The Trapping Redundancy of Linear Block Codes

Stefan Laendner, Thorsten Hehn, Olgica Milenkovic, and Johannes B. Huber

***Abstract*— We generalize the notion of the stopping redundancy in order to study the smallest size of a trapping set in Tanner graphs of linear block codes. In this context, we introduce the notion of the trapping redundancy of a code, which quantifies the relationship between the number of redundant rows in any parity-check matrix of a given code and the size of its smallest trapping set. Trapping sets with certain parameter sizes are known to cause error-floors in the performance curves of iterative belief propagation decoders, and it is therefore important to identify decoding matrices that avoid such sets. Bounds on the trapping redundancy are obtained using probabilistic and constructive methods, and the analysis covers both general and elementary trapping sets. Numerical values for these bounds are computed for the $[2640, 1320]$ Margulis code and the class of projective geometry codes, and compared with some new code-specific trapping set size estimates.**

Index Terms Belief Propagation, LDPC Codes, Margulis Codes, Projective Geometry Codes, Trapping Redundancy, Trapping Sets.

I. Introduction

The performance of linear error-correcting codes (and low-density parity-check (LDPC) codes in particular) under iterative decoding depends on the choice of the parity-check matrix of the code. More precisely, the error rate of a code is influenced by a class of combinatorial entities determined by the choice of the parity-check matrix, such as stopping [1] and trapping sets [2], [3]. Stopping and trapping sets are defined in terms of constraints on the weights of rows in the parity-check matrix induced by subsets of its columns. Certain such restrictions on the weight distributions of the rows can only be satisfied if the parity-check matrix of the code has a sufficiently large number of judiciously chosen rows. Thus, recent work focused on introducing redundant rows into parity-check matrices of a code in order to ensure that the size of their smallest stopping sets are sufficiently large or equal to the minimum distance of the code [4], [5], [6], [7]. Since adding redundant rows to the parity-check matrix increases the decoding complexity of the code, it is important to understand the inherent trade-off between the size of the smallest stopping set and the number of rows in a parity-check matrix. Several ideas for addressing these issues that exploit properties of orthogonal arrays and covering arrays [8] were described in [6], [4], and [9].

The contributions of this work are three-fold. First, we generalize the notion of the stopping redundancy for the case of trapping sets, and term this combinatorial number the *trapping redundancy*. Second, we describe simple probabilistic and deterministic methods for upper-bounding the trapping redundancy of binary linear block codes. Third, we present new analytical techniques for estimating the sizes of small trapping sets in the family of projective geometry (PG) codes and the Margulis $[2640, 1320]$ code, and compare these estimates with the upper bounds.

The paper is organized as follows. Section II provides relevant definitions and introduces the terminology used throughout the paper. Section III contains the main results – probabilistic and constructive upper bounds on the trapping redundancy of codes. Section IV describes the relationship between trapping sets and arcs in PG codes. In the same section, numerical results for the trapping redundancy of the Margulis $[2640, 1320]$ and the family of PG codes are compared with results concerning arcs and elementary trapping sets in the family of Margulis codes. Concluding remarks are given in Section V.

II. Definitions, Background, and Terminology

We start by defining the notion of the restriction of (redundant) parity-check matrices, and then proceed to introduce trapping sets and elementary trapping sets. Based on the notion of the restriction, we state the central definition of the paper, pertaining to the trapping redundancy of a linear block code.

A. Near-Codewords and Trapping Sets

Decoding of LDPC codes is usually performed in an iterative manner, using the suboptimal belief-propagation (BP) algorithm. This decoding approach, also known as message passing, can be seen as the process of exchanging reliability messages along the edges of a bipartite Tanner graph. The incidence matrix of the Tanner graph corresponds to the parity-check matrix used for decoding, so that the code variables index the vertices on the left hand side of the graph, while the parity-check equations index the right hand side vertices of the graph. For a more comprehensive treatment of LDPC codes and iterative decoding, the interested reader is referred to [10], [11], [12].

