Pseudo-Codeword Analysis of Tanner Graphs From Projective and Euclidean Planes

Roxana Smarandache, *Member, IEEE*, and Pascal O. Vontobel, *Member, IEEE*

***Abstract*—We consider coded data transmission over a binary-input output-symmetric memoryless channel using a binary linear code. In order to understand the performance of maximum-likelihood (ML) decoding, one studies the codewords, in particular the minimal codewords, and their Hamming weights. In the context of linear programming (LP) decoding, one's attention needs to be shifted to the pseudo-codewords, in particular, to the minimal pseudo-codewords and their pseudo-weights. In this paper, we investigate some families of codes that have good properties under LP decoding, namely certain families of low-density parity-check (LDPC) codes that are derived from projective and Euclidean planes: we study the structure of their minimal pseudo-codewords and give lower bounds on their pseudo-weight. Besides this main focus, we also present some results that hold for pseudo-codewords and minimal pseudo-codewords of any Tanner graph, and we highlight how the importance of minimal pseudo-codewords under LP decoding varies depending on which binary-input output-symmetric memoryless channel is used.**

***Index Terms*—Codes from Euclidean planes, codes from projective planes, linear programming decoding, message-passing iterative decoding, minimal codewords, minimal pseudo-codewords, pseudo-weight, pseudo-weight spectra.**

I. Introduction

LINEAR programming (LP) decoding has recently emerged as an interesting approach to decoding binary linear codes [1], [2]. The present paper is one of the first papers that attempts to present a characterization of the set of pseudo-codewords and, in particular, of the set of minimal pseudo-codewords, the objects which determine the performance of an LP decoder.

Let us be more specific. For an $[n, k]$ binary linear code $\mathcal{C}$ used for data communication over a memoryless binary-input output-symmetric channel [3, Definition 1], the maximum-likelihood (ML) decoding problem can be formulated as a linear optimization problem over the convex hull $\mathrm{conv}(\mathcal{C})$ of $\mathcal{C}$ in $\mathbb{R}^n$

$$\hat{\boldsymbol{x}} \triangleq \arg \min_{\boldsymbol{x} \in \mathrm{conv}(\mathcal{C})} \sum_{i=1}^{n} x_i \lambda_i \tag{1}$$

where λ_i is the log-likelihood ratio (LLR) associated to the ith observed channel output symbol. Assuming, without loss of generality, that the all-zeros codeword was sent, we will see that understanding ML decoding is tantamount to studying the neighboring vertices of the all-zeros vertex in the polytope $\mathrm{conv}(\mathcal{C})$, or, equivalently, to analyzing the edges of the conic hull $\mathrm{conic}\big(\mathrm{conv}(\mathcal{C})\big)$ of $\mathrm{conv}(\mathcal{C})$ [4], or to analyzing the so-called *minimal codewords*.[1]

However, for most codes of interest, the description complexity of $\mathrm{conv}(\mathcal{C})$ grows exponentially in the block length. Therefore, finding the minimum in (1) with a linear programming solver is highly impractical for reasonably long codes. A standard approach in optimization theory and practice is to replace a minimization problem by a relaxed minimization problem. Here, we replace the minimization over $\mathrm{conv}(\mathcal{C})$ by a minimization over some easily describable polytope $\mathcal{P}$ which is a relaxation of $\mathrm{conv}(\mathcal{C})$

$$\hat{\boldsymbol{x}} \triangleq \arg \min_{\boldsymbol{x} \in \mathcal{P}} \sum_{i=1}^{n} x_i \lambda_i. \tag{2}$$

If $\mathcal{P}$ is strictly larger than $\mathrm{conv}(\mathcal{C})$, then the decision rule in (2) obviously represents a suboptimal decoder. A relaxation that works particularly well for low-density parity-check (LDPC) codes was presented by Feldman *et al.* [1], [2] and will be discussed later on; the resulting decoder is generally referred to as the LP decoder. In LP decoding, the neighboring vertices of the all-zeros vector in the polytope $\mathcal{P}$, or, equivalently, the edges of the conic hull $\mathrm{conic}(\mathcal{P})$ of $\mathcal{P}$, play the crucial role. The points in $\mathcal{P}$ will be called *pseudo-codewords* and the neighboring vertices of the all-zeros vertex of $\mathcal{P}$ will be called *minimal pseudo-codewords*, to naturally generalize the notion of minimal codewords.[2] Similar to the way the Hamming-weight function measures the badness of a codeword, the *pseudo-weight* function will measure the badness of a pseudo-codeword.

Manuscript received February 26, 2006; revised April 1, 2007. The work of R. Smarandache was supported in part by the National Science Foundation under Grant ITR-0205310. The work of P. O. Vontobel was supported by the National Science Foundation under Grants ATM-0296033, CCF-0514801, DOE SciDAC, and by the Office of Naval Research under Grant N00014-00-1-0966. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Adelaide, Australia, September 2005 and at the 43rd Annual Allerton Conference on Communications, Control, and Computing, Monticello, IL, September 2005.

R. Smarandache is with the Department of Mathematics and Statistics, San Diego State University, San Diego, CA 92182 USA (e-mail: rsmarand@sciences.sdsu.edu).

P. O. Vontobel was with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53706 USA, and with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: pascal.vontobel@ieee.org).

Communicated by G. Zémor, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2007.899563

[1]Note that $\mathrm{conic}\Big(\mathrm{conv}\big(\mathcal{C}\big)\Big) = \mathrm{conic}(\mathcal{C})$.

[2]Note that in the papers by Feldman [1], [2] only the vertices of $\mathcal{P}$ are called pseudo-codewords. However, here we follow the convention of [5], [6] and call all the points in $\mathcal{P}$ pseudo-codewords.

The main focus of the present paper is the derivation of certain results on the structure of minimal pseudo-codewords and on their pseudo-weight. On the one hand, such results can—as mentioned above—be used to bound the performance of the LP decoder. On the other hand, the connection made by Koetter and Vontobel [5]–[7] between message-passing iterative (MPI) decoding and LP decoding suggests that results for LP decoding have immediate implications for MPI decoding.

Apart from presenting some general results on the set of minimal pseudo-codewords of an arbitrary binary linear code, we will mainly focus on certain families of codes based on projective and Euclidean planes. One of the reasons for this choice is that, in the past, several groups of authors have experimentally observed that such codes can perform very well under MPI decoding, see, e.g., [8], [9], and therefore these families of codes are a worthwhile object of study for making some first steps toward a rigorous understanding of the observed behavior. Another reason is that these families of codes have concise descriptions and large automorphism groups which may potentially be used to simplify their analysis.

The rest of this paper is structured as follows. In Section II, we introduce the two main families of codes: the first family contains codes that are derived from finite projective geometries and the second contains codes that are derived from finite Euclidean geometries. In Section III, we discuss some of the details of LP decoding. In particular, we will develop on the type of relaxation of $\mathrm{conv}(\mathcal{C})$ that we are using. In Sections IV and V, we explain in detail the importance of minimal codewords and minimal pseudo-codewords in the understanding of the performance of ML and LP decoding, respectively. We introduce a variety of enumerators/spectra that capture important features of the set of minimal codewords and the set of minimal pseudo-codewords, respectively. An important notion that is introduced in this section is that of the *pseudo-weight spectrum gap*. Roughly speaking, this gap tells where the lowest weight term that is not caused by a codeword appears in the pseudo-weight spectrum. If no pseudo-codewords with pseudo-weight smaller than or equal to the minimum Hamming weight of the code exist, then this gap is positive, otherwise, it is nonpositive. For most families of codes, and in particular, for essentially all randomly generated LDPC codes, this gap is negative. However, codes in the code family under investigation have a positive additive white Gaussian noise channel (AWGNC) pseudo-weight spectrum gap. One of the consequences of a positive gap is that asymptotically (i.e., when the signal-to-noise ratio (SNR) goes to infinity) the LP decoder performance achieves the ML decoder performance. In Section VI, we exemplify these concepts by presenting the pseudo-weight spectra and the decoding performance of certain selected codes. Section VII analyzes the possible pseudo-weights of pseudo-codewords: first, it presents a lower bound on the pseudo-weight of any pseudo-codeword in an arbitrary binary linear code; second, it presents lower bounds on the pseudo-weight of pseudo-codewords of projective-geometry-based codes. In Section VIII, we study the structure of minimal pseudo-codewords in projective-geometry-based codes and how minimum pseudo-codewords with large pseudo-weight relate to minimum pseudo-codewords with small pseudo-weight. In Section IX, we talk about an interesting feature of the binary-symmetric channel (BSC) and of the binary erasure channel (BEC) and discuss its implications for minimal pseudo-codewords: because the set of possible log-likelihood vectors is rather limited for these two channels, it can happen that there are minimal pseudo-codewords that will never be the solution of the LP decoder. We express this fact by saying that a minimal pseudo-codeword is *not effective* and we study which minimal pseudo-codewords are effective and which ones are not. Finally, Section X offers some conclusions. In the Appendix, we gathered the lengthy proofs of this paper in order to allow a better flow of the presented material.

A. Notation

The symbols $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ will denote the set of real numbers, the set of nonnegative real numbers, and the set of positive real numbers, respectively. Moreover, the *support* of a vector $\boldsymbol{x}$ will be defined as $\mathrm{supp}(\boldsymbol{x}) \triangleq \{i \mid x_i \neq 0\}$, the *Hamming weight* of a vector will be as usual $w_{\mathrm{H}}(\boldsymbol{x}) \triangleq |\mathrm{supp}(\boldsymbol{x})|$, and $\langle \boldsymbol{a}, \boldsymbol{b} \rangle \triangleq \sum_i a_i b_i$ will denote the standard inner product of two vectors of equal length. Finally, we define $0 \cdot \infty$ to be equal to 0. (This convention is used when computing inner products of pseudo-codewords and LLR vectors.)

II. The Families of Codes Under Investigation

The codes under investigation come from the families of codes that were called type-I PG-LDPC and type-I EG-LDPC codes in [9]. Type-I PG-LDPC codes are defined as follows. Let $q \triangleq 2^s$ for some positive integer s and consider a (finite) projective plane $\mathrm{PG}(2,q)$ (see, e.g., [10], [11]) with q^2+q+1 points and q^2+q+1 lines: each point lies on $q+1$ lines and each line contains $q+1$ points. A standard way of associating a parity-check matrix $\boldsymbol{H}$ of a binary linear code to a finite geometry is to let the set of points correspond to the columns of $\boldsymbol{H}$, to let the set of lines correspond to the rows of $\boldsymbol{H}$, and finally to define the entries of $\boldsymbol{H}$ according to the incidence structure of the finite geometry. In this way, we can associate to the projective plane $\mathrm{PG}(2,q)$ the code $\mathcal{C}_{\mathrm{PG}(2,q)}$ with parity-check matrix $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$, whose parameters are

length	$n = q^2+q+1$,
dimension	$k = n - 3^s - 1$,
minimum Hamming distance	$d_{\min} = q+2$,
uniform column weight of $\boldsymbol{H}$	$w_{\mathrm{col}} = q+1$,
uniform row weight of $\boldsymbol{H}$	$w_{\mathrm{row}} = q+1$,
size of $\boldsymbol{H}$	$n \times n$.

In the usual way [12], we associate a Tanner graph $\mathsf{T}(\boldsymbol{H}_{\mathrm{PG}(2,q)})$ to the parity-check matrix $\boldsymbol{H}_{\mathrm{PG}(2,q)}$: this graph consists of $n = q^2+q+1$ variable nodes of degree $w_{\mathrm{col}} = q+1$ and of $n = q^2+q+1$ check nodes of degree $w_{\mathrm{row}} = q+1$.

Type-I EG-LDPC codes are defined as follows. Let $q \triangleq 2^s$ for some positive integer s and consider a (finite) Euclidean plane $\mathrm{EG}(2,q)$ (see, e.g., [10], [11]) with q^2 points and q^2+q lines: each point lies on $q+1$ lines and each line contains q points. We essentially use the same procedure as outlined above in order to associate a parity-check matrix to this finite geometry. But before doing this, we modify the Euclidean plane slightly: we select a point of $\mathrm{EG}(2,q)$ and remove it together with the $q+1$ lines through it. Doing so, we obtain an $\mathrm{EG}(2,q)$-based code

$\mathcal{C}_{\mathrm{EG}(2,q)}$, with parity-check matrix $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{EG}(2,q)}$, whose parameters are

length	$n = q^2 - 1$,
dimension	$k = n - 3^s + 1$,
minimum Hamming distance	$d_{\min} = q + 1$,
uniform column weight of $\boldsymbol{H}$	$w_{\mathrm{col}} = q$,
uniform row weight of $\boldsymbol{H}$	$w_{\mathrm{row}} = q$,
size of $\boldsymbol{H}$	$n \times n$.

Again, we can associate a Tanner graph $\mathsf{T}(\boldsymbol{H}_{\mathrm{EG}(2,q)})$, [12], to the parity-check matrix $\boldsymbol{H}_{\mathrm{EG}(2,q)}$: this graph consists of $n = q^2 - 1$ variable nodes of degree $w_{\mathrm{col}} = q$ and of $n = q^2 - 1$ check nodes of degree $w_{\mathrm{row}} = q$.

Both families of codes have the nice property that, with an appropriate ordering of the columns and rows, the parity-check matrices are circulant matrices, meaning that $\mathcal{C}_{\mathrm{PG}(2,q)}$ and $\mathcal{C}_{\mathrm{EG}(2,q)}$ are cyclic codes. This fact can, e.g., be used for efficient encoding. Such symmetries can also substantially simplify the analysis. Note that the automorphisms groups of $\mathcal{C}_{\mathrm{PG}(2,q)}$ and $\mathcal{C}_{\mathrm{EG}(2,q)}$ contain many more automorphisms besides the cyclic-shift automorphism implied by the cyclicity of the codes.

Because the relaxed polytope proposed by Feldman *et al.* is a function of the Tanner graph, it follows—when analyzing LP decoding—that the automorphism group of the Tanner graph representing a code is actually more relevant than the automorphism group of the code.[3] For the above codes, this means that the relevant automorphisms are the automorphisms of the projective plane and of the modified (modified as explained above) Euclidean plane, respectively. Both automorphism groups are one-transitive, i.e., for any two points there exists an automorphism that maps the first point onto the second point. Moreover, later in the paper we will also use the two-transitivity of the automorphism group of the projective plane, i.e., for every two pairs of points there exists an automorphism mapping the first pair onto the second pair.

III. ML and LP Decoding

In this section, we briefly review ML and LP decoding [1], [2]. Consider a binary linear code $\mathcal{C} \subseteq \{0,1\}^n$ of length n and dimension k that is used for data communication over a memoryless binary-input channel with channel law $p_{Y|X}(y|x)$. The transmitted codeword will be called $\boldsymbol{x} \triangleq (x_1, \ldots, x_n)$ and the received vector will be called $\boldsymbol{y} \triangleq (y_1, \ldots, y_n)$. Based on the received vector, we can define the LLR vector $\boldsymbol{\lambda} \triangleq (\lambda_1, \ldots, \lambda_n) \in (\mathbb{R} \cup \{\pm\infty\})^n$ to be the vector containing the LLRs $\lambda_i \triangleq \log(p_{Y|X}(y_i|0)/p_{Y|X}(y_i|1))$, $i = 1, \ldots, n$. Using the canonical embedding of the set $\{0,1\}$ into $\mathbb{R}$ and of the set $\mathcal{C}$ into $\mathbb{R}^n$, ML decoding can then be cast as

$$\hat{\boldsymbol{x}} \triangleq \arg\min_{\boldsymbol{x} \in \mathcal{C}} \langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle. \tag{3}$$

Letting $\mathrm{conv}(\mathcal{C})$ be the convex hull of $\mathcal{C}$ in $\mathbb{R}^n$, the above ML decoding rule can also be formulated as

$$\hat{\boldsymbol{x}} \triangleq \arg \min_{\boldsymbol{x} \in \mathrm{conv}(\mathcal{C})} \langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle. \tag{4}$$

Unfortunately, for most codes of interest, the description complexity of $\mathrm{conv}(\mathcal{C})$ grows exponentially in the block length and therefore finding the minimum in (4) with a linear programming solver is highly impractical for reasonably long codes.[4]

A standard approach in optimization theory and practice is to replace a minimization problem by a relaxed minimization problem, in our case we replace the minimization over $\mathrm{conv}(\mathcal{C})$ by a minimization over some easily describable polytope $\mathcal{P}$ which is a relaxation of $\mathrm{conv}(\mathcal{C})$

$$\hat{\boldsymbol{x}} \triangleq \arg\min_{\boldsymbol{x} \in \mathcal{P}} \langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle. \tag{5}$$

If $\mathcal{P}$ is strictly larger than $\mathrm{conv}(\mathcal{C})$, then the decision rule in (5) obviously represents a suboptimal decoder. A relaxation that works particularly well for LDPC codes is given by the following approach [1], [2]. Let $\mathcal{C}$ be described by an $m \times n$ parity-check matrix $\boldsymbol{H}$ with rows $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_m$. Then the polytope $\mathcal{P} \triangleq \mathcal{P}(\boldsymbol{H})$, in this context also called the *fundamental polytope* [5], [6], is defined as

$$\mathcal{P} \triangleq \bigcap_{i=1}^{m} \mathrm{conv}(\mathcal{C}_i) \quad \text{with}$$
$$\mathcal{C}_i \triangleq \left\{ \boldsymbol{x} \in \{0,1\}^n \mid \langle \boldsymbol{h}_i, \boldsymbol{x} \rangle = 0 \bmod 2 \right\}.$$

Note that $\mathcal{P}$ is a convex set within $[0,1]^n$ that contains $\mathrm{conv}(\mathcal{C})$, but whose description complexity is typically much smaller than the one of $\mathrm{conv}(\mathcal{C})$. Points in the set $\mathcal{P}$ will be called *pseudo-codewords*. Because the set $\mathcal{P}$ is usually strictly larger than $\mathrm{conv}(\mathcal{C})$, it can obviously happen that the decoding rule in (5) delivers a vertex of $\mathcal{P}$ that is not a codeword. Such vertices that correspond to pseudo-codewords that are not codewords are the reason for the suboptimality of LP decoding (cf. [1], [2], [5], [6]). Note that $\mathcal{P} = \mathcal{P}(\boldsymbol{H})$ is a function of the parity-check matrix $\boldsymbol{H}$ that describes the code $\mathcal{C}$. Different parity-check matrices for the same code might therefore lead to different fundamental polytopes. It is worthwhile to remark though that all these fundamental polytopes have the property that $\mathcal{P}(\boldsymbol{H}) \cap \{0,1\}^n = \mathcal{C}$, i.e., all points of $\mathcal{P}(\boldsymbol{H})$ with integral coordinates are also codewords [1], [2].

