 2 A cubic cover of the graph in Figure 1a.

Figure 2 depicts a so-called cubic cover of the graph $\mathsf{G}_\mathbf{H}$ in Figure 1. The graph is obtained by replicating every node in $\mathsf{G}_\mathbf{H}$ three times and introducing edges so that the local adjacency relationships between replicated nodes is preserved. (A concise definition of a finite graph cover is given shortly). We emphasize two crucial observations:

- *In principle, locally operating decoding algorithms* cannot distinguish *if they are operating on a Tanner graph $\mathsf{G}_\mathbf{H}$ or any finite cover of this graph as, for example, the cubic cover depicted in Figure 2.*
- *The binary codes in finite covers support codewords that do not have an equivalent in the original graph. Such a codeword is indicated in Figure 2 for the cubic cover of G_H.*

It is clear, that any locally operating message-passing algorithm will automatically take into account all possible codewords in all possible covers of the original graph. In other words, the binary configuration indicated in Figure 2 will compete for the best solution along with all other valid configurations in the union of all covers. In the case of our example code, the existence of nonzero codewords in finite covers of the original graph explains the behavior of iterative decoding algorithms since it acts with respect to a received word as the all-one configuration. In other words, the codeword indicated in Figure 2 is "closer" than the all-zero word to a received word in a region that would correspond to a virtually present all-one word. Moreover, it can be shown that any *nonzero codeword in a finite cover of $\mathsf{G}_\mathbf{H}$ has the same effect as a virtually present, all-one codeword.* □

Example 1 shows how codewords in graph covers impact the performance of message-passing algorithms. At first glance it seems a formidable task to characterize all possible codewords being introduced by the union of finite covers of any degree. (The number of finite covers of a graph grows faster than exponential with the covering degree). However, it turns out that this becomes an object that itself is elegantly described and compactly represented in the original factor graph of Fig. 1.

3. Finite Graph Covers

Let a graph $\mathsf{G} = (V, E)$ be given with vertex set $V = \{v_0, v_1, \ldots, v_{\ell-1}\}$ and edge set E.

Definition 1 *A finite degree m cover of $\mathsf{G} = (V, E)$ is a graph $\hat{\mathsf{G}}$ with vertex set $\hat{V} = \bigcup_{i=0}^{\ell-1} \hat{V}_i$ where each set $\hat{V}_i = \{\hat{v}_{i,0}, \hat{v}_{i,1}, \ldots, \hat{v}_{i,m-1}\}$ contains exactly m vertices. The edge set $\hat{E}$ of $\hat{\mathsf{G}}$ is chosen as a subset of $\{\{\hat{v}_{i,s}, \hat{v}_{j,r}\} : \{v_i, v_j\} \in E, s, r \in \{0, 1, \ldots, m-1\}\}$ such that, for each vertex $\hat{v}_{i,s} \in \hat{V}$, $\delta(\hat{v}_{i,j})$ equals $\delta(v_i)$ and $\Gamma(\hat{v}_{i,s})$ contains precisely one vertex $\hat{v}_{j,r}$ for all j such that $v_j \in \Gamma(v_i)$ holds.* □

If a graph G is a Tanner graph for a code $\mathcal{C}$ of length n, a degree m cover $\hat{\mathsf{G}}$ is a Tanner graph of a code $\hat{\mathcal{C}}$ of length mn. (Any object relating to a finite cover of an underlying graph is distinguished by a $\hat{}$ symbol). Vertices in $\hat{V}_i$ are denoted as $\hat{c}_{i,0}, \hat{c}_{i,1}, \ldots, \hat{c}_{i,m-1}$ for lifted nodes $c_i \in V_c$ or $\hat{f}_{i,0}, \hat{f}_{i,1}, \ldots, \hat{f}_{i,m-1}$ for lifted nodes in $f_i \in V_f$. Any codeword in $\mathcal{C}$ can be lifted to a codeword in $\hat{\mathcal{C}}$ by assigning the value of c_i to all $\hat{c}_{i,l}$. In particular, the all-zero word in $\mathcal{C}$ will be lifted to the all-zero word in $\hat{\mathcal{C}}$. In order to characterize the effect of any nonzero word in $\hat{\mathcal{C}}$ we replicate the received values and log-likelihood ratios to obtain $\hat{y}_{i,l} = y_i$ and $\hat{\lambda}_{i,l} = \lambda_i$ for $l = 0, 1, \ldots, m-1$, thus obtaining vectors $\hat{\mathbf{y}}$ and $\hat{\boldsymbol{\lambda}}$. Let $\hat{\mathbf{c}}$ be a codeword in $\hat{\mathcal{C}}$ and let $\omega_i(\hat{\mathbf{c}})$ be defined as

$$\omega_i(\hat{\mathbf{c}}) \stackrel{\text{def}}{=} \frac{|\{l : \hat{c}_{i,l} = 1\}|}{m},$$

i.e. the fraction of times a variable in $\hat{V}_i$ assumes the value 1. The vector $\boldsymbol{\omega}(\hat{\mathbf{c}}) = (\omega_0(\hat{\mathbf{c}}), \omega_1(\hat{\mathbf{c}}), \ldots, \omega_n(\hat{\mathbf{c}}))$ plays a crucial role in characterizing the behavior of codewords in $\hat{\mathcal{C}}$.