The error-floor phenomenon of iterative decoders was first described by MacKay and Postol in [2], who observed that the bit error rate curve of the $[2640, 1320]$ Margulis code exhibits a sudden change of slope at signal-to-noise ratios approximately equal to 2.4 dB. This change of slope was attributed to the existence of *near-codewords* in the Tanner

Manuscript received December 30, 2006; revised December 17, 2007 and May 30, 2008. Part of this work was presented at the International Conference on Wireless Networks, Communications, and Mobile Computing (WirelessComm) 2005, Maui, Hawaii, at the IEEE Global Telecommunications Conference (GlobeCom) 2006, San Francisco, California, and at the Conference on Information Sciences and Systems (CISS) 2006, Princeton, New Jersey, USA. This work was supported in part by NSF Grant CCF-0514921 awarded to Olgica Milenkovic, by a research fellowship from the Institute for Information Transmission, University of Erlangen-Nuremberg, Erlangen, Germany, awarded to Stefan Laendner, and by a German Academic Exchange Service (DAAD) fellowship awarded to Thorsten Hehn. Stefan Laendner, Thorsten Hehn, and Johannes B. Huber are with the Institute for Information Transmission at the University of Erlangen-Nuremberg, Erlangen, Germany. Olgica Milenkovic is with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL, USA.

graph of the Margulis code with a parity-check matrix $\boldsymbol{H}$ described in [13]. Near codewords are error vectors $\boldsymbol{y}$ of small weight, with syndromes $\boldsymbol{s_y} \equiv \boldsymbol{Hy}$ that also have small weight. In his seminal paper [3], Richardson analyzed the effect of near-codewords on the performance of various classes of decoders and for a group of channels. He also introduced the notion of *trapping sets* to describe configurations of variable nodes in Tanner graphs of codes that cause failures of specific decoding schemes. There exist many different groups of trapping sets. For example, trapping sets of maximum likelihood decoders are sets of variables containing the supports of *codewords* of the code; trapping sets of iterative decoders used for messages transmitted over the binary erasure channel are stopping sets [1]. For the additive white Gaussian noise (AWGN) channel and BP decoding, no simple characterization of trapping sets is known. Nevertheless, extensive computer simulations revealed that a large number of trapping sets for this channel/decoder combination can be described in a simple and precise setting. Henceforth, we use the notion "trapping set" to refer exclusively to sets of the form described below. To define trapping sets, we first introduce the notion of the restriction of a matrix.

Definition 2.1: For a given $m \times n$ matrix $\boldsymbol{H} = (H_{i,j})$ with $1 \leq i \leq m$, $1 \leq j \leq n$, the restriction of a set of t columns indexed by $j_1, j_2, \ldots, j_t$ is defined as an $m \times t$ sub-matrix of $\boldsymbol{H}$ consisting of the elements $H_{i,j}$, $1 \leq i \leq m$, $\;j = j_1, j_2, \ldots, j_t$.

For a given linear $[n, k, d]$ code $\mathcal{C}$, with parity-check matrix $\boldsymbol{H}$ and corresponding Tanner graph $\mathcal{G}(\boldsymbol{H})$, trapping sets are defined as follows.

Definition 2.2: An (a, b) trapping set $\mathcal{T}(a, b)$ is a collection of a variable nodes for which the subgraph in $\mathcal{G}(\boldsymbol{H})$ induced by $\mathcal{T}(a, b)$ and its neighbors contains $b > 0$ odd-degree check nodes[1]. Equivalently, an (a, b) trapping set $\mathcal{T}(a, b)$ of $\boldsymbol{H}$ is a set of a columns with a restriction that contains b odd-weight rows.

The class of trapping sets that exhibits the strongest influence on the performance of iterative decoders is the class of *elementary* trapping sets.

Definition 2.3: An elementary (a, b) trapping set is a set $\mathcal{T}^{(e)}(a, b)$ of variable nodes for which all check nodes in the subgraph induced by $\mathcal{T}^{(e)}(a, b)$ and its neighbors have either degree one or two, and there are exactly b degree-one check nodes. Alternatively, an elementary (a, b) trapping set is a trapping set for which all non-zero rows in the restriction have either weight one or two, and exactly b rows have weight one.

For a fixed value of the parameter a (or b), the problem of finding the trapping set with smallest parameter b (or a) in a given parity-check matrix is NP-hard, and NP-hard to approximate [14]. This makes a general and complete characterization of the trapping set sizes and trapping redundancy prohibitively complex. We therefore focus our attention on deriving upper bounds on the trapping redundancy with set sizes restricted to $a \leq d - 1$ only, where d denotes the minimum distance of the code, and in particular, elementary trapping sets [15]. This is motivated by recent studies that suggest that trapping sets most detrimental to the code performance are elementary, and that they have small (a, b) parameters, usually such that $b < a < d$ [3], [16].