IV. Minimal Codewords

In this section, we will introduce minimal codewords and explain their importance with respect to ML decoding. Although ML decoding is often impractical, knowing bounds on the block-error rate of the ML decoder can help in assessing the performance of suboptimal but practical decoding algorithms.

Definition 1: Let $\mathcal{C}$ be a binary linear code. A nonzero codeword $\boldsymbol{x} \in \mathcal{C}$ is called *minimal* if and only if its support does not (strictly) contain the support of any other nonzero codeword as a proper subset.[5] The set of all minimal codewords of $\mathcal{C}$ is commonly denoted by $\mathcal{M}(\mathcal{C})$. □

We will henceforth assume that we transmit a binary linear code $\mathcal{C}$ over a binary-input output-symmetric channel. For this setup, when studying the ML decoder in (3) or (4), we can

[3] It is clear that the automorphism group of the Tanner graph of a code is always a subgroup of the automorphism group of the code.

[4] Exceptions to this observation include, for example, the class of convolutional codes with not too many states.

[5] It can easily be seen that this definition of minimal codeword equals the "geometric" definition, i.e., in terms of $\mathrm{conv}(\mathcal{C})$ and $\mathrm{conic}(\mathcal{C})$, that we gave in Section I.

without loss of generality assume that the all-zero codeword was sent, because all decision regions are congruent. The importance of minimal codewords lies in the following considerations.

Theorem 2 (cf., e.g.,[13]): Let $\mathcal{C}$ be a binary linear code of length n and for $\boldsymbol{x} \in \mathcal{C}$ let

$$\mathcal{D}_{\boldsymbol{x}}^{\mathrm{ML}} \triangleq \left\{ \boldsymbol{\lambda} \in \mathbb{R}^n \mid \langle \boldsymbol{x}', \boldsymbol{\lambda} \rangle \geqslant \langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle \text{ for all } \boldsymbol{x}' \in \mathcal{C} \setminus \{\boldsymbol{x}\} \right\}$$

be the region in the LLR space where the ML decoder decides in favor of the codeword $\boldsymbol{x}$.[6] Then the decision region $\mathcal{D}_{\boldsymbol{x}}^{\mathrm{ML}}$ of a codeword $\boldsymbol{x} \in \mathcal{C}$ shares a facet[7] with the decision region $\mathcal{D}_{\boldsymbol{0}}^{\mathrm{ML}}$ of the zero codeword if and only if $\boldsymbol{x}$ is a minimal codeword.

Therefore, knowing the minimal codewords of the code $\mathcal{C}$ is sufficient in order to assess its ML decoding performance.

We remark that in the context of linear codes, Hwang [15] was the first to consider the set of minimal codewords of a code (there called the "projecting set of a code"). He studied them in connection with two modifications of the correlation decoding algorithm.[8] Minimal codewords and their properties arise also in connection with secret sharing. As it was first pointed out in [17] and further pursued in [18], [19], minimal vectors in a linear code completely specify the access structure of the linear secret sharing scheme defined by the code. We finally note that minimal vectors were also studied in combinatorics under the concept of cycles of linear matroids.

V. The Fundamental Cone, Minimal Pseudo-Codewords, and Spectra

In this section, we will shift our attention to LP decoding and the objects of interest: pseudo-codewords and, in particular, minimal pseudo-codewords. For analyzing LP decoding of a binary linear code that is used for data transmission over a binary-input output-symmetric channel, it is sufficient to consider the part of the fundamental polytope $\mathcal{P}$ around the vertex $\boldsymbol{0}$, cf. [5], [6], i.e., the fundamental cone. (See also [1], [2] that discuss this so-called "$\mathcal{C}$-symmetry" property.)

Lemma 3 ([1], [2], [6]): Let $\mathcal{C}$ be an arbitrary binary linear code and let $\boldsymbol{H}$ be its parity-check matrix. We let $\mathcal{J} \triangleq \mathcal{J}(\boldsymbol{H})$ be the set of row indices of $\boldsymbol{H}$ and we let $\mathcal{I} \triangleq \mathcal{I}(\boldsymbol{H})$ be the set of column indices of $\boldsymbol{H}$, respectively. For each $j \in \mathcal{J}$, we let $\mathcal{I}_j \triangleq \mathcal{I}_j(\boldsymbol{H}) \triangleq \{i \in \mathcal{I} \mid h_{ji} = 1\}$, and for each $i \in \mathcal{I}$, we let $\mathcal{J}_i \triangleq \mathcal{J}_i(\boldsymbol{H}) \triangleq \{j \in \mathcal{J} | h_{ij} = 1\}$. Let the fundamental cone $\mathcal{K}(\boldsymbol{H})$ of $\boldsymbol{H}$ be the conic hull of the fundamental polytope $\mathcal{P}(\boldsymbol{H})$. Then, $\mathcal{K}(\boldsymbol{H})$ is the set of vectors $\boldsymbol{\omega} \in \mathbb{R}^n$ that satisfy

$$\forall\, j \in \mathcal{J},\ \forall\, i \in \mathcal{I}_j : \sum_{i' \in \mathcal{I}_j \setminus \{i\}} \omega_{i'} \geqslant \omega_i \tag{6}$$

$$\forall\, i \in \mathcal{I} : \omega_i \geqslant 0. \tag{7}$$

□

We note that if $\boldsymbol{\omega}$ is in $\mathcal{K}(\boldsymbol{H})$, then also $\alpha \cdot \boldsymbol{\omega}$ is in $\mathcal{K}(\boldsymbol{H})$ for any $\alpha \in \mathbb{R}_{++}$. Moreover, for any $\boldsymbol{\omega}$ in $\mathcal{K}(\boldsymbol{H})$ there exists an $\alpha \in \mathbb{R}_{++}$ (in fact, a whole interval of α's) such that $\alpha \cdot \boldsymbol{\omega}$ is in $\mathcal{P}(\boldsymbol{H})$. Let

$$\mathcal{D}_{\boldsymbol{0}}^{\mathrm{LP}} \triangleq \left\{ \boldsymbol{\lambda} \in \mathbb{R}^n \mid \langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \geqslant 0 \text{ for all } \boldsymbol{\omega} \in \mathcal{P}(\boldsymbol{H}) \setminus \{\boldsymbol{0}\} \right\}$$

be the region where the LP decoder decides in favor of the codeword $\boldsymbol{0}$.[9] It can easily been seen that

$$\mathcal{D}_{\boldsymbol{0}}^{\mathrm{LP}} = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^n \mid \langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \geqslant 0 \text{ for all } \boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H}) \right\}.$$

Therefore, when studying LP decoding it is enough to know $\mathcal{K}(\boldsymbol{H})$; all vectors $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ will henceforth be called pseudo-codewords. Moreover, two pseudo-codewords where one is a positive multiple of the other will be considered to be equivalent.

A class of pseudo-codewords that will be used a few times later on is the class of so-called unscaled pseudo-codewords [20], [6]. These pseudo-codewords have integer entries and are derived from codewords in finite covers of the Tanner graph $\mathsf{T}(\boldsymbol{H})$. An important property of an unscaled pseudo-codeword $\boldsymbol{\omega}$ is that $\boldsymbol{\omega} \pmod 2 \in \mathcal{C}$.[10]

Another important class of pseudo-codewords is the class of so-called minimal pseudo-codewords.

Definition 4 ([5], [6]): Let $\mathcal{C}$ be an arbitrary binary linear code described by the parity-check matrix $\boldsymbol{H}$ whose fundamental cone is $\mathcal{K}(\boldsymbol{H})$. A vector $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ is called a *minimal pseudo-codeword* if the set $\{\alpha \cdot \boldsymbol{\omega} \mid \alpha \in \mathbb{R}_+\}$ is an edge of $\mathcal{K}(\boldsymbol{H})$. The set of all minimal pseudo-codewords will be called $\mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))$.[11] □

For a given binary linear code $\mathcal{C}$ with parity-check matrix $\boldsymbol{H}$, the importance of the set $\mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))$ lies in the following fact. From basic cone properties (cf., e.g., [4]), it can easily be seen that $\mathcal{D}_{\boldsymbol{0}}^{\mathrm{LP}} = \{\boldsymbol{\lambda} \in \mathbb{R}^n \mid \langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \geqslant 0 \text{ for all } \boldsymbol{\omega} \in \mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))\}$. Therefore, the set $\mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))$ completely characterizes the behavior of the LP decoder. It can be shown [6] that for any minimal pseudo-codeword $\boldsymbol{\omega}$ there is an $\alpha \in \mathbb{R}_{++}$ such that $\alpha \cdot \boldsymbol{\omega}$ is an unscaled pseudo-codeword, which, among other things, implies that all components of $\alpha \cdot \boldsymbol{\omega}$ are integers.

Note that the above notion of minimal pseudo-codewords generalizes the notion of minimal codewords in the following sense: whereas minimal pseudo-codewords correspond one-to-one to the edges of $\mathcal{P}(\boldsymbol{H})$ (or $\mathcal{K}(\boldsymbol{H})$) around $\boldsymbol{0}$, the minimal codewords correspond one-to-one to the edges of $\mathrm{conv}(\mathcal{C})$ around $\boldsymbol{0}$. (Minimal codewords are usually also minimal pseudo-codewords, but not always, as was remarked in [6].)

Because of the one-to-one relationship between parity-check matrices and Tanner graphs, the fundamental cone can also be seen as a function of the Tanner graph representing a code. Therefore, in order to emphasize the dependence of minimal pseudo-codewords on the representation of the code, we will sometimes talk about the minimal pseudo-codewords of a Tanner graph.

The fundamental cone is *independent* of the specific memoryless binary-input channel through which we are transmitting, however, the influence of a pseudo-codeword on the LP

[6]We assume that during ML decoding ties between decoding regions are resolved randomly.

[7]A facet is an $n-1$-dimensional face of a polytope, see, e.g., [14].

[8]In the light of the ML decoder as formulated in (4), these algorithms can be seen as variations of the simplex method (cf., e.g., [16]) that minimizes $\langle \boldsymbol{x}, \boldsymbol{\lambda} \rangle$ over the polytope $\mathrm{conv}(\mathcal{C})$.

[9]We assume that during LP decoding ties between decoding regions are resolved randomly.

[10]See [20], [6] for an exact definition of unscaled pseudo-codewords and their properties.

[11]Note that this definition implies that $\boldsymbol{0} \notin \mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))$.

decoding performance depends very much on what channel is used. This influence will be measured by a channel-dependent *pseudo-weight* of pseudo-codewords; these pseudo-weights can be seen as generalizations of the Hamming weight that has traditionally been used to assess the performance of a code under ML decoding.

Definition 5: Let $\boldsymbol{\omega} \in \mathbb{R}_+^n$. The binary-input AWGNC pseudo-weight [21], [22], [6] of $\boldsymbol{\omega}$ is defined to be

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) \triangleq \frac{\|\boldsymbol{\omega}\|_1^2}{\|\boldsymbol{\omega}\|_2^2}$$

if $\boldsymbol{\omega} \neq \mathbf{0}$ and $w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) \triangleq 0$ otherwise, where $\|\boldsymbol{\omega}\|_1$ and $\|\boldsymbol{\omega}\|_2$ are the $\mathcal{L}_1$- and $\mathcal{L}_2$-norms of $\boldsymbol{\omega}$, respectively. Let $\boldsymbol{\omega}' \in \mathbb{R}_+^n$ be a vector with the same components as $\boldsymbol{\omega}$ but in nonincreasing order. Introducing

$$f(\xi) \triangleq \omega_i' \qquad (i-1 < \xi \leqslant i,\ 0 < \xi \leqslant n)$$
$$F(\xi) \triangleq \int_0^{\xi} f(\xi')\mathrm{d}\xi'$$
$$e \triangleq F^{-1}\left(\frac{F(n)}{2}\right)$$

the BSC pseudo-weight [22], [6] is defined to be $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}) \triangleq 2e$ if $\boldsymbol{\omega} \neq \mathbf{0}$ and $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}) \triangleq 0$ otherwise. Finally, the BEC pseudo-weight [22], [6] is defined to be

$$w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega}) = |\mathrm{supp}(\boldsymbol{\omega})|. \qquad \square$$

Note that for $\boldsymbol{x} \in \{0,1\}^n$ we have

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{x}) = w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{x}) = w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{x}) = w_{\mathrm{H}}(\boldsymbol{x}).$$

Let us briefly comment on the significance of the above pseudo-weights. When transmitting over an AWGNC, it can be shown that the squared Euclidean distance from the point $+\mathbf{1}$ in signal space, which corresponds to the codeword $\mathbf{0}$, to the plane $\{\boldsymbol{\lambda} \in \mathbb{R}^n \mid \langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle = 0\}$ is $w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega})$. When transmitting over a BSC, the LP decoder decides in favor of $\mathbf{0}$ and against $\boldsymbol{\omega}$ if the number of bit-flips on the BSC is smaller than $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega})/2$; on the other hand, there is at least one pattern with at least $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega})/2$ bit-flips such that the LP decoder decides in favor of $\boldsymbol{\omega}$ and against $\mathbf{0}$, assuming that ties are resolved randomly. Finally, when transmitting over a BEC, the LP decoder decides in favor of $\mathbf{0}$ and against $\boldsymbol{\omega}$ if the number of erasures on the BEC is smaller than $w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega})$; on the other hand, there is a pattern with $w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega})$ erasures such that the LP decoder decides in favor of $\boldsymbol{\omega}$ and against $\mathbf{0}$ (assuming that ties are resolved randomly). For a more detailed discussion, see [6], [22].

Definition 6: Let $\mathcal{C}$ be an arbitrary binary linear code. We recall the definition of the codeword weight enumerator to be the polynomial (cf. e.g., [23])

$$\chi_{\mathcal{C}}^{\mathrm{cw}}(X) \triangleq \sum_{\boldsymbol{x} \in \mathcal{C}} X^{w_{\mathrm{H}}(\boldsymbol{x})}.$$

We define the minimal codeword weight enumerator to be the polynomial

$$\chi_{\mathcal{C}}^{\mathrm{mcw}}(X) \triangleq \sum_{\boldsymbol{x} \in \mathcal{M}(\mathcal{C})} X^{w_{\mathrm{H}}(\boldsymbol{x})}.$$

Similarly, the minimal pseudo-codeword AWGNC pseudo-weight enumerator is defined to be the polynomial (with potentially noninteger exponents)

$$\chi_{\boldsymbol{H}}^{\mathrm{mpcw,AWGNC}}(X) = \sum_{[\boldsymbol{\omega}] \in \mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))} X^{w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega})}.$$

The summation in the last enumerator is over all equivalence classes of minimal pseudo-codewords.[12] (The minimal pseudo-codeword BSC pseudo-weight enumerator and the minimal pseudo-codeword BEC pseudo-weight enumerator are defined analogously.) In the case of the BEC, we additionally introduce a slightly modified minimal pseudo-codeword BEC pseudo-weight enumerator $\chi_{\boldsymbol{H}}^{\mathrm{mpcw,BEC}'}(X)$: the definition is as before except that for this enumerator two pseudo-codewords $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$ are said to be in the same equivalence class if $\mathrm{supp}(\boldsymbol{\omega}) = \mathrm{supp}(\boldsymbol{\omega}')$.[13] (See the comments about stopping sets in Section VI-F why this definition also makes sense.) $\square$

Instead of "weight enumerator" and "pseudo-weight enumerator" we will frequently use the words "weight spectrum" or "pseudo-weight spectrum," respectively, or simply "spectrum." Ideally, for a code defined by a parity-check matrix $\boldsymbol{H}$, we would like to know all the terms of the spectra that were defined in Definition 6. Often, we have to settle with less, in particular one often focuses on some quantities that characterize important aspects of a spectrum.

One such quantity is the minimum pseudo-weight: we let $w_{\mathrm{p}}^{\mathrm{AWGNC,min}}(\boldsymbol{H})$, $w_{\mathrm{p}}^{\mathrm{BSC,min}}(\boldsymbol{H})$, and $w_{\mathrm{p}}^{\mathrm{BEC,min}}(\boldsymbol{H})$ be the minimum AWGNC, BSC, and BEC pseudo-weights of a parity-check matrix $\boldsymbol{H}$, i.e., the minimum of the respective pseudo-weights, over all nonzero points in $\mathcal{K}(\boldsymbol{H})$. Applying Theorem 1 in [24] (or, alternatively, some simple tree-based techniques) we obtain $w_{\mathrm{p}}^{\mathrm{AWGNC,min}}(\boldsymbol{H}_{\mathrm{PG}(2,q)}) \geqslant q+2$, and because this lower bound matches the minimum Hamming weight, we actually know that $w_{\mathrm{p}}^{\mathrm{AWGNC,min}}(\boldsymbol{H}_{\mathrm{PG}(2,q)}) = q+2$. Similarly, one can derive that

$$w_{\mathrm{p}}^{\mathrm{BSC,min}}(\boldsymbol{H}_{\mathrm{PG}(2,q)}) = w_{\mathrm{p}}^{\mathrm{BEC,min}}(\boldsymbol{H}_{\mathrm{PG}(2,q)}) = q+2$$

and that

$$w_{\mathrm{p}}^{\mathrm{AWGNC,min}}(\boldsymbol{H}_{\mathrm{EG}(2,q)}) = w_{\mathrm{p}}^{\mathrm{BSC,min}}(\boldsymbol{H}_{\mathrm{EG}(2,q)}) = w_{\mathrm{p}}^{\mathrm{BEC,min}}(\boldsymbol{H}_{\mathrm{EG}(2,q)}) = q+1.$$

Another important quantity that characterizes pseudo-weight spectra is the pseudo-weight spectrum gap.