Let the inner product of two vectors $\mathbf{a}, \mathbf{b}$ be defined as $\langle \mathbf{a}, \mathbf{b} \rangle \stackrel{\text{def}}{=} \sum a_i b_i$.

Proposition 1 *Let a vector of log-likelihood values $\boldsymbol{\lambda}$ and its lifting $\hat{\boldsymbol{\lambda}}$ be given. Moreover let two words $\hat{\mathbf{c}}$ and $\hat{\mathbf{c}}'$ in $\hat{\mathcal{C}}$ be given. We have $\Pr\{\hat{\mathbf{c}}|\hat{\boldsymbol{\lambda}}\} > \Pr\{\hat{\mathbf{c}'}|\hat{\boldsymbol{\lambda}}\}$ if and only if $\langle \boldsymbol{\omega}(\hat{\mathbf{c}}), \boldsymbol{\lambda} \rangle < \langle \boldsymbol{\omega}(\hat{\mathbf{c}'}), \boldsymbol{\lambda} \rangle$ holds.* □

The most important property of Proposition 1 is that codewords in $\hat{\mathcal{C}}$ can be effectively characterized by the vectors $\boldsymbol{\omega}(\hat{\mathbf{c}})$. Assume that $\mathbf{c} \in \mathcal{C}$ and its lifted version $\hat{\mathbf{c}}$ are the all-zero codeword in Proposition 1. It follows that pairwise decisions between $\mathbf{c}$ and a competing nonzero codeword $\hat{\mathbf{c}'} \in \hat{\mathcal{C}}$ will partition the space of $\boldsymbol{\lambda}$ into two regions separated by the hyperplane $\langle \boldsymbol{\omega}(\hat{\mathbf{c}'}), \boldsymbol{\lambda} \rangle = 0$. For any particular channel model we can compute the distance of this hyperplane from the transmitted signal point in signal space, thus effectively characterizing a type of minimum distance, the so-called *pseudo-distance* [3,5,6].

Let $||\mathbf{x}||_q = (\sum_i x_i^q)^{\frac{1}{q}}$ denote the L_q norm of a vector. For the binary antipodal signaling on an additive, white, Gaussian noise (AWGN) channel we have the following definition [3,5,6]:

Definition 2 (Pseudo-codewords) *Let $\hat{\mathbf{c}} \in \hat{\mathcal{C}}$ be a codeword in a cover of the Tanner graph G. We call $\boldsymbol{\omega} = \boldsymbol{\omega}(\hat{\mathbf{c}})$ a pseudo-codeword of $\mathcal{C}$. Its pseudo-weight $w_{\mathrm{p}}(\boldsymbol{\omega})$ on an additive, white, Gaussian noise channel is given by*

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \left(\frac{||\boldsymbol{\omega}||_1}{||\boldsymbol{\omega}||_2} \right)^2. \tag{1}$$

Let $w_{\mathrm{p}}^{\min}(\mathcal{C})$ denote the minimum pseudo-weight of all nonzero pseudo-codewords of $\mathcal{C}$ taken over all finite degree covers of G. □

Remark 1 *Note that if $\mathbf{c}$ is a codeword with Hamming weight $w_{\mathrm{H}}(\mathbf{c})$, then $\boldsymbol{\omega} = \mathbf{c}$, $||\mathbf{c}||_1 = w_{\mathrm{H}}(\mathbf{c})$ and $||\mathbf{c}||_2 = \sqrt{w_{\mathrm{H}}(\mathbf{c})}$. It follows that $w_{\mathrm{p}}(\mathbf{c}) = ||\mathbf{c}||_1^2/||\mathbf{c}||_2^2$ $= w_{\mathrm{H}}(\mathbf{c})^2/\ w_{\mathrm{H}}(\mathbf{c}) = w_{\mathrm{H}}(\mathbf{c})$.* □

The pseudo-weight measures the distance of the all-zero codeword in signal space to a pairwise decision boundary caused by a pseudo-codeword $\boldsymbol{\omega}$.

Proposition 2 *Let a binary code be used on an additive, white, Gaussian noise channel with antipodal signaling with signal alphabet $\{\pm 1\}$. Let a nonnegative vector $\boldsymbol{\omega}$ be given. The squared Euclidean distance in the signal space between the signal point $\mathbf{1}$, corresponding to the all-zero word, and the hyperplane $\langle \boldsymbol{\omega}, \mathbf{y} \rangle = 0$ is given as $w_{\mathrm{p}}(\boldsymbol{\omega})$.* □

Remark 2 *Proposition 1 is independent of the particular channel. In the space of log-likelihood ratios $\boldsymbol{\lambda}$ the pseudo distance is always proportional to the pseudo-weight of Definition 2. However, signal space is, in general, not linearly related to $\boldsymbol{\lambda}$ and we get different pseudo-distance expression for non-AWGN channels. Expressions for the pseudo-distance in the context of nonbinary signaling, the binary symmetric* *channel and the binary erasure channel can be found in [5].* □

Proposition 1, in conjunction with the fact that locally operating decoding algorithms cannot distinguish between G and $\hat{\mathsf{G}}$ motivates our subsequent task to characterize vectors $\boldsymbol{\omega}(\hat{\mathbf{c}})$ for the union of all possible finite covers. While this, at first, appears to be a difficult task, we will see in the next section that it is elegantly solved by the original Tanner graph G.