[1]The case $b = 0$ corresponds to codewords. Henceforth, we consider the case $b > 0$ only.

B. Redundant Parity-Check Matrices

For every linear $[n, k, d]$ code $\mathcal{C}$, there exist many choices for parity-check matrices, although for iterative decoding not all of them may be adequate. This is due to the fact that some parity-check matrices have irregular row- and column-weights and that they contain a large number of stopping and trapping sets [6]. In order to mitigate this problem, one may resort to the use of *redundant* parity-check matrices, i.e., matrices that contain more than $n - k$ rows, although they have row-rank equal to $n - k$. Examples of the use of redundant parity-check matrices for signaling over the binary erasure channel can be found in [6], [4], [17], [18].

Henceforth, we use the phrase *redundant parity-check matrix* to refer to a parity-check matrix with row-rank $n - k$ that has more than $n-k$ rows. A redundant parity-check matrix contains rows that are linear combination of other rows that represent a basis of the dual code $\mathcal{C}^{\perp}$. For a fixed basis, rows of this form are referred to as redundant rows. On the other hand, a parity-check matrix of full row-rank and dimension $(n-k) \times n$ is simply termed a *parity-check matrix*. Redundant parity-check matrices are used to impose specific constraints on the structure of their corresponding Tanner graphs. Even one judiciously chosen redundant row can be used to lower the error floor of the Margulis code. This is achieved in terms of rendering the structure of a selected small trapping set so as to increase the number of its corresponding unsatisfied parity-check equations. From an application point of view, it is of interest to identify one or a few redundant rows that can be added to the parity-check matrix in order to eliminate a trapping set causing a special instant of decoding failure. Note that one or a few redundant rows of low weight do not significantly alter the performance of a code in the waterfall region, although they may have a significant bearing on its performance in the error-floor region. In what follows, we consider the more general theoretical problem of determining the smallest number of redundant rows needed to *simultaneously* eliminate the negative effect of classes of trapping sets on the performance of iterative decoders. In this context, our results can be seen as a generalization of the findings in [6] for the case of trapping sets.

An analytical study of the trapping redundancy is presented in the following section.

III. The Trapping Redundancy: A Probabilistic Approach

We investigate the fundamental theoretical trade-offs between the number of rows in a redundant parity-check matrix of a code and the size of its smallest trapping set with a given set of parameters.

A. Definition and Bounds of the Trapping Redundancy

In all our subsequent derivations, we make use of the following definition.

Definition 3.1: ([8, p. 5]) An orthogonal array $\mathcal{A}$ of strength t is an array of dimensions $m \times n$ such that every $m \times t$ subarray contains *each* possible t-tuple as rows *the same number of times*.

The codewords of an $[n, k, d]$ linear code $\mathcal{C}$ form an orthogonal array of dimension $2^k \times n$ and strength $d^\perp - 1$, where $d^\perp$ denotes the dual distance of $\mathcal{C}$. Note that if $\mathcal{A}$ is an array of strength t, then $\mathcal{A}$ is also an orthogonal array of strength s, for all integers $s < t$.

Let $\theta_H(a, b)$ denote the number of (a, b) trapping sets in the parity-check matrix $\boldsymbol{H}$. We have the following result for $\theta_H(a, b)$ corresponding to a matrix $\boldsymbol{H}$ that consists of *all* codewords of the dual code.

Proposition 3.2: Let $\boldsymbol{H}$ consist of all 2^{n-k} codewords of the dual code of an $[n, k, d]$ linear code $\mathcal{C}$, for $n - k \geq 1$. Then $\theta_H(a, b) = 0$ for all pairs (a, b) such that $1 \leq a \leq d - 1$, and $b \neq 2^{n-k-1}$.

Proposition 3.2 shows that a parity-check matrix that consists of all codewords of the dual code cannot contain trapping sets with $1 \leq a \leq d-1$ variables with less than or more than 2^{n-k-1} checks connected to them an odd number of times. This is a direct consequence of the fact that $\boldsymbol{H}$ in this case represents an orthogonal array, so that each restriction of $1 \leq a \leq d - 1$ columns of $\boldsymbol{H}$ contains each vector of length a the same number of times. Consequently, there are 2^{n-k-1} rows in the restriction of the a columns that have even weight and 2^{n-k-1} rows that have odd weight.

However, it is of much larger importance to determine if there exist parity-check matrices with a number of rows significantly smaller than 2^{n-k} that are free of trapping sets with fixed parameters (a, s), for all $1 \leq s < b$. For this purpose, we introduce the notion of the (a, b) trapping redundancy of a code.