Definition 7: Let $\mathcal{C}$ be an arbitrary binary linear code described by the parity-check matrix $\boldsymbol{H}$ and let $\mathcal{M}_{\mathrm{p}}'(\mathcal{K}(\boldsymbol{H}))$ be the set of all minimal pseudo-codewords that are not multiples of minimal codewords. We call the real-valued quantity

$$g_{\boldsymbol{H}}^{\mathrm{AWGNC}} \triangleq \min_{\boldsymbol{\omega} \in \mathcal{M}_{\mathrm{p}}'(\mathcal{K}(\boldsymbol{H}))} w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) - w_{\mathrm{H}}^{\min}(\mathcal{C}(\boldsymbol{H}))$$

the AWGNC pseudo-weight spectrum gap of $\boldsymbol{H}$. (The BSC pseudo-weight spectrum gap and the BEC pseudo-weight spectrum gap are defined analogously.) $\square$

[12]Two pseudo-codewords $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \mathcal{K}(\boldsymbol{H})$ are in the same equivalence class if there exists an $\alpha \in \mathbb{R}_{++}$ such that $\boldsymbol{\omega} = \alpha \cdot \boldsymbol{\omega}'$.

[13]Clearly, two pseudo-codewords $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$ that satisfy $\boldsymbol{\omega} = \alpha \cdot \boldsymbol{\omega}'$ for some $\alpha \in \mathbb{R}_{++}$ also satisfy $\mathrm{supp}(\boldsymbol{\omega}) = \mathrm{supp}(\boldsymbol{\omega}')$. However, the reverse is not necessarily true.

Using [5, Corollary 8] (see also [6, Sec. 7]), one can show that for a randomly constructed $(w_{\mathrm{col}}, w_{\mathrm{row}})$-regular code with $3 \leqslant w_{\mathrm{col}} < w_{\mathrm{row}}$ the AWGNC pseudo-weight spectrum gap becomes *strictly negative* with probability one as the block length goes to infinity. However, for the $\mathrm{PG}(2,q)$- and $\mathrm{EG}(2,q)$-based codes there is the following result.

Theorem 8: Let $q = 2^s$ for some positive integer s. The AWGNC pseudo-weight spectrum gaps $g^{\mathrm{AWGNC}}_{\boldsymbol{H}_{\mathrm{PG}(2,q)}}$ and $g^{\mathrm{AWGNC}}_{\boldsymbol{H}_{\mathrm{EG}(2,q)}}$ are positive.

Proof: One can prove this by using results that were presented in [25]: indeed, for the $\mathrm{PG}(2,q)$-based codes, our theorem is a consequence of [25, Theorem 1] with $(\gamma, g) = (q+1, 6)$ or of [25, Theorem 3] with $(\gamma, \lambda) = (q+1, 1)$, and for the $\mathrm{EG}(2,q)$-based codes, our theorem is a consequence of [25, Theorem 1] with $(\gamma, g) = (q, 6)$ or of [25, Theorem 3] with $(\gamma, \lambda) = (q, 1)$. Alternatively, this result can also be obtained from [24, Theorem 1], namely, by studying the conditions under which the lower bound in that theorem is exact/not exact. □

In fact, we will see in Section VI that for the codes investigated in Section VI, the pseudo-weight spectrum gap is significantly positive. We note that by applying simple performance bounding techniques it can be shown that the larger the gap is, the closer is the LP decoding performance (and potentially also the MPI decoding performance [5], [6]) to the ML decoding performance as the SNR goes to infinity.[14]

VI. Examples of Spectra

In order to get a feeling for the objects that were defined in the previous sections, we think that it is valuable to explicitly list them for short codes. Therefore, this section is devoted to presenting minimal pseudo-codewords, weight enumerators, and the pseudo-weight spectrum gap for some short $\mathrm{PG}(2,q)$- and $\mathrm{EG}(2,q)$-based codes.

A. Type-I PG-LDPC Code for $q = 2$

The $\mathrm{PG}(2,2)$-based code $\mathcal{C}_{\mathrm{PG}(2,2)}$ has parameters $[n{=}7, k{=}3, d_{\min}{=}4]$ and can be represented by the following circulant parity-check matrix $\boldsymbol{H}_{\mathrm{PG}(2,2)}$ of size 7×7:

$$\boldsymbol{H}_{\mathrm{PG}(2,2)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{8}$$

It is not difficult to find out that the set $\mathcal{M}(\mathcal{C}_{\mathrm{PG}(2,2)})$ of minimal codewords contains a total of seven codewords: the codeword $(1,0,0,1,0,1,1)$ of Hamming weight 4 and all its cyclic shifts. Because the code has $2^3 = 8$ codewords in total, it turns out that all nonzero codewords are minimal codewords.[15]

[14]Of course, the pseudo-weight spectrum gap is only a first approximation to how quickly the LP decoding performance attains the ML decoding performance as the SNR goes to infinity. A better approximation is given by initial parts (or the whole) minimal pseudo-codeword pseudo-weight enumerator.

[15]On the side we note that so-called intersecting codes [26] have the property that all nonzero codewords are minimal codewords.

The set $\mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}_{\mathrm{PG}(2,2)}))$ of minimal pseudo-codewords has 14 elements (we list one representative per equivalence class): all the cyclic shifts of $(1,0,0,1,0,1,1)$ and all the cyclic shifts of $(1,2,2,1,2,1,1)$. In this case, we observe that $\mathcal{M}(\mathcal{C}_{\mathrm{PG}(2,2)})$ is a subset of $\mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}_{\mathrm{PG}(2,2)}))$. The weight enumerators are given by

$$\begin{aligned} \chi^{\mathrm{cw}}_{\mathcal{C}_{\mathrm{PG}(2,2)}}(X) &= X^0 + 7X^4 \\ \chi^{\mathrm{mcw}}_{\mathcal{C}_{\mathrm{PG}(2,2)}}(X) &= 7X^4 \\ \chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}}(X) &= 7X^4 + 7X^{6.25} \\ \chi^{\mathrm{mpcw,BSC}}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}}(X) &= 7X^4 + 7X^5 \\ \chi^{\mathrm{mpcw,BEC}}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}}(X) &= 7X^4 + 7X^7 \\ \chi^{\mathrm{mpcw,BEC}'}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}}(X) &= 7X^4 + X^7. \end{aligned}$$

Hence, the pseudo-weight spectrum gaps are

$$\begin{aligned} g^{\mathrm{AWGNC}}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}} &= 6.25 - 4 = 2.25 \\ g^{\mathrm{BSC}}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}} &= 5 - 4 = 1 \\ g^{\mathrm{BEC}}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}} &= g^{\mathrm{BEC}'}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}} = 7 - 4 = 3. \end{aligned}$$

The codes introduced in Section II were based on square parity-check matrices. However, the code $\mathrm{PG}(2,2)$ can also be described by a parity-check matrix of size 4×7 such as

$$\boldsymbol{H}'_{\mathrm{PG}(2,2)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{9}$$

which is the matrix formed by the first four lines of $\boldsymbol{H}_{\mathrm{PG}(2,2)}$. Because $\boldsymbol{H}'_{\mathrm{PG}(2,2)}$ contains a subset of the rows of $\boldsymbol{H}_{\mathrm{PG}(2,2)}$ it is clear that $\mathcal{K}(\boldsymbol{H}_{\mathrm{PG}(2,2)}) \subseteq \mathcal{K}(\boldsymbol{H}'_{\mathrm{PG}(2,2)})$. Moreover, note that a minimal pseudo-codeword of $\mathcal{K}(\boldsymbol{H}_{\mathrm{PG}(2,2)})$ does not need to be minimal pseudo-codeword of $\mathcal{K}(\boldsymbol{H}'_{\mathrm{PG}(2,2)})$. Indeed, the set of minimal pseudo-codewords that are not codewords turns out to be the union of the following sets (in which we show one representative per equivalence class):

$$\begin{aligned} &\{(3,2,1,1,1,0,0)\}, \{(0,1,2,1,1,3,0)\}, \\ &\{(0,1,1,1,2,0,3)\}, \{(0,1,1,1,1,0,0)\}, \\ &\{(2,1,1,1,0,0,1),\ (1,2,1,1,1,0,0),\ (0,1,2,1,1,1,0)\}, \\ &\{(0,1,1,1,0,2,1),\ (1,0,1,1,1,0,2),\ (2,1,0,1,1,1,0)\}, \\ &\{(0,1,1,1,2,0,1)\ (1,0,1,1,1,2,0),\ (0,1,0,1,1,1,2)\} \end{aligned}$$

where cyclic shifts of the same pseudo-codeword are grouped in the same set. It is interesting to see that in the case of the parity-check matrix $\boldsymbol{H}'_{\mathrm{PG}(2,2)}$ a cyclic shift of a minimal pseudo-codeword is not necessarily a (minimal) pseudo-codeword, as it was in the case of the parity-check matrix $\boldsymbol{H}_{\mathrm{PG}(2,2)}$.

It follows that

$$\begin{aligned} \chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}}(X) &= 11X^4 + 9X^{4.5} \\ \chi^{\mathrm{mpcw,BSC}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}}(X) &= 3X^3 + 17X^4 \\ \chi^{\mathrm{mpcw,BEC}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}}(X) &= 8X^4 + 12X^5 \\ \chi^{\mathrm{mpcw,BEC}'}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}}(X) &= 8X^4 + 9X^5, \end{aligned}$$

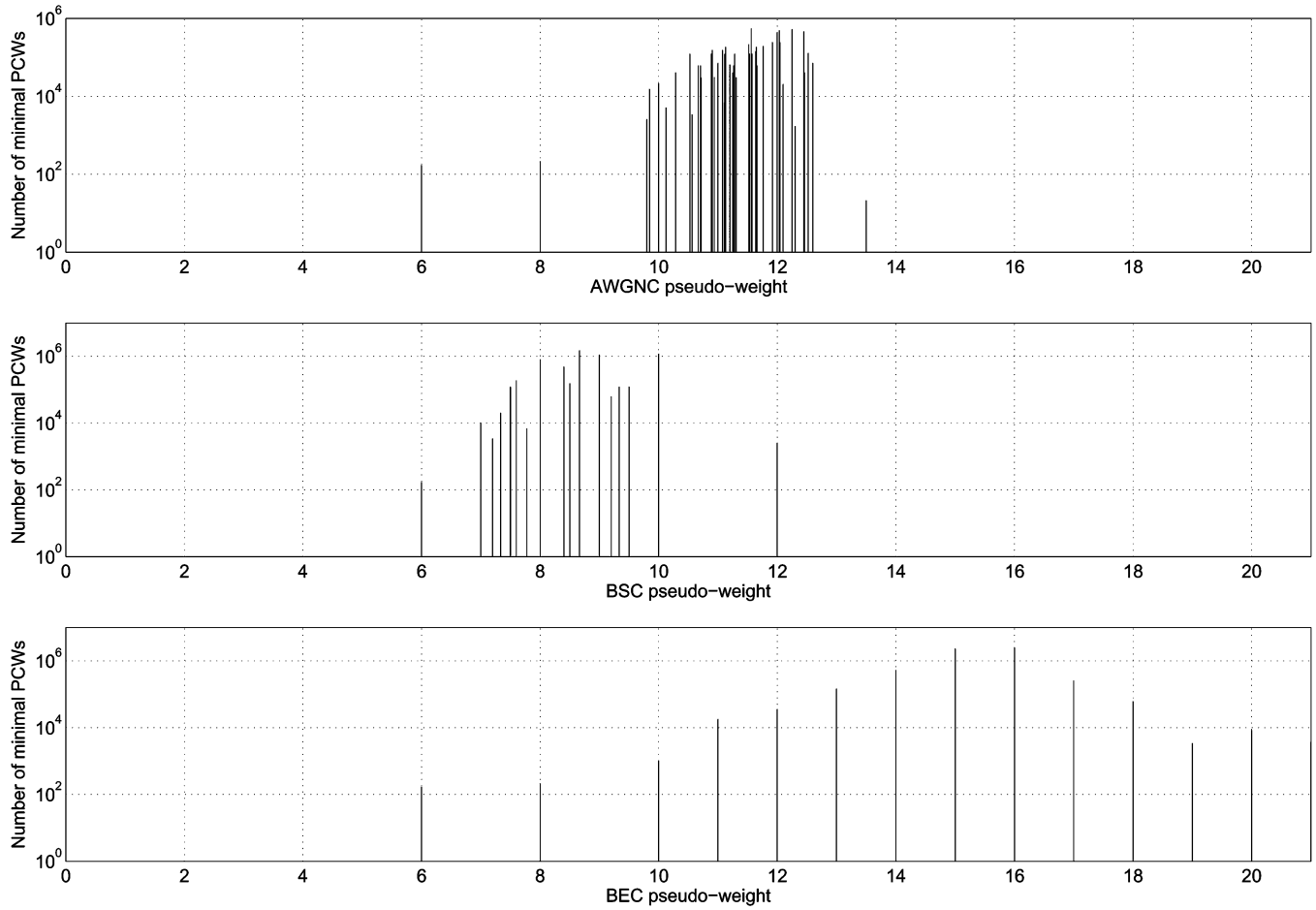

Fig. 1. Histograms of the AWGNC, BSC, and BEC pseudo-weight of minimal pseudo-codewords (PCWs) of the $\mathrm{PG}(2,4)$-based code. (Note that the y-axes are logarithmic.)

and that the pseudo-weight spectrum gaps are

$$g^{\mathrm{AWGNC}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}} = 4-4=0$$
$$g^{\mathrm{BSC}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}} = 3-4=-1$$
$$g^{\mathrm{BEC}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}} = g^{\mathrm{BEC}'_{\mathrm{PG}(2,2)}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}} = 4-4=0.$$

Comparing the enumerator $\chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}_{\mathrm{PG}(2,2)}}(X)$ with the enumerator $\chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}'_{\mathrm{PG}(2,2)}}(X)$ it is apparent that the performance of LP decoding using the second representation will be worse than the performance of LP decoding using the first representation. Based on MPI decoder simulations, MacKay and Davey [27, Sec. 4] observed a similar performance hierarchy between different representations of the same code. Note that the code under investigation in [27] was the $\mathrm{PG}(2,16)$-based code. Similar statements can be made for the BSC and the BEC.

Before concluding this subsection, let us comment on the vector $\boldsymbol{\omega} \triangleq (0,1,1,1,1,0,0)$, which is a minimal pseudo-codeword for $\boldsymbol{H}'_{\mathrm{PG}(2,2)}$ but not a codeword for $\mathcal{C}_{\mathrm{PG}(2,2)}$, even though it has only 0 and 1 components. From our remarks after Lemma 3, it follows that $\boldsymbol{\omega}$ cannot be an unscaled pseudo-codeword because $\boldsymbol{\omega} \pmod 2$ is not a codeword. However, its equivalent $(0,2,2,2,2,0,0)$ is an unscaled pseudo-codeword, and it stems from a triple cover.

Noncodeword pseudo-codewords that contain only zeros and ones will be discussed again in Theorem 13.

B. Type-I PG-LDPC Code for $q = 4$

The parity-check matrix $\boldsymbol{H}_{\mathrm{PG}(2,4)}$ of the $\mathrm{PG}(2,4)$-based code $\mathcal{C}_{\mathrm{PG}(2,4)}$ has size 21×21, uniform column and row weight 5, and yields a code with parameters $[n{=}21, k{=}11, d_{\min}{=}6]$. The codeword weight enumerator and the minimal codeword weight enumerator are

$$\chi^{\mathrm{cw}}_{\mathcal{C}_{\mathrm{PG}(2,4)}}(X) = X^0 + 168X^6 + 210X^8 + 1008X^{10} + 280X^{12} + 360X^{14} + 21X^{16}$$
$$\chi^{\mathrm{mcw}}_{\mathcal{C}_{\mathrm{PG}(2,4)}}(X) = 168X^6 + 210X^8 + 1008X^{10}$$

respectively. Looking at these enumerators we see that all codewords with Hamming weight $6, 8$, and 10 are minimal codewords. Analyzing the set of all weight-6 codewords one sees that they all have the same pattern, i.e., they can all be obtained from a single weight-6 codeword by applying suitable $\mathrm{PG}(2,4)$-automorphisms. The same is true for all other sets of codewords with the same weight. This makes the classification of all the codewords of $\mathcal{C}_{\mathrm{PG}(2,4)}$, and in particular of the minimal codewords of $\mathcal{C}_{\mathrm{PG}(2,4)}$, relatively easy.