4. The Fundamental Polytope

We start this section by considering a simple parity-check code $\mathcal{C}_\delta$ of length δ and its Tanner graph consisting of a single parity-check node f_0 of degree δ and δ variable nodes $c_0, c_1, \ldots, c_{\delta-1}$. Any finite cover of degree m of G is simply an m-fold copy of the original graph G. It is particularly simple to describe the pseudo-codeword induced by these m-fold repetitions. We consider a codeword in the original parity-check code described by G as a codeword over the real numbers with elements $\{0, 1\}$. Since any individual copy of G can support any codeword from $\mathcal{C}_\delta$, the possible set of words $\omega(\hat{\mathbf{c}})$ originating from the m-fold cover can be described as the set of vectors

$$\left\{ \frac{\sum_{i=1}^{m} \mathbf{c}_i}{m} : \mathbf{c}_i \in \mathcal{C}_\delta \right\}.$$

Let a matrix $\mathbf{P}'_\delta$ be defined as the $2^{\delta-1} \times \delta$ matrix containing all binary even weight vectors. As we consider covers over larger and larger degree m, we have the following proposition:

Proposition 3 *Let a Tanner graph G_δ be given consisting of a single parity-check node of degree δ and δ variable nodes. Consider the set $\mathcal{P}$ of pseudo-codewords $\boldsymbol{\omega}(\hat{\mathbf{c}})$ taken over the union of all covers of G of all degrees $m = 1, 2, \ldots$. The closure of $\mathcal{P}$ in the real numbers is described by the polytope*

$$\begin{aligned}\mathscr{P}(\mathsf{G}_\delta) \overset{\text{def}}{=} \{&\boldsymbol{\omega} \in \mathbb{R}^n : \\ &\boldsymbol{\omega} = \mathbf{x}\mathbf{P}'_\delta, \mathbf{x} \in \mathbb{R}^{2^{\delta-1}}, 0 \leqslant x_i \leqslant 1, \sum_i x_i = 1\}\end{aligned}$$

□

Example 2 *We consider the Tanner graph of a parity-check code of length three. The polytope $\mathscr{P}(\mathcal{C}_3)$ of all possible vectors $\omega(\hat{\mathbf{c}})$ is depicted in Figure 3.* □

It is actually possible to extend Proposition 3 to a nontrivial Tanner graph G. To this end, let the restriction of a vector $\boldsymbol{\omega}$ to a set V of variable nodes be denoted as $\boldsymbol{\omega}_V$.

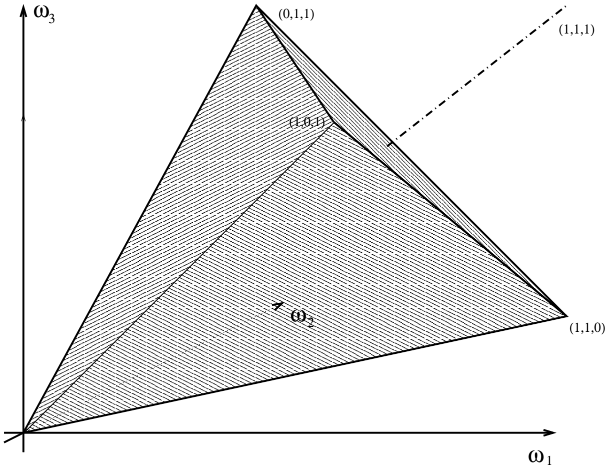

Fig. 3 The pseudo-codeword polytop for a [3,2] parity-check code.

Theorem 4 *Let a Tanner graph G be given with parity-check nodes $f_0, f_1, \ldots, f_{l-1}$ and variable nodes $c_0, c_1, \ldots, c_{n-1}$. Let $\mathcal{P}$ be the set of pseudo-codewords $\boldsymbol{\omega}(\hat{\mathbf{c}})$ taken over the union of all covers of G of all degrees $m = 1, 2, \ldots$. The closure of $\mathcal{P}$ in the real numbers is described by the polytope*

$$\begin{aligned}\mathscr{P}(\mathsf{G}) = \{&\boldsymbol{\omega} \in \mathbb{R}^n : \\ &\boldsymbol{\omega}_{\Gamma(f_i)} \in \mathscr{P}(\mathsf{G}_{\delta(f_i)}), i = 0, 1, \ldots, l-1\}.\end{aligned}$$ □