Definition 3.3: The (a, b) *trapping redundancy* $T_{a,b}(\mathcal{C})$ of an $[n, k, d]$ linear code $\mathcal{C}$ is the smallest number of rows m of any (redundant) parity-check matrix which does not contain trapping sets with parameters (a, s), $1 \leq s < b$. Similarly, the smallest number of rows $T^{(e)}_{a,b}(\mathcal{C})$ in a (redundant) parity-check matrix of $\mathcal{C}$ avoiding elementary (a, s) trapping sets with $1 \leq s < b$ is referred to as the (a, b) *elementary trapping redundancy* of $\mathcal{C}$.

Theorem 3.4: Let $\mathcal{C}$ be an $[n, k, d]$ code and $\mathcal{C}^\perp$ its dual. Use $\mathcal{M}_{\mathcal{C}}(m)$ to denote the ensemble of all $m \times n$ matrices with rows chosen independently and at random, with replacement, from the set of 2^{n-k} codewords of $\mathcal{C}^\perp$. Furthermore, let $1 \leq a \leq \lfloor (d-1)/2 \rfloor$ be fixed, let $\Theta(a, b)$ be the number of trapping sets with parameters (a, s), $0 \leq s < b$, $b \leq m$, in a randomly chosen matrix from $\mathcal{M}_{\mathcal{C}}(m)$, and let e denote the base of the natural logarithm. If

$$\mathrm{e} \cdot \left(\binom{n}{a} - \binom{n-a}{a} \right) \left(\frac{1}{2} \right)^m \sum_{j=0}^{b-1} \binom{m}{j} \leq 1, \qquad (1)$$

then $P\{\Theta(a, b) = 0\} > 0$. Consequently, if m satisfies (1), then there exists a parity-check matrix of $\mathcal{C}$ with not more than $m + n - k - 1$ rows that does not contain any (a, s) trapping sets with $1 \leq s < b$.

Note that $m + n - k - 1$, for any m satisfying (1), represents an upper bound on the trapping redundancy of the code, i.e. $T_{a,b}(\mathcal{C}) \leq m + n - k - 1$.

Proof: The proof of the claimed result is based on Lovász Local Lemma (LLL), stated below following the exposition of [19].

Lemma 3.5: Let $E_1, E_2, \ldots, E_N$ be a set of events in an arbitrary probability space. Suppose that each event E_i is independent of all other events E_j, $1 \leq i, j \leq N$, except for at most τ of them, and that $P\{E_i\} \leq p$ for all $1 \leq i \leq N$. If

$$\mathrm{e}\, p\, (\tau + 1) \leq 1, \qquad (2)$$

then $P\{\bigcap_{i=1}^N \overline{E}_i\} > 0$.

Let E_i be the event that the restriction of the i-th collection (out of $\binom{n}{a}$) of a columns from a randomly chosen matrix in $\mathcal{M}_{\mathcal{C}}(m)$ contains fewer than b odd rows. Then

$$P\{E_i\} = \left(\frac{1}{2} \right)^m \sum_{j=0}^{b-1} \binom{m}{j}. \qquad (3)$$

Equation (3) follows from the fact that the codewords of the dual code form an orthogonal array of strength $d - 1$, and that therefore even and odd weight rows in the restriction are equally likely. This is true independent of the choice of a, provided that $1 \leq a \leq \lfloor (d-1)/2 \rfloor$. The orthogonal array property of strength $d-1$ implies that all row-vectors of length up to $d-1$ are equally likely; henceforth, restricting the number of columns a to lie in the range $1 \leq a \leq \lfloor (d-1)/2 \rfloor$ ensures that two events E_i and E_j, $i \neq j$, are independent as long as their corresponding sets of column indices are disjoint.

In the above setting, $P\{\bigcap \overline{E}_i\}$ denotes the probability that the randomly chosen matrix is free of trapping sets with parameters (a, s), for all $0 \leq s < b$. In order to complete the proof, it suffices to observe that the following relationship holds for the dependence number τ of the events E_i,

$$\begin{aligned} \tau + 1 &= \sum_{l=1}^{a-1} \binom{a}{l} \binom{n-a}{a-l} + 1 \\ &= \sum_{l=0}^{a} \binom{a}{l} \binom{n-a}{a-l} - \binom{n-a}{a} = \binom{n}{a} - \binom{n-a}{a}. \end{aligned} \qquad (4)$$

Expression (4) is a consequence of the fact that two collections of a columns are "dependent" if and only if they share at least one column. In other words, if one collection of a columns is fixed, another collection of the same size is deemed independent from it if its columns are chosen from the remaining set of $n-a$ columns. The last line of (4) follows from the Vandermonde convolution formula, which asserts that

$$\sum_l \binom{r}{t+l} \binom{s}{u-l} = \binom{r+s}{t+u},$$

where r, t, s, u denote non-negative integers.