Instead of giving the formula for $\chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}_{\mathrm{PG}(2,4)}}(X)$, $\chi^{\mathrm{mpcw,BSC}}_{\boldsymbol{H}_{\mathrm{PG}(2,4)}}(X)$, and $\chi^{\mathrm{mpcw,BEC}}_{\boldsymbol{H}_{\mathrm{PG}(2,4)}}(X)$, we simply show their histogram, cf. Fig. 1. Without going into any details, it is apparent

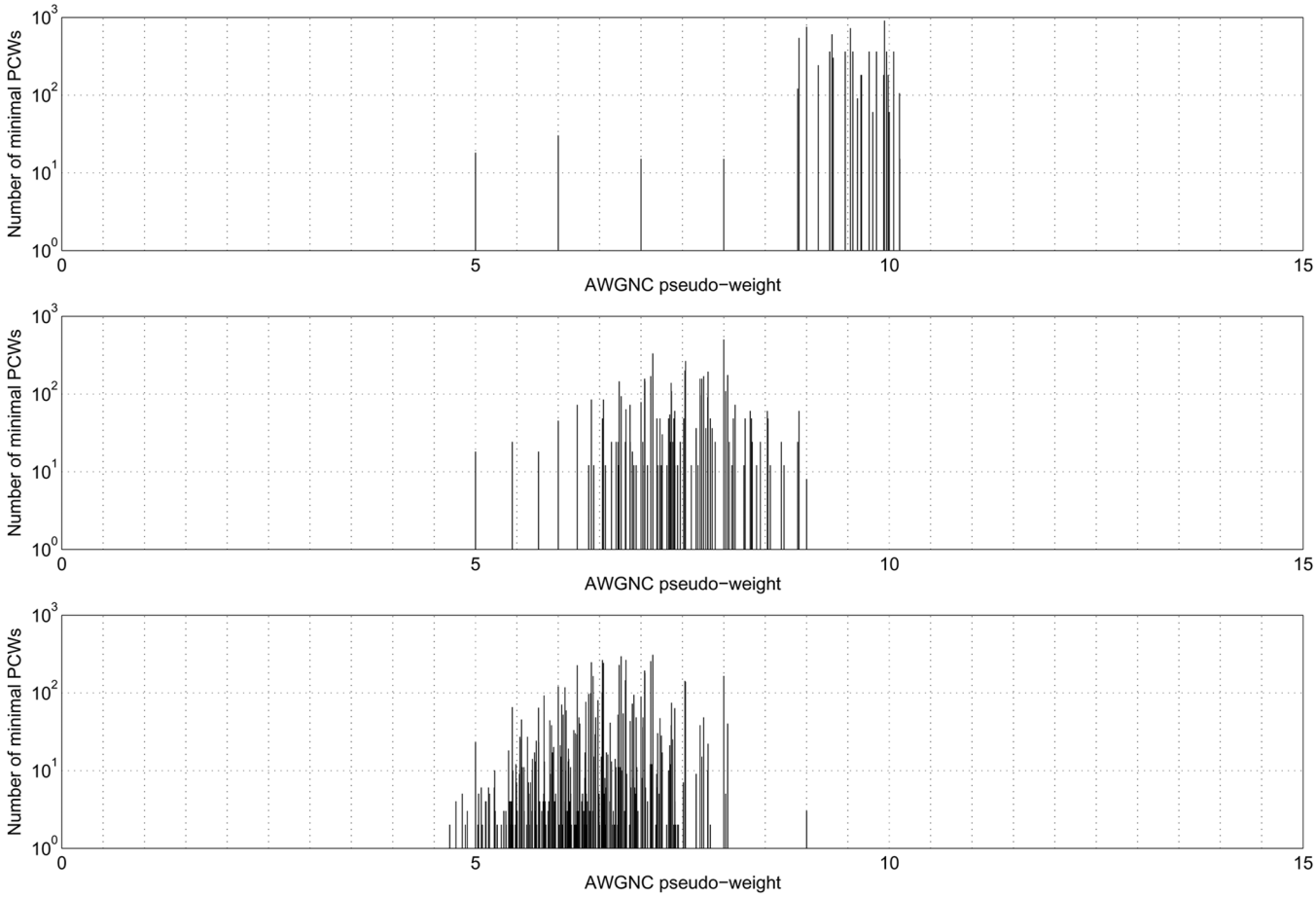

Fig. 2. Histogram of the AWGNC pseudo-weight of minimal pseudo-codewords (PCWs) of the $\mathrm{EG}(2,4)$-based code. (Note that the y-axes are logarithmic.) Top: For 15×15 parity-check matrix $\boldsymbol{H}_{\mathrm{EG}(2,4)}$, $g^{\mathrm{AWGNC}}_{\boldsymbol{H}_{\mathrm{EG}(2,4)}} = \frac{169}{19} - 5 \approx 8.89 - 5 = 3.89$. Middle: For 9×15 parity-check matrix $\boldsymbol{H}'_{\mathrm{EG}(2,4)}$, $g^{\mathrm{AWGNC}}_{\boldsymbol{H}'_{\mathrm{EG}(2,4)}} = \frac{49}{9} - 5 \approx 5.44 - 5 = 0.44$. Bottom: For 8×15 parity-check matrix $\boldsymbol{H}''_{\mathrm{EG}(2,4)}$, $g^{\mathrm{AWGNC}}_{\boldsymbol{H}''_{\mathrm{EG}(2,4)}} = \frac{361}{77} - 5 \approx 4.69 - 5 = -0.31$.

from Fig. 1 that the influence of minimal pseudo-codewords can vary depending on the channel that is used. (For related observations about varying influences of minimal pseudo-codewords, see also the discussion in [28].) The pseudo-weight gaps turn out to be

$$\begin{aligned} g^{\mathrm{AWGNC}}_{\boldsymbol{H}_{\mathrm{PG}(2,4)}} &= 9.8 - 6 = 3.8 \\ g^{\mathrm{BSC}}_{\boldsymbol{H}_{\mathrm{PG}(2,4)}} &= 7 - 6 = 1 \\ g^{\mathrm{BEC}}_{\boldsymbol{H}_{\mathrm{PG}(2,4)}} &= 11 - 6 = 5. \end{aligned}$$

We refer to the end of Section V for a discussion on the significance of positive pseudo-weight gaps.

C. Type-I PG-LDPC Code for $q = 8$

The parity-check matrix $\boldsymbol{H}_{\mathrm{PG}(2,8)}$ of the $\mathrm{PG}(2,8)$-based code $\mathcal{C}_{\mathrm{PG}(2,8)}$ has size 73×73, uniform column and row weight 9, and yields a code with parameters $[n{=}73, k{=}45, d_{\min}{=}10]$. Judging from some random search experiments in the fundamental cone $\mathcal{K}(\boldsymbol{H}_{\mathrm{PG}(2,8)})$, the AWGNC pseudo-weight spectrum gap $g^{\mathrm{AWGNC}}_{\boldsymbol{H}_{\mathrm{PG}(2,8)}}$ seems to be at least 6.0.

D. Type-I EG-LDPC Code for $q = 4$

The parity-check matrix $\boldsymbol{H}_{\mathrm{EG}(2,4)}$ of the $\mathrm{EG}(2,4)$-based code $\mathcal{C}_{\mathrm{EG}(2,4)}$ has size 15×15, uniform column and row weight 4, and yields a code with parameters $[n{=}15, k{=}7, d_{\min}{=}5]$. The codeword weight enumerator and the minimal codeword weight enumerator are

$$\begin{aligned} \chi^{\mathrm{cw}}_{\mathcal{C}_{\mathrm{EG}(2,4)}}(X) &= X^0 + 18X^5 + 30X^6 + 15X^7 \\ &\quad + 15X^8 + 30X^9 + 18X^{10} + X^{15} \\ \chi^{\mathrm{mcw}}_{\mathcal{C}_{\mathrm{EG}(2,4)}}(X) &= 18X^5 + 30X^6 + 15X^7 + 15X^8 + 30X^9 \end{aligned}$$

respectively. Looking at these enumerators we see that all codewords with Hamming weight $5, 6, 7, 8$, and 9 are minimal codewords. Analyzing the set of all weight-5 codewords one sees that they all have the same pattern, i.e., they can all be obtained from a single weight-5 codeword by applying suitable $\mathrm{EG}(2,4)$-automorphisms. The same is true for all other sets of codewords with the same weight.

The histograms in Fig. 2 correspond to various parity-check matrices that describe $\mathcal{C}_{\mathrm{EG}(2,4)}$. Fig. 2 (top) shows the histogram for $\chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}_{\mathrm{EG}(2,4)}}(X)$; Fig. 2 (middle) shows the histogram for $\chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}'_{\mathrm{EG}(2,4)}}(X)$ where $\boldsymbol{H}'_{\mathrm{EG}(2,4)}$ is a randomly selected 9×15 submatrix (with column weights at least 2) of $\boldsymbol{H}_{\mathrm{EG}(2,4)}$; and finally, Fig. 2 (bottom) shows the histogram for $\chi^{\mathrm{mpcw,AWGNC}}_{\boldsymbol{H}''_{\mathrm{EG}(2,4)}}(X)$ where $\boldsymbol{H}''_{\mathrm{EG}(2,4)}$ is an 8×15 submatrix (with five columns having weight only one) of consecutive rows of the (circulant) matrix $\boldsymbol{H}_{\mathrm{EG}(2,4)}$. It can easily be seen that for

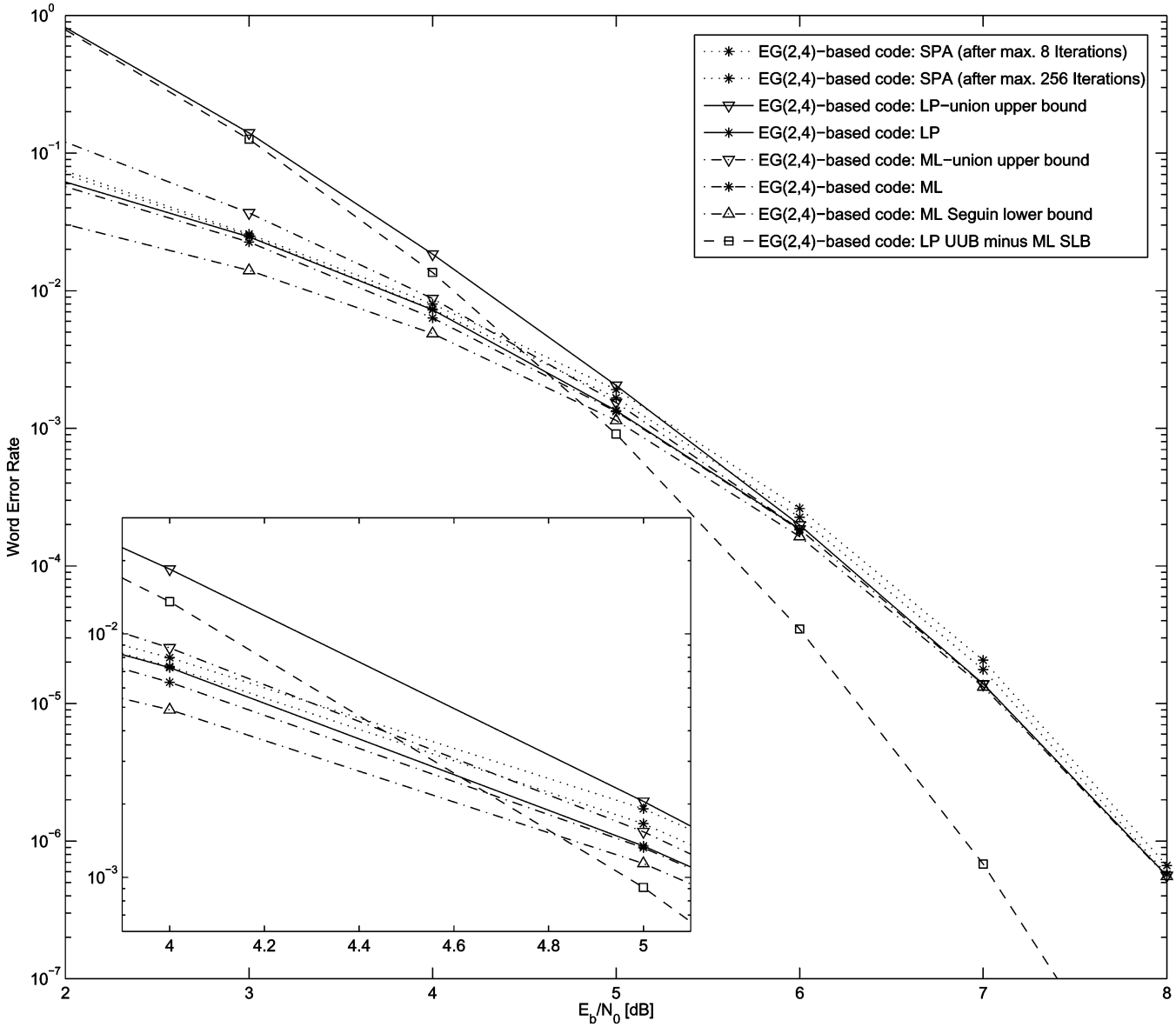

Fig. 3. Word-error rate for various decoding algorithms together with some upper and lower bounds. (See main text for explanations.) The inset highlights the curves in the range $E_{\mathrm{b}}/N_0 = 4\ \mathrm{dB}, \ldots, 5\ \mathrm{dB}$.

the parity-check matrices under investigation those with more dependent rows lead to more favorable histograms.

In Fig. 3, we show various decoding simulation results for data transmission over a binary-input AWGNC and lower and upper bounds: $\boldsymbol{H}_{\mathrm{EG}(2,4)}$-based sum-product algorithm (SPA) decoding (cf. [29]), $\boldsymbol{H}_{\mathrm{EG}(2,4)}$-based LP decoding, $\mathcal{C}_{\mathrm{EG}(2,4)}$-based ML decoding, an upper bound on LP decoding based on a union of events upper bound, an upper bound on ML decoding based on a union of events upper bound, and a lower bound on ML decoding based on an inequality by de Caen as presented by Séguin [30]. It can be seen that thanks to the knowledge of minimal codewords and minimal pseudo-codewords we are able to obtain bounds that are very tight from a certain SNR value on. This is witnessed by the fast decreasing line labeled "LP UUB minus ML SLB" which shows the difference between the union upper bound on the LP decoder word-error rate and the Séguin lower bound on the ML decoder word-error rate.

E. How the Results Were Obtained

Let us briefly mention how the results for the minimal pseudo-codewords were obtained in Section VI-A, -B, and -D. We used the program "lrs" [31] to search edges in cones. For the code $\mathcal{C}_{\mathrm{PG}(2,4)}$ in Section VI-B, we additionally used the two-transitivity of the points of a projective plane in order to formulate a simpler edge-enumeration subproblem which can be solved efficiently and from which all the minimal pseudo-codewords can be derived. This goes as follows: it is clear that a minimal pseudo-codeword cannot only fulfill inequalities of type (7) with equality (because only $\boldsymbol{\omega} = \mathbf{0}$ does this), i.e., at least one inequality of type (6) must be fulfilled with equality. Because the automorphism group of $\mathrm{PG}(2,4)$ is two-transitive (which means that for any two pair of points there exists an automorphism that maps the first pair onto the second pair) and because of the structure of the inequalities in (6), we can pick any of the inequalities of type (6) to be fulfilled with equality. Solving the

edge enumeration problem for this $(n-1)$-dimensional cone is simpler and using the automorphism group we can derive from this the minimal pseudo-codewords of the cone of interest.

F. Connection to Stopping Sets

Before finishing this section that showed some pseudo-weight enumerators, let us comment on stopping set weight enumerators for PG-based codes that were investigated by Kashyap and Vardy [32].[16] These stopping set weight enumerators are tightly related to BEC pseudo-weight enumerators because of the following reasons [5], [6], [2]: the support of any pseudo-codeword is a stopping set whose size equals the BEC pseudo-weight of the pseudo-codeword. Moreover, for every stopping set, there is at least one pseudo-codeword whose support equals that stopping set. (Note that the summation in the stopping set weight enumerator definition in [32] went over all stopping sets, whereas our enumerators only count stopping sets related to minimal pseudo-codewords.)

VII. Bounds on the AWGNC Pseudo-Weight

In this section, we will give some bounds on the AWGNC pseudo-weight of vectors with real nonnegative components. First, we discuss some general bounds on the pseudo-weight of arbitrary vectors in $\mathbb{R}_+^n$. As we will see, these bounds depend only on the *type* of the vector, a concept defined next. Obviously, the obtained results can be applied to pseudo-codewords of any Tanner graph, not just to pseudo-codewords of $\mathrm{PG}(2,q)$- and $\mathrm{EG}(2,q)$-based Tanner graphs.

After this, however, we will focus on $\mathrm{PG}(2,q)$-based Tanner graphs and present results for their pseudo-codewords. We will mainly present results for pseudo-codewords that have a certain structure, namely, for pseudo-codewords whose components take on only the values zero, one, two, or three (after suitable rescaling of the pseudo-codeword). Based on the examples in Section VI, we formed the belief that minimal pseudo-codewords with small pseudo-weight have this structure and are therefore the most problematic ones for LP decoding of $\mathrm{PG}(2,q)$-based codes.

Definition 9: Let $\boldsymbol{\omega} \in \mathbb{R}_+^n$ and let $t_\ell \triangleq t_\ell(\boldsymbol{\omega})$ be the number of components of the vector $\boldsymbol{\omega}$ that are equal to ℓ, where $\ell \in \mathbb{R}_+$. Then, we call $\boldsymbol{t} \triangleq \boldsymbol{t}(\boldsymbol{\omega}) = (t_\ell(\boldsymbol{\omega}))_{\ell \in \mathbb{R}_+}$ the type of $\boldsymbol{\omega}$. □

Note that we do not assume that ℓ is a nonnegative integer, only that it is a nonnegative real number. It follows from this definition that only finitely many t_ℓ's are nonzero and that $\sum_\ell t_\ell = |\mathcal{I}| = n$ for any $\boldsymbol{\omega} \in \mathbb{R}_+^n$. Moreover, because $|\mathrm{supp}(\boldsymbol{\omega})| = \sum_{\ell>0} t_\ell$, $\|\boldsymbol{\omega}\|_1 = \sum_\ell \ell t_\ell$, and $\|\boldsymbol{\omega}\|_2^2 = \sum_\ell \ell^2 t_\ell$, we have

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) = \frac{\left(\sum_\ell \ell t_\ell\right)^2}{\sum_\ell \ell^2 t_\ell} \quad \text{and} \quad w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega}) = \sum_{\ell>0} t_\ell.$$

[16]Note that Kashyap and Vardy used the following nomenclature: 1) a minimal stopping set is a stopping set whose size equals the stopping distance (which equals the minimum BEC pseudo-weight), 2) an irreducible stopping set $\mathcal{S}$ is a stopping set such that there is no nonempty stopping set that is a proper subset of $\mathcal{S}$. In other words, our use of the word "minimal" corresponds more to their use of word "irreducible" and not to their use of the word "minimal."