Theorem 4 gives a compact and elegant characterization of the possible vectors $\boldsymbol{\omega}$ for any given Tanner graph G. In fact, the polytope $\mathscr{P}(\mathsf{G})$ is itself compactly representable in G by choosing for variable nodes the alphabet $\mathbb{R}$ and associating with node f_i the indicator functions of the parity-check polytopes $\mathscr{P}(\mathsf{G}_{\delta(f_i)})$. $\mathscr{P}(\mathsf{G})$ is a convex body entirely inside the positive orthant and with one corner of $\mathscr{P}(\mathsf{G})$ located in the origin. To any vector $\boldsymbol{\omega}$ in $\mathscr{P}(\mathsf{G})$ we can find at least one (in general there are many) codewords $\hat{\mathbf{c}}$ in some finite cover of G such that $\boldsymbol{\omega} = \boldsymbol{\omega}(\hat{\mathbf{c}})$. Moreover, this pseudo-codeword has pseudo-distance $w_{\mathrm{p}}(\boldsymbol{\omega})$ from the all-zero codeword. Note that all multiples of the vector $\boldsymbol{\omega}$ have the same pseudo-weight. Hence, provided we relate our future discussion to the all-zero codeword we can restrict our attention to the (convex) cone that is generated by $\mathscr{P}(\mathsf{G})$. We call this object the *fundamental cone* of the graph G.

Definition 3 *Let a Tanner graph G be given with associated polytope $\mathscr{P}(\mathsf{G})$. The fundamental cone $\mathscr{F}(\mathsf{G})$ associated with G is defined as*

$$\mathscr{F}(\mathsf{G}) = \{\mu\boldsymbol{\omega} \in \mathbb{R}^n : \boldsymbol{\omega} \in \mathscr{P}(\mathsf{G}), \mu \geqslant 0\}$$

□

Assuming that the all-zero word was transmitted, Proposition 1 motivates the definition of a region $\mathscr{D}_0$ in $\mathbb{R}$ as

$$\mathscr{D}_0 = \{\boldsymbol{\lambda} \in \mathbb{R} : \langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle > 0, \forall \boldsymbol{\omega} \in \mathscr{F}(\mathsf{G})\}.$$

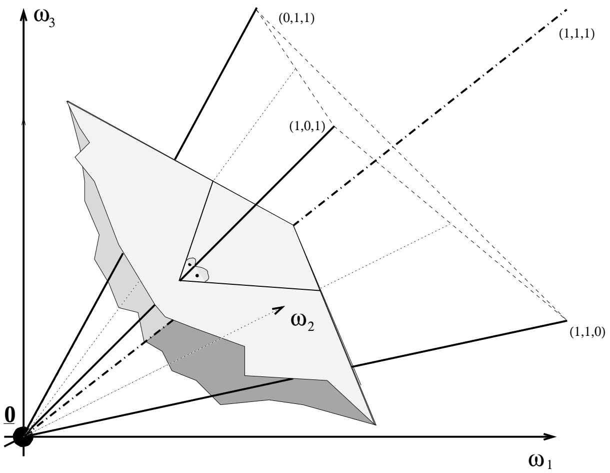

Fig. 4 Decision region in binary on-off keying due to the corners of the pseudo-codeword polytope for a [3,2] parity-check code.

$\mathscr{D}_0$ is the region where the all-zero word is more likely than any competing codeword $\hat{\mathbf{c}}$ in a finite cover.

The pseudo-weight of a vector $\boldsymbol{\omega}$ may be expressed as $w_{\mathrm{p}}(\boldsymbol{\omega}) = n(\cos(\angle(\boldsymbol{\omega}, \mathbf{1})))^2$ where $\angle(\omega, \mathbf{1})$ denotes the angle between the vector $\boldsymbol{\omega}$ and the all-one vector. Hence, the minimum pseudo-weight $w_{\mathrm{p}}^{\min}(\mathcal{C})$ is achieved by a corner of the convex cone that encloses the maximal angle with the all-one vector. Let $\mathcal{U}(\mathsf{G})$ be the set of corner points in $\mathscr{F}(\mathsf{G})$. For the AWGN channel we have the following theorem:

Theorem 5 *Let $\mathscr{F}(\mathsf{G})$ be the fundamental cone of a Tanner graph G. For the AWGN channel, the region $\mathscr{D}_0$ may be described by the cornerpoints of $\mathscr{F}(\mathsf{G})$ alone, i.e.*

$$\mathscr{D}_0 = \{\boldsymbol{\lambda} \in \mathbb{R} : \langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle > 0, \forall \omega \in \mathcal{U}(\mathsf{G})\}$$

□

Remark 3 *Theorem 5 allows us to compactly represent the region $\mathscr{D}_0$. Maximum likelihood decision regions on an AWGN channel are determined by so-called* minimal codewords *which are the subset of codewords that contribute a face to the maximum likelihood decision region polytope. Here we have a quite similar situation where for the AWGN channel again a* finite *set of* minimal pseudocodewords*, i.e. the set $\mathcal{U}(\mathsf{G})$, contributes faces to the polytope $\mathscr{D}_0$.* □

Remark 4 *For the AWGN channel, Theorem 5 translates directly into a description of the set $\mathscr{D}_0$ in signal space since $\boldsymbol{\lambda}$ depends in an affine way on the received vector $\mathbf{y}$. For example, the region $\mathscr{D}_0$ for binary on-off signaling and the fundamental cone of Figure 3 is indicated in Figure 4. However, we note that, $\boldsymbol{\lambda}$ does, for other channels, in general, not depend in an affine way on a signal space representation of the received vector. Thus, the shape of $\mathscr{D}_0$ for general channels is* not *necessarily a polytope.* □