In the worst case, additional $n - k - 1$ rows may be needed to make the randomly chosen matrix have full row-rank $n - k$. This is due to the fact that a matrix containing no $(a, 0)$ trapping sets must have rank at least one.

Note that appending $n-k-1$ additional rows to the selected set of m rows can only increase the number of odd-weight rows in each restriction, and hence cannot reduce the value of the parameter b. This completes the proof of the theorem. ■

Although Theorem 3.4 ensures the existence of at least one matrix in $\mathcal{M}_{\mathcal{C}}(m)$ that is free of trapping sets with given parameters (a, b), the actual probability of selecting such a matrix may be very small. It is therefore of interest to identify values of the parameter m for which the probability of drawing a matrix of the desired form from the ensemble $\mathcal{M}_{\mathcal{C}}(m)$ is close to one.

Theorem 3.6: For a linear code $\mathcal{C}$, let $\Theta(a, b)$ be the number of trapping sets with parameters (a, s), $1 \leq a \leq \lfloor (d-1)/2 \rfloor$, $0 \leq s < b \leq m$, that exist in an arbitrarily chosen $m \times n$ array from the $\mathcal{M}_{\mathcal{C}}(m)$ ensemble. If

$$\left(\frac{1}{2}\right)^m \sum_{j=0}^{b-1} \binom{m}{j} \leq \frac{\epsilon}{\binom{n}{a}} \left(1 - \frac{\epsilon}{\binom{n}{a}}\right)^{\tau}, \tag{5}$$

then $P\{\Theta(a, b) = 0\} > 1 - \epsilon$, where τ is given by (4), and where $0 < \epsilon < 1$ is a real number.

Consequently, if m satisfies (5), then for small values of ϵ one can find with high probability a (redundant) parity-check matrix for $\mathcal{C}$ with not more than $m + n - k - 1$ rows that does not contain any (a, s) trapping sets with $1 \leq s < b$.

Proof: The result in (5) is obtained from the high-probability variation of LLL [19], stated below.

Lemma 3.7: Let $E_1, E_2, \ldots, E_N$ be a set of events in an arbitrary probability space, and let $0 < \epsilon < 1$. Suppose that each event E_i is independent of all other events E_j, except for at most τ of them. If

$$P\{E_i\} \leq \frac{\epsilon}{N} \left(1 - \frac{\epsilon}{N}\right)^{\tau}, \ 1 \leq i \leq N, \tag{6}$$

then $P\{\bigcap_{i=1}^{N} \overline{E}_i\} > 1 - \epsilon$.

To prove the theorem, let E_i, $1 \leq i \leq N$, denote the event that the i-th collection of a columns contains $0 \leq s < b$ rows of odd weight. Replace the expression for $P\{E_i\}$ in (6) by the right-hand side of (3) and use the formula for τ stated in (4). ∎

We derive next upper bounds on the elementary trapping redundancy of linear block codes.

Theorem 3.8: Let $\mathcal{C}$ be an $[n, k, d]$ code and $\mathcal{C}^{\perp}$ its dual. Let $\mathcal{M}_{\mathcal{C}}(m)$ be the ensemble of all $m \times n$ matrices with rows chosen independently and at random, with replacement, from the set of 2^{n-k} codewords of $\mathcal{C}^{\perp}$. Furthermore, let $1 \leq a \leq \lfloor (d-1)/2 \rfloor$ be fixed and let $\Theta_e(a, b)$ be the number of elementary trapping sets with parameters (a, s), $0 \leq s < b$, in a randomly chosen matrix of $\mathcal{M}_{\mathcal{C}}(m)$. If

$$\mathrm{e}\left(\binom{n}{a} - \binom{n-a}{a}\right) \cdot \frac{1}{2^{(a+1)\cdot m}} \cdot \sum_{j=0}^{b-1} \left[\binom{m}{j} 2^j a^j \cdot \left(a^2 - a + 2\right)^{m-j}\right] \leq 1, \tag{7}$$

then $P\{\Theta_e(a, b) = 0\} > 0$. Consequently, if m satisfies (7), then there exists a parity-check matrix of $\mathcal{C}$ with no more than $m + n - k - 1$ rows that does not contain any elementary (a, s) trapping sets with $1 \leq s < b$.