If $\tilde{\boldsymbol{\omega}} = \alpha \cdot \boldsymbol{\omega}$ for some $\alpha \in \mathbb{R}_{++}$ then its type $\tilde{\boldsymbol{t}} \triangleq \boldsymbol{t}(\tilde{\boldsymbol{\omega}})$ is such that $\tilde{t}_{\alpha\ell} = t_\ell$ for all ℓ.

A. Arbitrary Vectors in $\mathbb{R}_+^n$

Lemma 10: Let $\boldsymbol{\omega} \in \mathbb{R}_+^n$ and let $\eta \neq 0$ be some arbitrary real number. Then

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{2\eta\|\boldsymbol{\omega}\|_1 - \|\boldsymbol{\omega}\|_2^2}{\eta^2} = \frac{\sum_{i=1}^n \omega_i(2\eta - \omega_i)}{\eta^2}$$

with equality if and only if $\boldsymbol{\omega} = \mathbf{0}$ or $\eta = \|\boldsymbol{\omega}\|_2^2/\|\boldsymbol{\omega}\|_1$.

Proof: If $\boldsymbol{\omega} = \mathbf{0}$ then the statement is certainly true, so let us assume that $\boldsymbol{\omega} \neq \mathbf{0}$. The square of any real number is nonnegative, therefore, $\left(\eta\|\boldsymbol{\omega}\|_1 - \|\boldsymbol{\omega}\|_2^2\right)^2 \geqslant 0$, with equality if and only if $\eta = \|\boldsymbol{\omega}\|_2^2/\|\boldsymbol{\omega}\|_1$, which, after rearranging, gives $\eta^2\|\boldsymbol{\omega}\|_1^2 \geqslant 2\eta\|\boldsymbol{\omega}\|_1\|\boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}\|_2^4$ Finally, dividing by $\eta^2\|\boldsymbol{\omega}\|_2^2$ and using the definition of $w_{\mathrm{p}}(\boldsymbol{\omega})$, we obtain the desired result. □

Corollary 11: Let $\boldsymbol{\omega} \in \mathbb{R}_+^n$, let $\boldsymbol{t} \triangleq \boldsymbol{t}(\boldsymbol{\omega})$ be the type of $\boldsymbol{\omega}$, and let $\eta \neq 0$ be some arbitrary real number. Then

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \sum_\ell \beta_\ell t_\ell \quad \text{with}$$
$$\beta_\ell = \frac{\ell(2\eta - \ell)}{\eta^2} = 1 - \left(1 - \frac{\ell}{\eta}\right)^2.$$

Proof: The result follows immediately from Lemma 10. □

Corollary 12: Let $\boldsymbol{\omega} \in \mathbb{R}_+^n$ and let $\boldsymbol{t} \triangleq \boldsymbol{t}(\boldsymbol{\omega})$. Moreover, let r be the ratio of the largest positive ℓ such that t_ℓ is nonzero and the smallest positive ℓ such that t_ℓ is nonzero. Then we have the lower bound

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{4r}{(r+1)^2} \cdot |\mathrm{supp}(\boldsymbol{\omega})|.$$

This bound was also obtained by Wauer [33], [34] using a different derivation.

Proof: Let m be the largest positive ℓ such that t_ℓ is nonzero and let m' be the smallest positive ℓ such that t_ℓ is nonzero. These definitions obviously yield $r = m/m'$. Consider Corollary 11 with $\eta = \frac{m+m'}{2}$. We obtain $w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \sum_\ell \beta_\ell t_\ell$ (a) with

$$\beta_\ell = 4\ell\frac{m+m'-\ell}{(m+m')^2} = 1 - \left(1 - \frac{2\ell}{m+m'}\right)^2.$$

We observe that

$$\beta_{m'} = \beta_m = \frac{4mm'}{(m+m')^2} = \frac{4r}{(r+1)^2}.$$

Since β_ℓ is strictly concave in ℓ we must have $\beta_\ell > \beta_{m'} = \beta_m = \frac{4r}{(r+1)^2}$ for all $m' < \ell < m$. It follows that

$$w_{\mathrm{p}}(\boldsymbol{\omega}) = \sum_{m' \leqslant \ell \leqslant m} \beta_\ell t_\ell \geqslant \sum_{m' \leqslant \ell \leqslant m} \frac{4r}{(r+1)^2} t_\ell$$
$$= \frac{4r}{(r+1)^2} \sum_{m' \leqslant \ell \leqslant m} t_\ell = \frac{4r}{(r+1)^2} \cdot |\mathrm{supp}(\boldsymbol{\omega})|. \quad \square$$

Under the same assumptions as in Corollary 12 , Kelley and Sridhara [35] proved that

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{2r^2}{(1+r^2)(r-1)+2r} \cdot |\mathrm{supp}(\boldsymbol{\omega})|.$$

Note that for $r = 1$ and $r = 2$, the bound in Corollary 12 equals this bound and that for integers r larger than 2, the bound in Corollary 12 is larger than this bound. (Note that for a minimal pseudo-codeword $\boldsymbol{\omega}$ the ratio r is 1 or at least 2.)[17]

B. Pseudo-Codewords With Zeros and Ones From $\mathrm{PG}(2,q)$-Based Tanner Graphs

After having considered general vectors in $\mathbb{R}_+^n$, the following subsections will mainly focus on pseudo-codewords of $\mathrm{PG}(2,q)$-based Tanner graphs. We start our analysis with pseudo-codewords of smallest possible entries, i.e., pseudo-codewords with zeros and ones. The following theorem gives a lower bound on their weight.

Theorem 13: Let $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$ and let $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ be a (not necessarily minimal) pseudo-codeword of type $\boldsymbol{t}$ with t_0 nonnegative, t_1 positive, and $t_\ell = 0$ otherwise. If $\boldsymbol{\omega}$ is not a codeword in $\mathcal{C}_{\mathrm{PG}(2,q)}$ then

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \left\lceil \frac{q}{2} + 1 + \frac{1}{2}\sqrt{q^2+16q+16} \right\rceil \geqslant q+4.$$

Proof: See the Appendix, part A. □

Note that the above bound yields $w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant q+4$ for $q = 2, 4$, and $w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant q+5$ for $q \geqslant 8$.

An example of a non-codeword pseudo-codeword with only zeros and ones as discussed in Theorem 13 was presented at the end of Section VI-A. Note, however, that this example was for $\boldsymbol{H}'_{\mathrm{PG}(2,2)}$ in (9) and not for $\boldsymbol{H}_{\mathrm{PG}(2,2)}$ in (8).

Observations for small $\mathrm{PG}(2,q)$-based codes suggest the following conjecture.

Conjecture 14: Let $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$ and let $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ be a minimal pseudo-codeword of type $\boldsymbol{t}$ with t_0 nonnegative, t_1 positive, and $t_\ell = 0$ otherwise. The vector $\boldsymbol{\omega}$ is then a minimal codeword.

It is well known, cf., e.g., [18], that a minimal codeword of an $[n,k]$ binary linear code satisfies $|\mathrm{supp}(\boldsymbol{x})| = w_{\mathrm{H}}(\boldsymbol{x}) \leqslant n-k+1$. In the following theorem, we generalize this upper bound on the Hamming weight of minimal codewords to an upper bound on the pseudo-weight of minimal pseudo-codewords whose components are either zero or one. Note that this bound holds for any parity-check matrix.

Lemma 15: Let $\mathcal{C}$ be an $[n,k]$ binary linear code represented by a parity-check matrix $\boldsymbol{H}$. (Note that we do not assume that $\mathcal{C} = \mathcal{C}_{\mathrm{PG}(2,q)}$ or $\mathcal{C} = \mathcal{C}_{\mathrm{EG}(2,q)}$.) Let $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ be a minimal pseudo-codeword of type $\boldsymbol{t}$ with t_0 nonnegative, t_1 positive, and $t_\ell = 0$ otherwise. Then

$$w_{\mathrm{p}}(\boldsymbol{\omega}) = w_{\mathrm{H}}(\boldsymbol{\omega}) \leqslant n-k+1.$$

Proof: Without loss of generality, we can assume that the pseudo-codeword indices have been reordered such that the first $n_1 \triangleq w_{\mathrm{H}}(\boldsymbol{\omega})$ components of $\boldsymbol{\omega}$ are equal to one and such that the remaining $n_2 \triangleq n - w_{\mathrm{H}}(\boldsymbol{\omega})$ components of $\boldsymbol{\omega}$ are equal to zero.

Let $\boldsymbol{K}_1\boldsymbol{\omega} \geqslant \boldsymbol{0}$ be the collection of inequalities of type (6) that a pseudo-codeword must fulfill and let $\boldsymbol{K}_2\boldsymbol{\omega} \geqslant \boldsymbol{0}$ be the collection of inequalities of type (7) that a pseudo-codeword must fulfill. Then there exists a full-rank $(n_1-1)\times n$-submatrix $\boldsymbol{A}_1 \triangleq (\boldsymbol{A}_{11} \mid \boldsymbol{A}_{12})$ of $\boldsymbol{K}_1$ and a full-rank $n_2 \times n$-submatrix of $\boldsymbol{A}_2 \triangleq (\boldsymbol{A}_{21} \mid \boldsymbol{A}_{22})$ of $\boldsymbol{K}_2$ such that $\boldsymbol{A}_1\boldsymbol{\omega} = \boldsymbol{0}$ and $\boldsymbol{A}_2\boldsymbol{\omega} = \boldsymbol{0}$, such that $\boldsymbol{A}_{21} = \boldsymbol{0}$ and $\boldsymbol{A}_{22} = \boldsymbol{I}_{n_2}$, and such that $\mathrm{rank}_{\mathbb{R}}(\boldsymbol{A}) = n-1$, where

$$\boldsymbol{A} \triangleq \begin{pmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{0} & \boldsymbol{I}_{n_2} \end{pmatrix}.$$

Applying elementary row operations to the matrix $\boldsymbol{A}$, we obtain the matrix

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n_2} \end{pmatrix}$$

. Because all nonzero entries of $\boldsymbol{\omega}$ are equal to one and because $\boldsymbol{\omega}$ is assumed to be a minimal pseudo-codeword, it turns out that all rows of $\boldsymbol{A}_{11}$ must contain exactly two nonzero elements, one equal to $+1$ and one equal to -1.

These facts can be used as follows. First, we will show that $\mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{A}_{11}) = n_1 - 1$. Second, we will show that $\mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{A}_{11}) \leqslant n-k$. Finally, combining these results we will obtain the desired statement that

$$w_{\mathrm{p}}(\boldsymbol{\omega}) = w_{\mathrm{H}}(\boldsymbol{\omega}) = n_1 \leqslant n-k+1.$$

So, let us show that $\mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{A}_{11}) = n_1 - 1$, i.e., that $\mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{A}_{11}) = \mathrm{rank}_{\mathbb{R}}(\boldsymbol{A}_{11})$. Indeed, using the special row structure of $\boldsymbol{A}_{11}$, it can be verified that the only vector in the (right-hand side) kernel of $\boldsymbol{A}_{11} \pmod 2$ is the all-ones vector over $\mathbb{F}_2$ of length n_1.

Second, let us show that $\mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{A}_{11}) \leqslant n-k$. Indeed, we observe that every row in $\boldsymbol{A}_1 \pmod 2$ corresponds to a row in $\boldsymbol{H}$. This implies that $\mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{A}_{11}) \leqslant \mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{A}_1) \leqslant \mathrm{rank}_{\mathbb{F}_2}(\boldsymbol{H}) \leqslant n-k$. □

A crucial element in the above proof was the fact that all rows of $\boldsymbol{A}_{11}$ contain exactly two nonzero entries, one equal to $+1$ and one equal to -1. For minimal pseudo-codewords where not all nonzero entries are equal, this is not the case anymore and therefore we cannot use the above proof to generalize the lemma statement to other types of minimal pseudo-codewords.

C. Pseudo-Codewords With Zeros, Ones, and Twos From $\mathrm{PG}(2,q)$-Based Tanner Graphs

In this subsection, we consider pseudo-codewords that have not only zeros and ones, but also twos, as components. We have the following result.

[17](To keep the notation simple, we show here only the proof for a code where all the check nodes have degree 3; the generalization to other codes is straightforward.) The claim follows from the following observations. First, $r \geqslant 1$ by definition. Second, if $r > 1$ then there must be at least one inequality of type (6), say $\omega_{i_1} \leqslant \omega_{i_2} + \omega_{i_3}$, that is fulfilled with equality and where $\omega_{i_1}, \omega_{i_2}$, and ω_{i_3} are nonzero. From $\omega_{i_1} = \omega_{i_2} + \omega_{i_3}$ it follows that $\min(\omega_{i_2}, \omega_{i_3}) \leqslant \omega_{i_1}/2$. Finally, because $r \geqslant \omega_{i_1}/\min(\omega_{i_2}, \omega_{i_3})$ we must have $r \geqslant 2$.

Theorem 16: Let $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$ and let $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ be of type $\boldsymbol{t}$ with t_0 nonnegative, t_1 positive, t_2 positive, and $t_\ell = 0$ otherwise. Then

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{32}{27}(q+2) \approx 1.185(q+2).$$

Proof: For any $i \in \mathcal{I}$ we must have

$$\sum_{i' \in \mathcal{I} \setminus \{i\}} \omega_{i'} \overset{\text{(a)}}{=} \sum_{j \in \mathcal{J}_i} \sum_{i' \in \mathcal{I}_j \setminus \{i\}} \omega_{i'} \overset{\text{(b)}}{\geqslant} \sum_{j \in \mathcal{J}_i} \omega_i = (q+1)\omega_i$$

where at step (a) we used the fact that all variable nodes are at graph distance two from each other in the Tanner graph associated to $\boldsymbol{H}$, and where at step (b) we used the inequalities in (6). Adding ω_i to both sides we obtain $\sum_{i' \in \mathcal{I}} \omega_{i'} \geqslant (q+2)\omega_i$. Now, fix an $i \in \mathcal{I}$ for which $\omega_i = 2$ holds and express $\sum_{i' \in \mathcal{I}} \omega_{i'}$ in terms of $\boldsymbol{t}$: it must hold that $t_1 + 2t_2 \geqslant 2(q+2)$ (c).

In the second step, we construct a vector $\boldsymbol{\omega}' = (\omega'_1, \ldots, \omega'_n) \in \mathbb{R}^n$ such that

$$\omega'_i \triangleq \begin{cases} 0, & \text{if } \omega_i = 0 \\ 2, & \text{if } \omega_i = 1 \\ 1, & \text{if } \omega_i = 2. \end{cases} \quad (\text{for all } i \in \mathcal{I}).$$

It can easily be seen that $\boldsymbol{\omega}'$ lies also in the fundamental cone, i.e., $\boldsymbol{\omega}' \in \mathcal{K}(\boldsymbol{H})$, and that $\boldsymbol{\omega}'$ has type $\boldsymbol{t}'$ with $t'_1 = t_2$ positive, $t'_2 = t_1$ positive, and $t'_\ell = t_\ell$ otherwise. In other words, switching $0 \mapsto 0$, $1 \mapsto 2$, $2 \mapsto 1$ we obtain another pseudo-codeword. Arguing as above, for any $i \in \mathcal{I}$ we must have $\sum_{i' \in \mathcal{I}} \omega'_{i'} \geqslant (q+2)\omega'_i$. Now, fix an $i \in \mathcal{I}$ for which $\omega'_i = 2$ holds, and express $\sum_{i' \in \mathcal{I}} \omega'_{i'}$ in terms of $\boldsymbol{t}'$: it must hold that $t'_1 + 2t'_2 \geqslant 2(q+2)$, i.e., that $t_2 + 2t_1 \geqslant 2(q+2)$ (d).

Combining (c) and (d) we obtain $3(t_1 + t_2) \geqslant 4(q+2)$, i.e., $|\mathrm{supp}(\boldsymbol{\omega})| = t_1 + t_2 \geqslant \frac{4}{3}(q+2)$. Using Corollary 12 we can conclude that

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{4 \cdot 2}{(2+1)^2} \cdot \frac{4}{3}(q+2) = \frac{8}{9} \cdot \frac{4}{3}(q+2) = \frac{32}{27}(q+2). \quad \square$$

Using some stronger assumptions on the pseudo-codeword $\boldsymbol{\omega}$ we can obtain a stronger lower bound, as is shown in the next theorem.

Theorem 17: Let $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$ and let $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ be of type $\boldsymbol{t}$ with t_0 nonnegative, $t_1 \geqslant q+2$, t_2 positive, and $t_\ell = 0$ otherwise.[18] Then

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{4}{3}(q+2) \approx 1.333(q+2).$$

Proof: The start is similar to the beginning of the proof of Theorem 16. For any $i \in \mathcal{I}$ we must have $\sum_{i' \in \mathcal{I}} \omega_{i'} \geqslant (q+2)\omega_i$. Now, fix an $i \in \mathcal{I}$ for which $\omega_i = 2$ holds, and express $\sum_{i' \in \mathcal{I}} \omega_{i'}$ in terms of $\boldsymbol{t}$: it must hold that $t_1 + 2t_2 \geqslant 2(q+2)$, or, equivalently, $t_2 \geqslant q + 2 - t_1/2$ (a). For any $\eta \geqslant 1$ we obtain

$$\begin{aligned} w_{\mathrm{p}}(\boldsymbol{\omega}) &\overset{\text{(b)}}{\geqslant} \frac{(2\eta-1)t_1 + (4\eta-4)t_2}{\eta^2} \\ &\overset{\text{(c)}}{\geqslant} \frac{(2\eta-1)t_1 + (4\eta-4)(q+2-t_1/2)}{\eta^2} \\ &= \frac{t_1 + (4\eta-4)(q+2)}{\eta^2} \end{aligned}$$

[18]See Remark 18 for a comment on these conditions.