Example 3 *Theorem 5 gives a crisp characterization of the region $\mathscr{D}_0$. We can use this characterization to investigate LDPC codes and their parameters. A particularly nice LDPC code was constructed by Tanner et al. [11]. The code is a regular (3,5)-LDPC code (all variable nodes in the Tanner graph have degree three and all check nodes have degree five), of length 155, dimension* 64 *and minimum Hamming distance 20. Its parity-check matrix of size* 93×155 *would actually suggest a $R = 2/5$ code, but because of rank loss, the actual rate is slightly higher, namely $R = 64/155 = 0.4129$.*

The underlying graph G has a girth of 8 and a diameter of 6 which, together with the relatively large minimum distance of twenty (the best known code with the the same length and dimension has minimum Hamming distance 28), makes this code an outstanding candidate for iterative decoding. However, it is relatively easy to find a pseudo-codeword in $\mathcal{U}(\mathsf{G})$ which has pseudo-weight only 16.406*. Thus the large minimum distance of the code is largely irrelevant for iterative decoding and does not determine the performance of the code. In particular, based on the automorphism group of the graph, the multiplicity of pseudo-codewords of weight* 16.406 *is, at least, 155.* □

We conclude this section with a theorem for the well understood case that the Tanner graph of a code $\mathcal{C}$ is a tree. In this case iterative decoding realizes the optimal decoding algorithm. This is nicely reflected in the shape of the fundamental cone $\mathscr{F}(\mathsf{G})$.

Theorem 6 *Let $\mathscr{F}(\mathsf{G})$ be the fundamental cone of a Tanner graph G. Moreover, assume that G is a tree. Let $\mathcal{M}$ be the set of minimal codewords of $\mathcal{C}$. The fundamental cone $\mathcal{F}(\mathsf{G})$ is generated by the set $\mathcal{M}$, i.e.*

$$\mathscr{F}(\mathsf{G}) = \{\boldsymbol{\omega} \in \mathbb{R} : \boldsymbol{\omega} = \sum_{\mathbf{c} \in \mathcal{M}} \alpha(\mathbf{c})\mathbf{c}, 0 \leqslant \alpha(\mathbf{c}) \in \mathbb{R}\}.$$

Thus, if G is a tree, $\mathscr{D}_0$ is exactly the maximum likelihood decision region of the all-zero codeword. □

5. An Upper Bound on the Minimal Pseudo-Weight

In this section we investigate the asymptotic behavior of the minimum pseudo-weight of a Tanner graph G. Let $g(\mathsf{G})$ be the girth of G, and let $\Delta(\mathsf{G})$ be its diameter. Given any variable node v in G let $\Delta_v(\mathsf{G})$ denote the maximal distance from v that any node in G can have. The code $\mathcal{C}$ is called a (j,k)-regular code if the uniform column weight of parity-check matrix $\mathbf{H}$ is j and the uniform row weight of $\mathbf{H}$ is k.

Definition 5 *We denote an arbitrary variable node v of G to be the root. We classify the remaining variable and check nodes according to their (graph) distance from the root, i.e. the root is a tier 0, all nodes at distance 1 from the root will be called nodes of tier 1, all nodes at distance 2 from the root node will be called nodes of tier 2, etc.. We call this ordering "breadth first spanning tree ordering with root v." Because of the bipartiteness of G, it follows easily that the nodes of the even tiers are variable nodes whereas the nodes of the odd tiers are check nodes. Furthermore, a check node at tier $2t+1$ can only be connected to variable nodes in tier $2t$ and possibly to variable nodes in tier $2t+2$. Note that the last variable node tier is tier $\Delta_v(\mathsf{G})$ and that the symbol nodes are at tiers $0, 2, \ldots, 2\lfloor \Delta_v(\mathsf{G})/2 \rfloor$.* □

Remark 6 *Let the Tanner graph of a binary (j,k)-regular code $\mathcal{C}$ be given and let v be an arbitrary bit node. We perform breadth first spanning tree ordering with respect to v according to Def. 5. Let $N_t(\mathcal{C})$ be the number of nodes at tier t and let $N_t^{\max} = N_{t,j,k}^{\max}$ be the maximal number of nodes possible at tier t. It is not difficult to see that $N_0^{\max} = 1$, $N_1^{\max} = j$, $N_2^{\max} = j(k-1)$, $N_3^{\max} = j(k-1)(j-1)$, $N_4^{\max} = j(k-1)(j-1)(k-1)$. In general, $N_{2t}^{\max} = j(j-1)^{t-1}(k-1)^t$ for $t > 0$ and $N_{2t+1}^{\max} = j(j-1)^t(k-1)^t$ for $t \geqslant 0$.* □

Definition 4 (Canonical completion) *Let the Tanner graph of a binary (j,k)-regular code $\mathcal{C}$ be given and let v be an arbitrary symbol node. After performing the breadth first spanning tree ordering with root v we construct a pseudo-codeword $\boldsymbol{\omega}$ in the following way. If bit i corresponds to a variable node in tier $2t$, then*

$$\omega_i \overset{\text{def}}{=} \frac{1}{(k-1)^t}. \tag{2}$$

We call this the canonical completion with root v. □

Proposition 7 *The canonical completion with root v yields a vector $\boldsymbol{\omega}$ such that $\boldsymbol{\omega}$ is in the fundamental cone $\mathscr{F}(\mathsf{G})$. The vector $\boldsymbol{\omega}$ has pseudo-weight $w_{\mathrm{p}}(\boldsymbol{\omega}) = ||\boldsymbol{\omega}||_1^2 / ||\boldsymbol{\omega}||_2^2$, where*

$$||\boldsymbol{\omega}||_1 = \sum_{t=0}^{\lfloor \Delta_v(\mathsf{G})/2 \rfloor} N_{2t}(\mathsf{G}) \frac{1}{(k-1)^t}, \tag{3}$$

$$||\boldsymbol{\omega}||_2^2 = \sum_{t=0}^{\lfloor \Delta_v(\mathsf{G})/2 \rfloor} N_{2t}(\mathsf{G}) \left(\frac{1}{(k-1)^t} \right)^2. \tag{4}$$

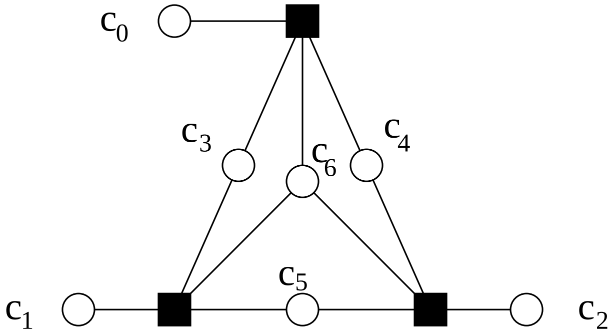

Fig. 5 Tanner graph for the [7,4,3] Hamming code.

□

For a given G, one can calculate the pseudo-weight of the pseudo-codeword given by the canonical completion for any given root; this will always yield an upper bound on $w_{\mathrm{p}}^{\min}(\mathcal{C})$.

Example 4 *We consider the Tanner graph of the $[7,4,3]$ Hamming code given in Figure 5.*

The canonical completion with root c_0 corresponds to a vector $\boldsymbol{\omega} = (1, \frac{1}{9}, \frac{1}{9}, \frac{1}{3}, \frac{1}{3}, \frac{1}{9}, \frac{1}{3})$. It is easy to check that this pseudo-codeword is indeed inside the fundamental polytope for this graph. The pseudo-weight in this case equals $\frac{(1+\frac{1}{9}+\frac{1}{9}+\frac{1}{3}+\frac{1}{3}+\frac{1}{9}+\frac{1}{3})^2}{1+\frac{1}{81}+\frac{1}{81}+\frac{1}{9}+\frac{1}{9}+\frac{1}{81}+\frac{1}{9}} = 3.973$. We note that the Tanner graph of Figure 5 also supports a pseudo-codeword $\boldsymbol{\omega}'$ of type $\boldsymbol{\omega}' = (1, 0, 0, \frac{1}{3}, \frac{1}{3}, 0, \frac{1}{3})$. The pseudo-weight of $\boldsymbol{\omega}'$ equals only three and is thus at "minimum distance" for this code. □

The canonical completion with a given root is not only a generally good candidate in order to find a pseudo-codeword of low weight but it is also a poweful enough technique to show the asymptotic behavior of the pseudo-weight by properly bounding $||\omega||_1$ and $||\omega||_2^2$.

Theorem 7 *Let $\mathcal{C}$ be a (j,k)-regular LDPC code with $3 \leqslant j < k$. Then the minimum pseudo-weight is upper bounded by*

$$w_{\mathrm{p}}^{\min}(\mathcal{C}) \leqslant \beta'_{j,k} \cdot n^{\beta_{j,k}}, \tag{5}$$

where

$$\beta'_{j,k} \triangleq \left(\frac{j(j-1)}{j-2} \right)^2, \quad \beta_{j,k} \triangleq \frac{\log\big((j-1)^2\big)}{\log\big((j-1)(k-1)\big)} < 1. \tag{6}$$

□

Corollary 8 *Consider a sequence of (j,k)-regular LDPC codes whose length goes to infinity. The relative minimum pseudo-weight (i.e. the fraction of minimum pseudo-weight to code length) must go to zero.* □

Remark 9 *Note that Corollary 8 is in sharp contrast to the fact that the relative minimum weight of a randomly generated (j, k)-regular LDPC code is lower bounded by a nonzero number with probability one for $n \to \infty$ [7].* □

Remark 10 *The different nature of pseudo-weight with respect to different channels is underlined by the fact that the canonical completion with respect to any given root yields a small pseudo-weight in the AWGN case while its normalized pseudo-weight on the erasure channel equals one. Nevertheless, the fundamental cone still characterizes the set of pseudo-codewords — it is the worst case pseudo-codeword within the fundamental cone that is different.* □

6. Relations to Stopping Sets and Near Codewords

Stopping Sets Stopping sets were introduced in [8] as a means to understand the suboptimal behavior of iterative decoding techniques for the erasure channel. It has been observed later that stopping sets seem to also reflect, to some degree, the performance of iteratively decoded codes for other channels. Let $\mathcal{S}$ be a subset of variable nodes and consider the subgraph G' of G induced by $\mathcal{S}$ and the neighbors of $\mathcal{S}$. $\mathcal{S}$ is called a stopping set if G' does not contain any check nodes of degree one.