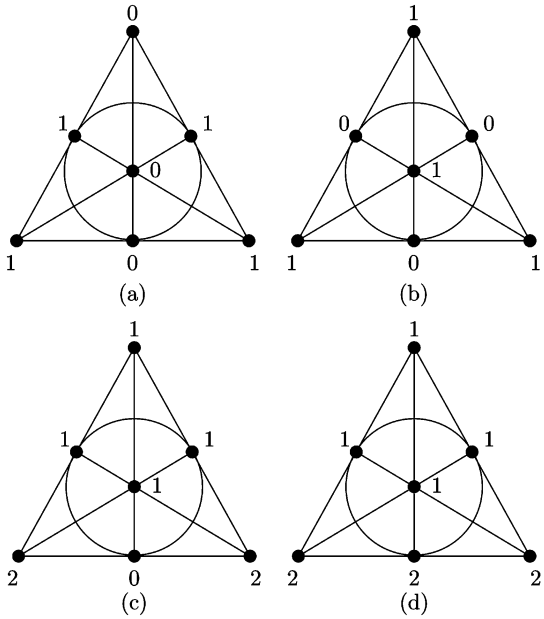

Fig. 4. Codewords and pseudo-codewords used in Example 19.

where at step (b) we used Corollary 11 and at step (c) we used the inequality (a) and the assumption that $\eta \geqslant 1$, i.e., $4\eta - 4 \geqslant 0$. Using the assumption that $t_1 \geqslant q+2$ from the theorem statement we get $w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{(4\eta-3)(q+2)}{\eta^2}$. The right-hand side of this expression is maximized by $\eta^* = \frac{3}{2}$: inserting this value yields the lower bound in the theorem statement. (Note that $\eta^* \geqslant 1$ and so η^* fulfills the assumption about η that was made in the derivation.) $\square$

Remark 18: Let $\mathcal{C}$ be the code defined by $\boldsymbol{H}$. If a pseudo-codeword is an unscaled pseudo-codeword [20], [6] then it is equal (modulo 2) to a codeword of $\mathcal{C}$. Therefore, the number of odd components of an unscaled pseudo-codeword must either be zero or at least equal to the minimum Hamming weight of the code. So, if we actually know that $\boldsymbol{\omega}$ in Theorem 17 is an unscaled pseudo-codeword then the requirement $t_1 \geqslant q+2$ in the theorem statement is equivalent to the requirement $t_1 \geqslant 1$.

Note also that Theorem 17 can be generalized to the setup where $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ has type $\boldsymbol{t}$ with t_0 nonnegative, t_m positive for some integer $m \geqslant 2$, t_ℓ nonnegative for $1 \leqslant \ell \leqslant m-1$, $t_\ell = 0$ for $\ell \geqslant m+1$, and $\sum_{\text{odd } \ell} t_\ell \geqslant q+2$. Then $w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{m^2}{m^2-m+1}(q+2)$. $\square$

Example 19: Let us exhibit some low-weight minimal pseudo-codewords that contain only zeros, ones, and twos. Consider first the case $q = 2$. The projective plane for $q = 2$ is shown in Fig. 4(a): it has seven points and seven lines and we consider the points to be variables and the lines to be checks. Fig. 4(a) and (b) shows two codewords of weight $q + 2 = 4$; note that their supports overlap in $\frac{q+2}{2} = 2$ positions. Adding these two codewords together yields the pseudo-codeword shown in Fig. 4(c). Switching the zero into a two results in the pseudo-codeword in Fig. 4(d); it can be checked that this pseudo-codeword is actually a minimal pseudo-codeword. It has AWGNC pseudo-weight 6.25, whereas the lower bounds in Theorems 16 and 17 are 4.74 and 5.33, respectively.

Similarly, in the case of $q = 4$ it is possible to start with two codewords of weight $q+2 = 6$ whose supports overlap in $\frac{q+2}{2} = 3$ positions. After adding them and switching two zeros (that are specifically chosen and lie on the same line) into two twos, one gets a minimal pseudo-codeword of AWGNC pseudo-weight 9.85, whereas the lower bounds in Theorems 16 and 17 are 7.11 and 8.00, respectively.

In the case $q = 8$ it is possible to start with two codewords of weight $q + 2 = 10$ whose supports overlap in $\frac{q+2}{2} = 5$ positions. After adding them and switching three zeros (that are specifically chosen and form a triangle) into two twos, one gets a minimal pseudo-codeword of AWGNC pseudo-weight 16.10, whereas the lower bound in Theorems 16 and 17 are 11.85 and 13.33, respectively.

We conjecture that, with suitable generalizations, the above construction can be extended to larger q. □

D. Pseudo-Codewords With Zeros, Ones, Twos, and Threes From $\mathrm{PG}(2,q)$-Based Tanner Graphs

Finally, we consider pseudo-codewords that not only have zeros, ones, and two, but also threes.

Theorem 20: Let $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$ and let $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ be of type $\boldsymbol{t}$ with t_0 nonnegative, t_1 positive, t_2 nonnegative, t_3 positive, and $t_\ell = 0$ otherwise. We require that $\boldsymbol{\omega}$ is an unscaled pseudo-codeword. Then

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{9}{8} \cdot (q+2) = 1.125(q+2).$$

Proof: The start is similar to the beginning of the proof of Theorem 16. For any $i \in \mathcal{I}$ we must have $\sum_{i' \in \mathcal{I}} \omega_{i'} \geqslant (q+2)\omega_i$. Now, fix an $i \in \mathcal{I}$ for which $\omega_i = 3$ holds and express $\sum_{i' \in \mathcal{I}} \omega_{i'}$ in terms of $\boldsymbol{t}$: it must hold that $t_1 + 2t_2 + 3t_3 \geqslant 3(q+2)$ (a).

In a second step, we construct a vector $\boldsymbol{\omega'} = (\omega'_1, \ldots, \omega'_n) \in \mathbb{R}^n$ such that

$$\omega'_i \triangleq \begin{cases} 3, & \text{if } \omega_i = 1 \\ 2, & \text{if } \omega_i = 2 \\ 1, & \text{if } \omega_i = 3 \\ 0, & \text{otherwise.} \end{cases} \quad \text{(for all } i \in \mathcal{I})$$

It can be seen that $\boldsymbol{\omega'}$ lies also in the fundamental cone, i.e., $\boldsymbol{\omega'} \in \mathcal{K}(\boldsymbol{H})$,[19] and that $\boldsymbol{\omega'}$ has type $\boldsymbol{t'}$ with $t'_1 = t_3$ positive, $t'_2 = t_2$ nonnegative, $t'_3 = t_1$ positive, and $t'_\ell = t_\ell$ otherwise. Arguing as above, for any $i \in \mathcal{I}$ we must have $\sum_{i' \in \mathcal{I}} \omega'_{i'} \geqslant (q+2)\omega'_i$. Now, fix an $i \in \mathcal{I}$ for which $\omega'_i = 3$ holds and express $\sum_{i' \in \mathcal{I}} \omega'_{i'}$ in terms of $\boldsymbol{t'}$: it must hold that $t'_1 + 2t'_2 + 3t'_3 \geqslant 3(q+2)$, i.e., that $t_3 + 2t_2 + 3t_1 \geqslant 3(q+2)$ (b).

Combining (a) and (b) we obtain $4(t_1+t_2+t_3) \geqslant 6(q+2)$, i.e.,

$$|\mathrm{supp}(\boldsymbol{\omega})| = t_1 + t_2 + t_3 \geqslant \frac{3}{2}(q+2).$$

Using Corollary 12, we can conclude that

$$w_{\mathrm{p}}(\boldsymbol{\omega}) \geqslant \frac{4 \cdot 3}{(3+1)^2} \cdot \frac{3}{2}(q+2) = \frac{3}{4} \cdot \frac{3}{2}(q+2) = \frac{9}{8}(q+2). \quad \square$$

[19]Note that the inequality $3 + 3 \geqslant 1$ goes into the inequality $1 + 1 \geqslant 3$, which is wrong. However, we assumed that $\boldsymbol{\omega}$ is an unscaled pseudo-codeword, which, among other things, implies that the modulo-2 sum of the ω_i's that are involved in a check is zero. Therefore, it cannot happen that the nonzero ω_i's that are involved in a check have the values 3, 3, and 1.

VIII. The Structure of Minimal Pseudo-Codewords

In this section, we discuss the geometry of minimal pseudo-codewords. Minimum-weight codewords correspond to point-line configurations in the projective plane that have been studied by several authors. Let us introduce some notation and results from finite geometries, cf., e.g., [11]. A k-arc in $\mathrm{PG}(2,q)$ is a set of k points no three of which are collinear. A k-arc is complete if it is not contained in a $(k+1)$-arc. The maximum number of points that a k-arc can have is denoted by $m(2,q)$, and a k-arc with this number of points is called an oval (in the case where q is even this is sometimes also called a hyper-oval). One can show that $m(2,q) = q+2$ for q even and $m(2,q) = q+1$ for q odd. One can make the following two interesting observations for the case q even. First, if two ovals have more than half their points in common, then these two ovals coincide. Second, if a q-arc is contained in an oval then the number of such ovals is one if $q > 2$ and two if $q = 2$

It turns out that in the case q even, the codewords with minimal weight are $q+2$-arcs and therefore ovals. However, whereas the classification of ovals for odd q is simple (they all correspond to conics), the ovals for even q are not classified that easily. For even q, one says that an oval is regular if it comprises the points of a conic and its nucleus; one can show that for $q = 2^s$, irregular ovals exist if and only if $s \geqslant 4$. It turns out that the classification for irregular ovals is highly nontrivial.[20] So, given that even the classification of the codewords of minimal weight is difficult, it is probably hopeless to obtain a complete classification of the minimal codewords and minimal pseudo-codewords of codes defined by $\boldsymbol{H}_{\mathrm{PG}(2,q)}$, however, it is an interesting goal to try to understand as much as possible about the structure of these codewords and pseudo-codewords.

In some recent papers, the structure of codewords of projective-plane-based codes has been discussed by Kashyap and Vardy [32] (that paper deals also with stopping sets in Tanner graphs derived from projective planes), by Justesen *et al.* [36], [37], and by Laendner and Milenkovic [38] (that paper also deals with trapping sets in Tanner graphs derived from projective planes). Moreover, the minimal-weight codewords of Euclidean-plane-based codes were discussed by Høholdt *et al.* [39]; also, here these configurations are tightly related to ovals. However, because not all ovals are regular, the classification is not that simple also for these codes.[21]

From now on, we will only consider projective planes $\mathrm{PG}(2,q)$ and q will always be even, i.e., a power of two. Before we state our conjecture about the structure of minimal pseudo-codewords, let us first look at an example.

Example 21: Let $q = 4$. We can find a minimal pseudo-codeword $\boldsymbol{\omega}$ whose type $\boldsymbol{t}$ is $t_0 = 8$, $t_1 = 8$, $t_2 = 5$, and $t_\ell = 0$ otherwise. This pseudo-codeword can be obtained using a procedure similar to the one used in Example 19. First, one must add two minimal codewords $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ of weight 6

[20]According to [11, Theorem 8.35], there are precisely two projectively distinct ovals in $\mathrm{PG}(2,16)$, the so-called regular oval $\mathcal{D}(T^2)$ and the so-called oval $\mathcal{O}_0 = \mathcal{D}(F_0)$. Moreover, according to [11, Theorem 8.36], there are precisely six projectively distinct ovals in $\mathrm{PG}(2,32)$.

[21]Note that the remark "It is also known that an oval in $\mathrm{EG}(2,q)$ consists of a conic and a nucleus" (which would imply that they are regular) in [39, Sec.2] is wrong in general [40].

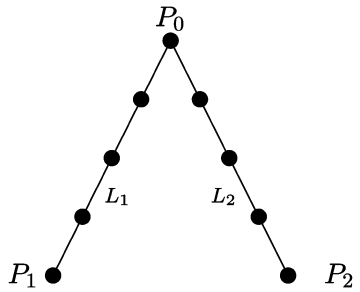

Fig. 5. Part of $\mathrm{PG}(2,4)$ discussed in Example 21.

whose supports overlap in two positions. This yields a pseudo-codeword $\tilde{\boldsymbol{\omega}}$ of type $\tilde{\boldsymbol{t}}$ with $\tilde{t}_0 = 11$, $\tilde{t}_1 = 8$, $\tilde{t}_2 = 2$, and $\tilde{t}_\ell = 0$ otherwise. Second, in order to obtain a minimal pseudo-codeword, one must switch three zeros (that were appropriately chosen) into three twos.

Let us analyze this procedure. Since a minimal pseudo-codeword corresponds to an edge of the fundamental cone, it is clear that the inequalities in (6) and (7) that are fulfilled with equality must form a system of linear equations of rank $21 - 1 = 20$. We start with two minimal codewords $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ that each yield a system of linear equations of rank $21 - 1 = 20$. These two codewords have been chosen such that their sum $\tilde{\boldsymbol{\omega}}$ yields a system of linear equations of rank $21 - 2 = 19$.

To find the three zeros that we have to switch, we proceed as follows. It turns out that in the projective plane $\mathrm{PG}(2,4)$ there are two lines, say L_1 and L_2, such that all the entries of $\tilde{\boldsymbol{\omega}}$ that correspond to the points on these two lines are zero. Let P_0 be the intersection point of these two lines, cf. Fig. 5. There exists a point P_1 on L_1 and a point P_2 on L_2 such that setting the entries of $\tilde{\boldsymbol{\omega}}$ corresponding to P_0, P_1, and P_2 equal to the same value $\alpha \geqslant 0$ yields a vector in the fundamental cone, as long as α is not too large. In fact, for $\alpha > 2$, the vector is outside the fundamental cone, and for $\alpha = 2$, it yields a vector that is a pseudo-codeword, and that yields a system of equations of rank $21 - 1 = 20$, i.e., it is a minimal pseudo-codeword. □

Conjecture 22: For the Tanner graph defined by $\boldsymbol{H}_{\mathrm{PG}(2,q)}$ every minimal pseudo-codeword is a sum of a few minimal pseudo-codewords with a change of a few low-value components such that they become the large components in the equations associated to the lines that pass through them.

If this conjecture is true, finding minimal pseudo-codewords reduces to taking sums of two minimal pseudo-codewords that give rank $n-2$ (if possible; lower otherwise) and changing few components that are "not significant" into a "significant" ones. We call a component significant if it is the sum of the other components that belong to a line passing through the point, for most of such lines.

The following proposed open problem is based on our observations (for small q) that among the minimal pseudo-codewords of $\mathrm{PG}(2,q)$-based codes the ones with only zeros, ones, and twos yield non-codeword minimal pseudo-codewords of smallest weight. A positive answer to this conjecture would result in a much better understanding of the minimal pseudo-codewords in general and of the AWGNC pseudo-weight spectrum gap, in particular.

Open Problem 23: Let $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$ and consider the pseudo-codewords that have minimal AWGNC pseudo-weight among all minimal pseudo-codewords that are not multiples of minimal codewords. Then find the smallest $\tilde{\ell}$ such that these pseudo-codewords have type $\boldsymbol{t}$ with $t_\ell \geqslant 0$ for $\ell \in \{0, 1, \ldots, \tilde{\ell}\}$ and $t_\ell = 0$ otherwise.

IX. Effective Minimal Pseudo-Codewords

The preceding two sections focused on the properties of vectors in the fundamental polytope of some Tanner graphs. In this section, we broaden our view and discuss the relevance of minimal pseudo-codewords in the context of LP decoding.

LP decoding always gives back a vertex of the fundamental polytope of the code under consideration.[22] Clearly, what vertex is chosen is a function of the LLR vector $\boldsymbol{\lambda}$.[23] If the channel is such that $\boldsymbol{\lambda}$ can be any vector in $(\mathbb{R} \cup \{\pm\infty\})^n$, then for every vertex of the fundamental polytope there is at least one $\boldsymbol{\lambda}$ for which that vertex is the unique solution in the LP decoder. However, if the channel is such that the set of possible $\boldsymbol{\lambda}$-vectors is *not* a superset of $\mathbb{R}^n$, then there might be some vertices of the fundamental polytope that will never be chosen by the LP decoder or will at best tie with another vertex. In the case of the BSC, the $\boldsymbol{\lambda}$-vector lies in the set $\{-L, +L\}^n \not\supseteq \mathbb{R}^n$, where

$$\begin{aligned} L &\triangleq \log\bigl(p_{Y|X}(0|0)/p_{Y|X}(0|1)\bigr) \\ &= -\log\bigl(p_{Y|X}(1|0)/p_{Y|X}(1|1)\bigr) \end{aligned}$$

and, therefore, the above-mentioned scenario can happen for the BSC. A similar observation holds for the BEC where the $\boldsymbol{\lambda}$-vector must lie in the set $\{-\infty, 0, +\infty\}^n \not\supseteq \mathbb{R}^n$.

In order to analyze this behavior, we assume that the zero codeword is sent. We known that if the LP decoder does not decide for the zero codeword then there is at least one minimal pseudo-codeword whose cost function is at least as good as the cost function of the zero codeword. This observation is the motivation for the following definition that introduces the *effectiveness* of a minimal pseudo-codeword.

Definition 24: Fix a memoryless binary-input output-symmetric channel and let $\mathcal{L}^{(n)} \subseteq (\mathbb{R} \cup \{\pm\infty\})^n$ be the set of all possible LLR vectors upon sending the all-zero codeword.[24] Moreover, let us fix a parity-check matrix $\boldsymbol{H}$ and let $\mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))$ be the set of minimal pseudo-codewords. A minimal pseudo-codeword $\boldsymbol{\omega} \in \mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))$ is called effective of the first kind for that particular channel if there exists a $\boldsymbol{\lambda} \in \mathcal{L}^{(n)}$ such that $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle < 0$ and $\langle \boldsymbol{\omega}', \boldsymbol{\lambda} \rangle \geqslant 0$ for all $\boldsymbol{\omega}' \in \mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H})) \setminus \{\boldsymbol{\omega}\}$. A minimal pseudo-codeword $\boldsymbol{\omega} \in \mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H}))$ is called effective of the second kind for that particular channel if there exists a $\boldsymbol{\lambda} \in \mathcal{L}^{(n)}$ such that $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \leqslant 0$ and $\langle \boldsymbol{\omega}', \boldsymbol{\lambda} \rangle \geqslant 0$ for all

[22] In the case of ties, we randomly pick one of the vertices in the minimizing set of the LP decoder.

[23] In the case of ties, this function takes on randomly one of the vertices of the minimizing set of the LP decoder.

[24] For the AWGNC we have $\mathcal{L}^{(n)}_{\mathrm{AWGNC}} = \mathbb{R}^n$, for the BSC we have $\mathcal{L}^{(n)}_{\mathrm{BSC}} = \{\pm L\}^n$ for $L \triangleq \log\Bigl((1-\varepsilon)/\epsilon\Bigr) \in \mathbb{R}_{++}$ (we assume that the crossover probability ε fulfills $0 < \varepsilon < 1/2$), and for the BEC we have $\mathcal{L}^{(n)}_{\mathrm{BEC}} = \{0, +\infty\}^n$ (we assume that the erasure probability δ fulfills $0 < \delta < 1$). Please note that there was a slight mistake in [41, Definition 10], i.e., in [41, Definition 10] we forgot to require that the all-zeros codeword was sent. Nevertheless, compared to [41, Definition 10], the sets $\mathcal{L}^{(n)}_{\mathrm{AWGNC}}$ and $\mathcal{L}^{(n)}_{\mathrm{BSC}}$ remain unchanged, whereas for the BEC we have $\{0, +\infty\}^n$ instead of $\{-\infty, 0, +\infty\}^n$.

$\boldsymbol{\omega}' \in \mathcal{M}_{\mathrm{p}}(\mathcal{K}(\boldsymbol{H})) \setminus \{\boldsymbol{\omega}\}$. (Obviously, a minimal pseudo-codeword that is effective of the first kind is also effective of the second kind.) □

Let $\mathcal{L}_{\mathbf{0}}^{(n)} \subseteq \mathcal{L}^{(n)}$ be the set of $\boldsymbol{\lambda}$-vectors in $\mathcal{L}^{(n)}$ where LP decoding decides in favor of the codeword $\mathbf{0}$. From the above definition it follows that a minimal pseudo-codeword "shapes" the set $\mathcal{L}_{\mathbf{0}}^{(n)}$ if and only if it is an effective minimal pseudo-codeword. More precisely, in the case where a minimal pseudo-codeword $\boldsymbol{\omega}$ is effective of the first kind then there exists at least one $\boldsymbol{\lambda} \in \mathcal{L}^{(n)}$ where $\boldsymbol{\omega}$ wins against all other minimal pseudo-codewords (and the zero codeword). Moreover, in the case where $\boldsymbol{\omega}$ is effective of the second kind we are guaranteed that there is at least one $\boldsymbol{\lambda} \in \mathcal{L}^{(n)}$ where $\boldsymbol{\omega}$ is involved in a tie; if and how often $\boldsymbol{\omega}$ wins against all other minimal pseudo-codewords (and the zero codeword) depends on how ties are resolved.[25]

The following subsections deal separately with the AWGNC, the BSC, and the BEC. Given that the sets $\mathcal{L}_{\mathrm{AWGNC}}^{(n)}$, $\mathcal{L}_{\mathrm{BSC}}^{(n)}$, and $\mathcal{L}_{\mathrm{BEC}}^{(n)}$ are rather different, it is not surprising that the effectiveness of the minimal pseudo-codewords will vary quite a bit from channel to channel.

A. AWGNC

We start with the AWGNC where the classification of the effectiveness of minimal pseudo-codewords is very simple, independently of the chosen code.

Theorem 25: For the binary-input AWGNC and any parity-check matrix $\boldsymbol{H}$ all minimal pseudo-codewords of $\mathcal{K}(\boldsymbol{H})$ are effective of the first kind.

Proof: This follows from basic cone properties (cf., e.g., [4]). □

B. BSC

We now turn to the BSC. As the following observations for $\mathrm{PG}(2,q)$-based codes show, for this channel not all minimal pseudo-codewords need to be effective of the first or of the second kind.

Theorem 26: Consider data transmission over a BSC using the code defined by $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$. Then LP decoding can correct any pattern of $\frac{q}{2}$ bit-flips and no pattern of more than q bit flips.

Proof: Because $w_{\mathrm{p}}^{\mathrm{BSC,min}}(\boldsymbol{H}) = q+2$, the BSC pseudo-weight of any pseudo-codeword in $\mathcal{K}(\boldsymbol{H})$ is at least $q+2$. Therefore, LP decoding can correct at least $\left\lfloor \frac{q+2-1}{2} \right\rfloor = \frac{q}{2}$ bit-flips.

Let us now show that LP decoding can correct at most q bit-flips. Remember that a necessary condition for LP decoding to decode a received log-likelihood vector $\boldsymbol{\lambda}$ to the zero codeword is that $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \geqslant 0$ for all $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$.[26] Assume that we are transmitting the zero codeword and that e bit-flips happened. Hence, e components of $\boldsymbol{\lambda}$ are equal to $-L$ and $n-e$ components of $\boldsymbol{\lambda}$ are equal to $+L$. It can easily be checked that the following $\boldsymbol{\omega}$ is in $\mathcal{K}(\boldsymbol{H})$: let $\omega_i \triangleq 1$ if $\lambda_i = -L$ and $\omega_i \triangleq 1/q$ otherwise. For this $\boldsymbol{\omega}$, the condition $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \geqslant 0$ translates into

$$e \cdot (-L) + (n-e) \cdot (1/q) \cdot (+L) \geqslant 0$$

i.e.,

$$e \leqslant \frac{n}{q+1} = \frac{q^2+q+1}{q+1} = q + \frac{1}{q+1}.$$

Because e must be an integer, this inequality turns into the inequality $e \leqslant \left\lfloor q + \frac{1}{q+1} \right\rfloor = q$. □

Observe that the way we constructed the pseudo-codeword $\boldsymbol{\omega}$ in the proof of Theorem 26 can be seen as a generalization of the so-called canonical completion [5], [6], however, instead of assigning values according to the graph distance with respect to a single node, we assign values according to the graph distance with respect to the set of nodes where λ_i is negative. (Note that the Tanner graph of $\boldsymbol{H} = \boldsymbol{H}_{\mathrm{PG}(2,q)}$ has a special property: all variable nodes are at graph distance 2 from each other.) Such a generalization of the canonical completion was also used by Haley and Grant [28] for the analysis of their codes.

Corollary 27: Consider the code defined by $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$. For the BSC, a necessary condition for a minimal pseudo-codeword $\boldsymbol{\omega}$ of $\mathcal{K}(\boldsymbol{H})$ to be effective of the second kind is that $q+2 \leqslant w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}) \leqslant 2q+2$.

Proof: See the Appendix, part B. □

For $q = 4$ it turns out that $\mathcal{K}(\boldsymbol{H}_{\mathrm{PG}(2,4)})$ has minimal pseudo-codewords with BSC pseudo-weight equal to 12. (These minimal pseudo-codewords have type $\boldsymbol{t}$ with $t_2 = 1$, $t_1 = 12$, $t_0 = 8$, and $t_\ell = 0$ otherwise.) Corollary 27 clearly shows that these cannot be effective of the second kind for the BSC, since, for $q = 4$, any effective minimal pseudo-codeword of the second kind must fulfill $6 \leqslant w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}) \leqslant 10$.

Judging from Fig. 1, it also seems—as far as AWGNC and BSC pseudo-weight are comparable—that soft information is quite helpful for the LP decoder when decoding the code $\mathcal{C}_{\mathrm{PG}(2,4)}$ defined by $\boldsymbol{H}_{\mathrm{PG}(2,4)}$.

We finish this subsection on the BSC by noting that the statement in Theorem 26 can be generalized to other classes of codes. Indeed, such a generalization was pursued in [42] and lead to the formulation of upper bounds on the BSC crossover probability threshold for certain families of LDPC codes.

C. BEC

We now discuss effective minimal pseudo-codewords for the BEC; except for the last paragraph, this subsection will make statements about general codes. We start off by noting that because $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \geqslant 0$ for all minimal pseudo-codewords $\boldsymbol{\omega}$ and all $\boldsymbol{\lambda} \in \mathcal{L}_{\mathrm{BEC}}^{(n)}$, no minimal pseudo-codeword can be effective of the first kind.

Theorem 28: Consider data transmission over a BEC using a code defined by some parity-check matrix $\boldsymbol{H}$. Let $\boldsymbol{\omega}$ be a minimal pseudo-codeword such that there is an unscaled pseudo-codeword associated to $\boldsymbol{\omega}$ with at least one odd component. Then there exists a nonzero codeword $\boldsymbol{c}$ such that whenever $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle = 0$ for some $\boldsymbol{\lambda} \in \mathcal{L}_{\mathrm{BEC}}^{(n)}$ then also $\langle \boldsymbol{c}, \boldsymbol{\lambda} \rangle = 0$.

Proof: Let $\boldsymbol{\omega}'$ be an unscaled pseudo-codeword that is a positive multiple of $\boldsymbol{\omega}$. By assumption, we have that at least one

[25] The fact that a minimal pseudo-codeword is effective of the second kind does of course not exclude the possibility that there are also $\boldsymbol{\lambda} \in \mathcal{L}^{(n)}$ where $\boldsymbol{\omega}$ wins (unconditionally) against all other minimal pseudo-codewords (and the zero codeword).

[26] Note that this is usually not a sufficient condition for correct decoding, e.g., in the case where ties are resolved randomly.

component of $\boldsymbol{\omega}'$ is an odd integer. It follows [20], [6] that $\boldsymbol{c} \triangleq \boldsymbol{\omega}' \pmod 2$ is a nonzero codeword. Let $\boldsymbol{\lambda} \in \mathcal{L}_{\mathrm{BEC}}^{(n)}$ be such that $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle = 0$. Because $\boldsymbol{\omega}'$ is a positive multiple of $\boldsymbol{\omega}$ we must have $\langle \boldsymbol{\omega}', \boldsymbol{\lambda} \rangle = 0$ and because $\mathrm{supp}(\boldsymbol{c}) \subseteq \mathrm{supp}(\boldsymbol{\omega}')$ we must have $\langle \boldsymbol{c}, \boldsymbol{\lambda} \rangle = 0$. □

Corollary 29: Consider data transmission over a BEC using a code defined by some parity-check matrix $\boldsymbol{H}$. Under block-wise ML decoding we define a block error to be the event that there is a tie among at least two codewords. Similarly, under LP decoding, we define a block error to be the event that there is a tie among at least two pseudo-codewords. If for all minimal pseudo-codewords there exists an associated pseudo-codeword with at least one odd component then the block-error rate of block-wise ML decoding coincides with the block-error rate of LP decoding. □

By listing all the minimal pseudo-codewords, it can be shown numerically that the condition in Corollary 29 is fulfilled for $\boldsymbol{H} \triangleq \boldsymbol{H}_{\mathrm{PG}(2,q)}$ when $q = 2$ and $q = 4$. It follows that for these two codes block-wise ML and LP decoding yield the same block-error rate (under the above definition of block-error rate). This corroborates the observations made in [43, Fig. 1] for $q = 4$.

X. Conclusion

We have investigated the minimal pseudo-codewords of some codes whose Tanner graphs are derived from projective and Euclidean planes and we have introduced the notion of pseudo-weight spectrum gap for a parity-check matrix, a concept which is certainly worthwhile to be further explored. Although our numerical results are for codes of modest length, to the best of our knowledge, this is the first study that tries to *analytically* quantify the behavior of $\mathrm{PG}(2,q)$- and $\mathrm{EG}(2,q)$-based binary linear codes under LP decoding. Extending these results to somewhat longer codes has the potential to explain many experimental observations made in the past. Moreover, we have obtained a clearer picture on the structure of the minimal pseudo-codewords of the Tanner graphs under investigation, nevertheless, more work is required to get a sufficiently tight characterization of them. Throughout the text, we have also listed some conjectures and statements of open problems; we hope that they stimulate some research toward a better understanding of minimal pseudo-codewords, be it for codes from the code families that we have considered or any other codes. Let us remark that Chertkov and Stepanov [44] have recently proposed a heuristic algorithm for enumerating minimal pseudo-codewords. In their paper, they show some interesting (approximate) spectra of other families of codes than the ones presented here.

In order to classify which minimal pseudo-codewords can be the solution of LP decoding and which ones cannot, we have introduced the notion of the effectiveness of a minimal pseudo-codeword. In the case of the AWGNC, all minimal pseudo-codewords are effective (of the first kind and the second kind); however, we have seen that in the case of non-AWGNC channels there are minimal pseudo-codewords that are not effective, i.e., not effective of the first kind or not even effective of the second kind. Interestingly, for deriving that result, we were able to use the canonical completion, a tool that so far has been very useful for characterizing families of $(w_{\mathrm{col}}, w_{\mathrm{row}})$-regular LDPC codes, with w_{col}, w_{row} bounded when the block length goes to infinity, i.e., code families where the Tanner graph diameter grows with the block length. (This is in contrast to the $\mathrm{PG}(2,q)$-based Tanner graphs which have girth 6 and diameter 3, independently of q).

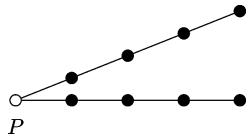

Fig. 6. Part of $\mathrm{PG}(2,q)$ showing the relevant part of a codeword with Hamming weight $2q$. (Here for $q = 4$).

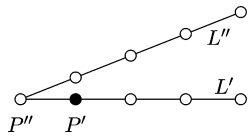

Fig. 7. Part of $\mathrm{PG}(2,q)$ showing the points P' and P'' and the lines L' and L'' that were used in proof of Theorem 13. (Here for $q = 4$).

Appendix

This appendix contains two of the longer proofs of this paper.

A. Proof of Theorem 13

Let $\boldsymbol{s}^{\mathsf{T}} \triangleq \boldsymbol{H} \cdot \boldsymbol{\omega}^{\mathsf{T}}$ (in $\mathbb{R}$) and let $\mathcal{J}_{\mathrm{odd}} \triangleq \{j \in \mathcal{J} \mid s_j \text{ is odd}\}$ be the set of all rows of $\boldsymbol{H}$ where the corresponding entry in $\boldsymbol{s}$ is an odd integer. If $\boldsymbol{\omega}$ were a codeword then all entries of $\boldsymbol{s}$ would be nonnegative even integers. However, because $\boldsymbol{\omega}$ is assumed to be a non-codeword, $|\mathcal{J}_{\mathrm{odd}}| \geqslant 1$. Note moreover that because $\mathrm{supp}(\boldsymbol{\omega})$ is a stopping set, we must have $s_j \neq 1$ for all $j \in \mathcal{J}$.

The proof proceeds as follows. In a first step, we will show that for any non-codeword $\boldsymbol{\omega}$ we must have $|\mathcal{J}_{\mathrm{odd}}| \geqslant q + 1$. In a second step, we will show that this lower bound on $|\mathcal{J}_{\mathrm{odd}}|$ implies the lower bound mentioned in the theorem.

So, let us show that $|\mathcal{J}_{\mathrm{odd}}| \geqslant q + 1$ for any non-codeword $\boldsymbol{\omega}$. The way we will reach this conclusion is based on the following observation: because the rows of $\boldsymbol{H}$ are linearly dependent (over $\mathbb{F}_2$), there must be some dependency on the even-/odd-ness of the components of $\boldsymbol{s}$. In order to find some suitable linearly dependent rows of $\boldsymbol{H}$, we will use some special properties of the projective plane $\mathrm{PG}(2,q)$.

The proof is by contradiction, i.e., if $\mathcal{J}_{\mathrm{odd}}$ has $|\mathcal{J}_{\mathrm{odd}}| \leqslant q$, we will show that there exists a $j \in \mathcal{J}_{\mathrm{odd}}$ such that the value of s_j must be an even integer. To that end, it turns out to be useful to reverse the usual interpretation of the columns and rows of $\boldsymbol{H}$: the columns will correspond to lines and the rows will correspond to points of $\mathrm{PG}(2,q)$. With this, there is a one-to-one relationship between the points of $\mathrm{PG}(2,q)$ and the entries of $\boldsymbol{s}$.

Before we proceed, consider Fig. 6 that shows two lines in $\mathrm{PG}(2,q)$ and the points on them (here for $q = 4$). Letting $\boldsymbol{x} \in \mathbb{R}^n$ be a vector where the entries corresponding to black dots equal 1 and the other entries equal 0, one can easily verify that $\boldsymbol{x} \cdot \boldsymbol{H} = \boldsymbol{0} \pmod 2$, i.e., $\boldsymbol{x}$ is a codeword of the code with parity-check matrix $\boldsymbol{H}^{\mathsf{T}}$.