Theorem 11 *Let $\mathbf{x}$ be a vector that equals one in a stopping set $\mathcal{S}$ and which is zero otherwise. There exists an α with $0 < \alpha \leqslant 1$ such that $\boldsymbol{\omega} = \alpha\mathbf{x}$ is a pseudo-codeword of pseudo-weight $|\mathcal{S}|$.* □

While the notion of stopping set is well suited to the erasure channel where the pseudo-weight is defined as the support of a pseudo-codeword [5], it is not refined enough to capture the situation for the AWGN channel. Figure 6 shows two Tanner graphs that only allow the all-zero word as valid codeword. Both graphs admit a pseudo-codeword $\boldsymbol{\omega} = (2/3, 2/3, 2/3, 2/3)$ in the corresponding fundamental cones that has an interpretation as stopping set. However, in addition to this pseudo-codeword, the fundamental cone $\mathcal{F}(\mathsf{G})$ of one of the two graphs contains a pseudo-codeword of pseudo-weight only three.

Near-Codewords MacKay and Postol [10] introduced the notion of near-codewords. These are vectors $\mathbf{x}$ with $x_i = 0$ or $x_i = 1$ for all $1 \leqslant i \leqslant n$ such that the syndrome $\mathbf{s} = \mathbf{x}\mathbf{H}^\mathsf{T}$ has low Hamming weight. Especially interesting are the low-weight near-codewords.

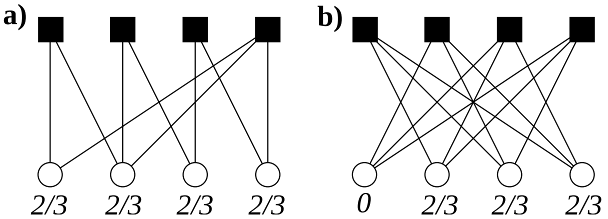

Fig. 6 Two stopping sets of size four. Pseudo-Codewords $\boldsymbol{\omega}$ are indicated that achieve different minimum pseudo-weight on an AWGN.

While the notion of near codewords is helpful in understanding potential problems in the design of iteratively decodable codes it suffers from being quantifiable in a precise sense. For example, a single one in a (j, k)-regular code may be considered as a near codeword with syndrome weight j. In order to make a precise statement on how problematic this near codeword is, one can find a corner in the fundamental cone that is close to the vector containing a single one. Note that any near codeword can be completed into a pseudo-codeword with a procedure similar to the canonical completion (now rooted at the near codeword). This gives a precise measure of the effect of a near codeword.

7. Algorithmic Issues

Theorem 5 gives a crisp characterization of the minimal pseudo-codewords, i.e. the set of pseudo-codewords that determine the shape of the region $\mathscr{D}_0$. In this section we investigate algorithmic issues to find pseudo-codewords of small pseudo-weight. In this context it is interesting to note that the fundamental cone is readily represented in the original Tanner graph by re-interpreting the function nodes and the variable nodes. To this end let a matrix $\mathbf{P}_\delta$ be defined as the $\binom{\delta}{2} \times \delta$ matrix contining all binary weight two vectors. For a real valued vector of length δ, let an indicator function $I_\delta(\boldsymbol{\omega})$ be defined as

$$I_\delta(\mathbf{z}) = \begin{cases} 1 & \exists \mathbf{x} \in \mathbb{R}^\delta : \mathbf{z} = \mathbf{x}\mathbf{P}_\delta, x_i \geqslant 0 \\ 0 & \text{otherwise.} \end{cases}$$

Membership in the fundamental cone $\mathscr{F}(\mathsf{G})$ can thus be tested by checking the indicator function

$$I_\mathsf{G}(\boldsymbol{\omega}) = \prod_{i=0}^{l} I_{\delta(f_i)}(\boldsymbol{\omega}_{\Gamma(f_i)}).$$

The factor graph [2] that is obtained by assigning the indicator functions $I_{\delta(f_i)}(\boldsymbol{\omega}_{\Gamma(f_i)})$ to the individual function nodes f_i and by letting the variable alphabets be $\mathbb{R}$ gives, in fact, a suitable framework for an iterative algororithm to find pseudo-codewords. While there is some conceptual appeal to this approach it is essentially similar to a gradient descent algorithm.