Now, choose any $j \in \mathcal{J}_{\mathrm{odd}}$, consider Fig. 7, and let P' correspond to the jth row of $\boldsymbol{H}$. Because $|\mathcal{J}_{\mathrm{odd}}| \leqslant q$ (and hence the weaker $|\mathcal{J}_{\mathrm{odd}}| - 1 \leqslant q$), it is possible to choose a line L' through

P' such that all points on it (except for P') have an even s_j. On this line it is then possible to choose (again because $|\mathcal{J}_{\text{odd}}| \leqslant q$) a $P'' \neq P'$ such that there is a line L'' through it such that all points on it have an even s_j. Let $\boldsymbol{x} \in \mathbb{R}^n$ be a vector where the nonzero entries corresponding to the points on L' and L'' (except for P'') equal 1 and the other entries equal 0. Because of the considerations in the preceding paragraph, it is clear that $\boldsymbol{x} \cdot \boldsymbol{H} = \boldsymbol{0} \pmod 2$, i.e., all entries in $\boldsymbol{x} \cdot \boldsymbol{H}$ are even integers. This implies that $\boldsymbol{x} \cdot \boldsymbol{s}^{\mathsf{T}} = \boldsymbol{x} \cdot (\boldsymbol{H} \cdot \boldsymbol{\omega}^{\mathsf{T}}) = (\boldsymbol{x} \cdot \boldsymbol{H}) \cdot \boldsymbol{\omega}^{\mathsf{T}}$ must be an even integer. This is a contradiction; namely, because of the way we have chosen P', L', P'', and L'', the inner product $\boldsymbol{x} \cdot \boldsymbol{s}^{\mathsf{T}}$ must be an odd integer.

Let us now prove that the lower bound $|\mathcal{J}_{\text{odd}}| \geqslant q+1$ on $|\mathcal{J}_{\text{odd}}|$ implies the lower bound on $w_{\text{p}}(\boldsymbol{\omega})$ mentioned in the theorem. Because of the special properties of $\boldsymbol{H}$ we have $\boldsymbol{H}^{\mathsf{T}}\boldsymbol{H} = q\boldsymbol{I} + \boldsymbol{J}$, where $\boldsymbol{I}$ is the identity matrix of size $n \times n$ and where $\boldsymbol{J}$ is the all-ones matrix of size $n \times n$. Then $\|\boldsymbol{s}\|_2^2 = \boldsymbol{s}\boldsymbol{s}^{\mathsf{T}} = \boldsymbol{\omega}\boldsymbol{H}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{\omega}^{\mathsf{T}} = q\boldsymbol{\omega}\boldsymbol{I}\boldsymbol{\omega}^{\mathsf{T}} + \boldsymbol{\omega}\boldsymbol{J}\boldsymbol{\omega}^{\mathsf{T}} = q\|\boldsymbol{\omega}\|_2^2 + \|\boldsymbol{\omega}\|_1^2 = qt_1 + t_1^2$. On the other hand

$$\|\boldsymbol{s}\|_2^2 = \sum_{j \in \mathcal{J}} s_j^2 = \sum_{j \in \mathcal{J}} \left(\sum_{i \in \mathcal{I}_j} \omega_i \right)^2 = \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} \omega_i \sum_{i' \in \mathcal{I}_j} \omega_{i'}.$$

We would like to find a lower bound on $\|\boldsymbol{s}\|_2^2$. If $j \in \mathcal{J} \setminus \mathcal{J}_{\text{odd}}$, then we use $\sum_{i' \in \mathcal{I}_j} \omega_{i'} \geqslant 2\omega_i$, for all i in $\mathcal{I}_j$ (which is implied by (6)), otherwise, we use $\sum_{i' \in \mathcal{I}_j} \omega_{i'} \geqslant 2\omega_i + 1$ for all i in $\mathcal{I}_j$ (which also follows from (6), together with the observation that $s_j = \sum_{i' \in \mathcal{I}_j} \omega_{i'} \geqslant 3$)

$$\begin{aligned}
\|\boldsymbol{s}\|_2^2 &= \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} \omega_i \sum_{i' \in \mathcal{I}_j} \omega_{i'} \geqslant 2 \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} \omega_i^2 + \sum_{j \in \mathcal{J}_{\text{odd}}} \sum_{i \in \mathcal{I}_j} \omega_i \\
&= 2 \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} \omega_i^2 + \sum_{j \in \mathcal{J}_{\text{odd}}} \sum_{i \in \mathcal{I}_j} \omega_i \\
&= 2(q+1)\|\boldsymbol{\omega}\|_2^2 + \sum_{j \in \mathcal{J}_{\text{odd}}} \sum_{i \in \mathcal{I}_j} \omega_i \\
&\geqslant 2(q+1)\|\boldsymbol{\omega}\|_2^2 + 3|\mathcal{J}_{\text{odd}}| \\
&= 2(q+1)t_1 + 3|\mathcal{J}_{\text{odd}}|.
\end{aligned}$$

Combining the above results we obtain $qt_1 + t_1^2 = \|\boldsymbol{s}\|_2^2 \geqslant 2(q+1)t_1 + 3|\mathcal{J}_{\text{odd}}|$, or, equivalently, $t_1^2 - (q+2)t_1 - 3|\mathcal{J}_{\text{odd}}| \geqslant 0$. It follows that $t_1 \geqslant \frac{q}{2} + 1 + \frac{1}{2}\sqrt{(q+2)^2 + 12|\mathcal{J}_{\text{odd}}|}$.[27] Inserting the lower bound $|\mathcal{J}_{\text{odd}}| \geqslant q+1$, we obtain

$$t_1 \geqslant \frac{q}{2} + 1 + \frac{1}{2}\sqrt{q^2 + 16q + 16}.$$

Because $w_{\text{p}}(\boldsymbol{\omega}) = t_1$ and because t_1 is an integer, the final result follows.

We conclude with two remarks.

- More sophisticated considerations might lead to better lower bounds on $w_{\text{p}}(\boldsymbol{\omega})$; however, note that $|\mathcal{J}_{\text{odd}}| \geqslant q+1$ is the best lower bound that can be given on the size of $\mathcal{J}_{\text{odd}}$ without additional information about the set. Namely, if $\mathcal{J}_{\text{odd}}$ happens to be equal to $\mathcal{J}_i$ for some $i \in \mathcal{I}$ then $|\mathcal{J}_{\text{odd}}| = q+1$ and no row in $\boldsymbol{H}$ corresponding to an entry in $\mathcal{J}_{\text{odd}}$ can be expressed as a linear combination of rows corresponding to entries in $\mathcal{J} \setminus \mathcal{J}_{\text{odd}}$. In that case, our technique of exhibiting linear dependent rows of $\boldsymbol{H}$ in order to obtain constraints on the even-/odd-ness of the components of $\boldsymbol{s}$ does not work anymore and we cannot improve the lower bound on the size of the set $\mathcal{J}_{\text{odd}}$.
- The bounding techniques used in the second part of the proof were inspired by the bounding techniques that were used in [24], which in turn were generalizations of [45].

[27] $t_1 \leqslant \frac{q}{2} + 1 - \frac{1}{2}\sqrt{(q+2)^2 + 12|\mathcal{J}_{\text{odd}}|}$ is not possible because we know that $t_1 \geqslant q+2$.

B. Proof of Corollary 27

Let $\boldsymbol{\omega}$ be a minimal pseudo-codeword with $w_{\text{p}}^{\text{BSC}}(\boldsymbol{\omega}) > 2q+2$. The proof is by contradiction, i.e., we will assume that $\boldsymbol{\omega}$ is effective of the second kind and then we will show that for any $\boldsymbol{\lambda} \in \mathcal{L}_{\text{BSC}}^{(n)}$ with $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \leqslant 0$ there exists a minimal pseudo-codeword $\boldsymbol{\omega}'' \neq \boldsymbol{\omega}$ such that $\langle \boldsymbol{\omega}'', \boldsymbol{\lambda} \rangle < 0$.

Assume that we are transmitting the zero codeword and that bit-flips happened at positions $\mathcal{E}$. Hence, $|\mathcal{E}|$ components of $\boldsymbol{\lambda}$ are equal to $-L$ and $n - |\mathcal{E}|$ components of $\boldsymbol{\lambda}$ are equal to $+L$. Assume that $\langle \boldsymbol{\omega}, \boldsymbol{\lambda} \rangle \leqslant 0$. From the definition of the BSC pseudo-weight it follows that $|\mathcal{E}| > q+1$, i.e., $|\mathcal{E}| \geqslant q+2$.

Choose a subpattern $\mathcal{E}' \subset \mathcal{E}$ of bit-flips with $|\mathcal{E}'| = q+1$ and define the corresponding $\boldsymbol{\lambda}'$. From the definition of the BSC pseudo-weight it follows that $\langle \boldsymbol{\omega}, \boldsymbol{\lambda}' \rangle > 0$ (otherwise, $w_{\text{p}}^{\text{BSC}}(\boldsymbol{\omega}) \leqslant 2q+2$).

Similarly to the proof of Theorem 26, we can construct a pseudo-codeword $\boldsymbol{\omega}'$ based on $\mathcal{E}'$ such that $\boldsymbol{\omega}'_i \triangleq 1$ if $i \in \mathcal{E}'$ and $\boldsymbol{\omega}'_i \triangleq 1/q$ otherwise. This pseudo-codeword has the property that

$$\langle \boldsymbol{\omega}', \boldsymbol{\lambda}' \rangle = |\mathcal{E}'| \cdot (-L) + (n - |\mathcal{E}'|) \cdot (1/q) \cdot (+L) = -L < 0.$$

Let $\{\boldsymbol{\omega}^{(\ell)}\}_\ell$ be the set of all minimal pseudo-codewords. Then $\boldsymbol{\omega}' = \sum_\ell \alpha_\ell \boldsymbol{\omega}^{(\ell)}$ for some choice of $\{\alpha_\ell\}_\ell$ where all α_ℓ are nonnegative. Therefore, there exists at least one minimal pseudo-codeword, say $\boldsymbol{\omega}''$, such that $\langle \boldsymbol{\omega}'', \boldsymbol{\lambda}' \rangle < 0$. (Because $\langle \boldsymbol{\omega}, \boldsymbol{\lambda}' \rangle > 0$ and $\langle \boldsymbol{\omega}'', \boldsymbol{\lambda}' \rangle < 0$ it is clear that $\boldsymbol{\omega}'' \neq \boldsymbol{\omega}$.)

Because $\boldsymbol{\omega}'' \geqslant \boldsymbol{0}$, it is easy to see that $\langle \boldsymbol{\omega}'', \boldsymbol{\lambda} \rangle \leqslant \langle \boldsymbol{\omega}'', \boldsymbol{\lambda}' \rangle$. This implies that $\langle \boldsymbol{\omega}'', \boldsymbol{\lambda} \rangle < 0$, which is the promised contradiction.

ACKNOWLEDGMENT

The authors greatly appreciate the reviewers' constructive comments that lead to an improved presentation of the results.

REFERENCES

[1] J. Feldman, "Decoding Error-Correcting Codes Via Linear Programming," Ph.D. dissertation, MIT, Cambridge, MA, 2003.

[2] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.

[3] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[5] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. 3rd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sep. 2003, pp. 75–82.

[6] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," *IEEE Trans. Inf. Theory* [Online]. Available: http://www.arxiv.org/abs/cs.IT/0512078, submitted for publication

[7] P. O. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," in *Proc. Int. Symp. Information Theory and its Applications (ISITA)*, Parma, Italy, Oct. 2004, pp. 991–996.

[8] R. Lucas, M. Fossorier, Y. Kou, and S. Lin, "Iterative decoding of one-step majority logic decodable codes based on belief propagation," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 931–937, Jun. 2000.

[9] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[10] L. M. Batten, *Combinatorics of Finite Geometries, 2nd ed*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[11] J. W. P. Hirschfeld, *Projective Geometries over Finite Fields*, ser. Oxford Mathematical Monographs, 2nd ed. New York: Clarendon/Oxford Univ. Press, 1998.

[12] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inf. Theory*, vol. IT–27, no. 5, pp. 533–547, Sep. 1981.

[13] E. Agrell, "Voronoi regions for binary linear block codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 310–316, Jan. 1996.

[14] M. M. Deza and M. Laurent, *Geometry of Cuts and Metrics*, ser. Algorithms and Combinatorics. Berlin, Germany: Springer-Verlag, 1997, vol. 15.

[15] T. Y. Hwang, "Decoding linear block codes for minimizing word error rate," *IEEE Trans. Inf. Theory*, vol. IT–25, no. 6, pp. 733–737, Nov. 1979.

[16] D. Bertsimas and J. N. Tsitsiklis, *Linear Optimization*. Belmont, MA: Athena Scientific, 1997.

[17] J. L. Massey, "Minimal codewords and secret sharing," in *Proc. 6th Joint Swedish-Russian Int. Workshop on Information Theory*, Mölle, Sweden, Aug. 1993, pp. 276–279.

[18] A. Ashikhmin and A. Barg, "Minimal vectors in linear codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 2010–2017, Sep. 1998.

[19] A. Ashikhmin, A. Barg, G. Cohen, and L. Huguet, "Variations on minimal codewords in linear codes," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (Paris, 1995) (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1995, vol. 948, pp. 96–105.

[20] R. Koetter, W.-C. W. Li, P. O. Vontobel, and J. L. Walker, "Characterizations of pseudo-codewords of LDPC codes," *Adv. Math.*, vol. 213, no. 1, pp. 205–229, Aug. 2007.

[21] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1996.

[22] G. D. Forney, Jr., R. Koetter, F. R. 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)*, B. Marcus and J. Rosenthal, Eds. New York: Springer-Verlag, 2001, vol. 123, pp. 101–112.

[23] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York: North-Holland, 1977.

[24] P. O. Vontobel and R. Koetter, "Lower bounds on the minimum pseudo-weight of linear codes," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun. 2004, p. 70.

[25] S.-T. Xia and F.-W. Fu, "Minimum pseudo-weight and minimum pseudo-codewords of LDPC codes," *IEEE Trans. Inf. Theory* [Online]. Available: http://www.arxiv.org/abs/cs.IT/060605, submitted for publication

[26] G. Cohen and A. Lempel, "Linear intersecting codes," *Discr. Math*, vol. 56, pp. 35–43, 1985.

[27] D. J. C. MacKay and M. C. Davey, "Evaluation of Gallager codes for short block length and high rate applications," in *Codes, Systems, and Graphical Models (Minneapolis, MN, 1999)*, B. Marcus and J. Rosenthal, Eds. New York: Springer-Verlag, 2001, pp. 113–130.

[28] D. Haley and A. Grant, "Improved reversible LDPC codes," in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, Sep. 2005, pp. 1367–1371.

[29] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[30] G. E. Séguin, "A lower bound on the error probability for signals in white Gaussian noise," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 3168–3175, Nov. 1998.

[31] D. Avis, "LRS: A revised implementation of the reverse search vertex enumeration algorithm," in *Polytopes—Combinatorics and Computation*, G. Kalai and G. M. Ziegler, Eds. Basel, Switzerland: Birkhäser Verlag, 2000, pp. 177–198 [Online]. Available: http://cgm.cs.mcgill.ca/~avis/C/lrs.html

[32] N. Kashyap and A. Vardy, "Stopping sets in codes from designs," in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, Japan, Jul. 2003, p. 122 [Online]. Available: http://www.mast.queensu.ca/~nkashyap

[33] M. Wauer, "LDPC Codes Based on Projective Geometries," Master's thesis, Dept. Math. Statist., San Diego State Univ., San Diego, CA, 2005.

[34] R. Smarandache and M. Wauer, "Bounds on the pseudo-weight of minimal pseudo-codewords of projective geometry codes," in *Contemporary Mathematics, Algebra and Its Applications*, D. V. Huynh, S. K. Jain, and S. R. Lopez-Permouth, Eds. Providence, RI: Amer. Math. Soc., Dec. 2006, vol. 419, pp. 285–296 [Online]. Available: http://www.arxiv.org/abs/cs.IT/0510049

[35] C. Kelley and D. Sridhara, "Pseudocodewords of Tanner graphs," *IEEE Trans. Inf. Theory* [Online]. Available: http://www.arxiv.org/abs/cs.IT/0504013, submitted for publication

[36] J. Justesen, T. Høholdt, and J. Hjaltason, "Complete ML decoding for the (73,45) PG code," in *Proc. 43rd Allerton Conf. Communications, Control, and Computing*, Monticello, IL, Sep. 2005, pp. 1076–1086.

[37] J. Justesen, T. Høholdt, and J. Hjaltason, "Iterative list decoding," *IEEE Trans. Inf. Theory*, submitted for publication.

[38] S. Laendner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Proc. 2005 Int. Conf. Wireless Networks, Communications, and Mobile Computing (Wirelesscom 2005)*, Maui, HI, Jun. 2005.

[39] T. Høholdt, J. Justesen, and B. Jonsson, "Euclidean geometry codes, minimum weight words and decodable error-patterns using bit-flipping," in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, Sep. 2005, pp. 2171–2173.

[40] T. Høholdt, Dec. 2005, personal communication.

[41] P. O. Vontobel and R. Smarandache, "On minimal pseudo-codewords of Tanner graphs from projective planes," in *Proc. 43rd Allerton Conf. Communications, Control, and Computing*, Monticello, IL, Sep. 2005, pp. 20–30 [Online]. Available: http://www.arxiv.org/abs/cs.IT/0510043

[42] P. O. Vontobel and R. Koetter, "Bounds on the threshold of linear programming decoding," in *Proc. IEEE Inf. Theory Workshop*, Punta del Este, Uruguay, Mar. 2006, pp. 175–179 [Online]. Available: http://www.arxiv.org/abs/cs.IT/0602087

[43] M. Zhu and K. M. Chugg, "Lower bounds on stopping distance of linear codes and their applications," in *Proc. 43rd Allerton Conf. Communications, Control, and Computing*, Monticello, IL, Sep. 2005, pp. 1778–1787.

[44] M. Chertkov and M. G. Stepanov, "An efficient pseudo-codeword search algorithm for linear programming decoding of LDPC codes," *IEEE Trans. Inf. Theory*, Sep. 2006 [Online]. Available: http://www.arxiv.org/abs/cs.IT/0601113, submitted for publication

[45] R. M. Tanner, "Minimum-distance bounds by graph analysis," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 808–821, Feb. 2001.