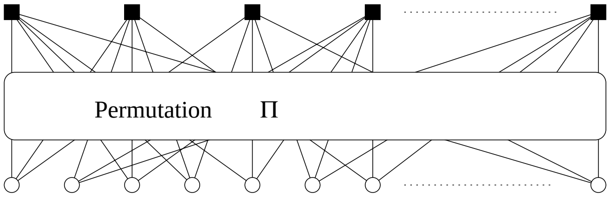

Fig. 7 The permutation in an LDPC code

In the sequel we describe a linear programming approach to finding pseudo-codewords of small pseudo-weight. For simplicity we restrict the subsequent description to (j,k)-regular LDPC codes. The generalization to irregular codes is straightforward.

LDPC codes may be described by a permutation that maps edges in $\mathsf{G} = (V,E)$ which are incident with variable nodes to edges that are incident to function nodes (see Fig. 7). Let $\mathbf{\Pi}$ be the corresponding $|E| \times |E|$ permutation matrix. Let a $(j-1) \times j$ matrix $\mathbf{F}_j$ be defined as $\mathbf{F}_j = [-\mathbf{1} : \mathbf{I}_{j-1}]$ where $\mathbf{I}_{j-1}$ is a $j-1 \times j-1$ identity matrix. $\mathbf{F}_1$ is defined as the empty matrix.

Let $\mathbf{A}$ be a $m \times n$ matrix and let the Kronecker product of two matrices $\mathbf{A}, \mathbf{B}$ be defined as

$$\mathbf{A} \circ \mathbf{B} = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & & a_{2,n}B \\ \vdots & & & \vdots \\ a_{m,1}B & a_{m,2}B & \dots & a_{m,n}B \end{pmatrix}.$$

We have the following proposition characterizing the fundamantal cone:

Proposition 8 *Let a length n, (j,k)-regular LDPC code be given with associated graph G and permutation matrix $\mathbf{\Pi}$. Let matrices $\mathbf{W}, \mathbf{Z}$ be defined as $\mathbf{W} = \mathbf{I}_{\frac{nj}{k}} \circ \mathbf{P}_k$ and $\mathbf{Z} = \mathbf{W\Pi}(\mathbf{I}_{\frac{nk}{j}} \circ \mathbf{F}_j)$. Moreover, let for a given vector $\mathbf{x}$, $\mathbf{x}_{:j}$ denote the sub-sampled vector $(x_0, x_j, x_{2j}, \ldots)$*

The fundamental cone may be described as

$$\mathscr{F}(\mathsf{G}) = \{\boldsymbol{\omega} \in \mathbb{R}^n : \boldsymbol{\omega} = (\mathbf{xW\Pi})_{:j}, \mathbf{xZ} = \mathbf{0}, x_i \geqslant 0\}$$

□

While the description of the fundamental cone in Proposition 8 seems cumbersome at first, it is well suited to formulate a linear program to find pseudo-codewords of small pseudo-weight:

Linear Program: Given $\mathbf{v}$ and the graph G
Minimize $\langle \mathbf{v}, (\mathbf{xW\Pi})_{:j} \rangle$
Subject to: $\mathbf{xZ} = \mathbf{0}$, $\langle \mathbf{x}, \mathbf{1} \rangle = 1$, $x_i \geqslant 0$.

The above linear program can be used to check a given graph G for pseudo-codewords in the set $\mathcal{U}(\mathsf{G})$. For a random choice of the vector $\mathbf{v}$ we will typically get a pseudo-codeword in $\mathcal{U}(\mathsf{G})$ of relatively high weight. However, choosing a vector $\mathbf{v}$ which contains a single one in a position and is zero otherwise will yield pseudo-codewords of smaller weight. The same is true in general if the support of $\mathbf{v}$ is chosen according to a near-codeword.

REFERENCES

[1] R. M. Tanner, “A recursive approach to low-complexity codes,” *IEEE Trans. on Inform. Theory*, vol. IT–27, pp. 533–547, Sept. 1981.

[2] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. on Inform. Theory*, vol. IT–47, no. 2, pp. 498–519, 2001.

[3] N. Wiberg, *Codes and Decoding on General Graphs.* PhD thesis, Linköping University, Sweden, 1996.

[4] N. Wiberg, H.-A. Loeliger, R. Kötter, “Codes and Iterative Decoding on General Graphs”, *European Transactions on Telecommunications,* 6(5), pp. 513-525, September 1995.

[5] 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 *Codes, systems, and graphical models (Minneapolis, MN, 1999)*, vol. 123 of *IMA Vol. Math. Appl.*, pp. 101–112, New York: Springer, 2001.

[6] B. J. Frey, R. Koetter, and A. Vardy, “Signal-space characterization of iterative decoding,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 766–781, 2001.

[7] R.G. Gallager, *Low-Density Parity-Check Codes.* M.I.T. Press, Cambridge, MA, 1963, available online under `http://justice.mit.edu/people/gallager.html`.

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

[9] S. M. Aji and R.J. McEliece The Generalized Distributive Law, *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 325–343, March 2000.

[10] D. J. C. MacKay and M. S. Postol, “Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes,” *preprint*, 2002.

[11] R. M. Tanner, D. Sridhara, and T. Fuja, “A class of group-structured LDPC codes,” Proc. of ICSTA 2001, Ambleside, England, 2001.

[12] R. Koetter, P.O. Vontobel, “Graph-covers and iterative decoding of finite length codes,” in preparation, 2003