Note that $m + n - k - 1$, with any m satisfying (7) represents an upper bound on the elementary trapping redundancy, i.e. $T_{a,b}^{(e)} \leq m + n - k - 1$.

Proof: The proof follows the proof of Theorem 3.4. Let E_i be the event that the restriction of the i-th collection of a columns from a randomly chosen matrix in $\mathcal{M}_{\mathcal{C}}(m)$ contains only rows of weight at most two. Among these rows, fewer than b rows are required to have weight one.

Let W_ω denote the number of rows of weight ω in the restriction of a columns. Then

$$\begin{aligned} P\{E_i\} &= \sum_{j=0}^{b-1} P\{W_1 = j \ , \ (W_0 + W_2) = (m - j)\} \\ &= \sum_{j=0}^{b-1} \left[\binom{m}{j} \left(\frac{a}{2^a}\right)^j \cdot \left(\frac{a^2 - a + 2}{2^{a+1}}\right)^{m-j}\right] \\ &= \frac{1}{2^{(a+1)\cdot m}} \sum_{j=0}^{b-1} \left[\binom{m}{j} 2^j a^j \left(a^2 - a + 2\right)^{m-j}\right], \end{aligned} \tag{8}$$

where the second equation is a consequence of the fact that

$$\begin{aligned} &P\{W_1 = j \ , \ (W_0 + W_2) = (m - j)\} \\ &= \binom{m}{j} \left(\frac{\binom{a}{1}}{2^a}\right)^j \sum_{\ell=0}^{m-j} \binom{m-j}{\ell} \left(\frac{1}{2^a}\right)^{\ell} \left(\frac{\binom{a}{2}}{2^a}\right)^{m-j-\ell}. \end{aligned}$$

Equation (8) follows from the observation that the codewords of the dual code form an orthogonal array of strength $d - 1$, and that therefore all 1-, $\binom{a}{1}$-, and $\binom{a}{2}$-collections of rows of weight 0, 1, and 2, are equally likely, respectively. ∎

Similarly, based on the high-probability variation of LLL, one can derive upper bounds on the number of rows needed to guarantee that a randomly chosen matrix has no elementary trapping sets with a given set of parameters with probability at least $1-\epsilon$. The following theorem is an analogue of Theorem 3.6 for the case of elementary trapping sets, i.e. with $P\{E_i\}$ defined by (8).

Theorem 3.9: For a linear code $\mathcal{C}$, let $\Theta_e(a, b)$ be the number of elementary trapping sets with parameters (a, s), $1 \leq a \leq \lfloor (d-1)/2 \rfloor$, $1 \leq s < b \leq m$, in an $m \times n$ array from the $\mathcal{M}_{\mathcal{C}}(m)$ ensemble. If

$$\frac{1}{2^{(a+1)\cdot m}} \sum_{j=0}^{b-1} \left[\binom{m}{j} 2^j a^j \cdot \left(a^2 - a + 2\right)^{m-j}\right] \leq \frac{\epsilon}{\binom{n}{a}} \left(1 - \frac{\epsilon}{\binom{n}{a}}\right)^{\tau}, \tag{9}$$

where τ is given in (4) and $0 < \epsilon < 1$, then $P\{\Theta_e(a, b) = 0\} > 1 - \epsilon$.

For m satisfying (9) and small values of ϵ, every randomly constructed parity-check matrix of $\mathcal{C}$ with no more than $m + n - k - 1$ rows does not contain elementary (a, s) trapping sets with $1 \leq s < b$.

If E_i, $1 \leq i \leq N$, is used to denote the event that the i-th collection of a columns contains only rows of weight at most two, and less than b rows of weight one, then the result represents a straightforward application of the high-probability variation of LLL. The proof is therefore omitted.

B. The Trapping Redundancy: A Constructive Approach

The problem of finding the trapping redundancy of a linear block code can also be addressed in a deterministic manner, by invoking arguments similar to those used for upper bounding the stopping redundancy of codes. The results of this analysis are summarized in the theorem below, for the case that the parameter a of the trapping set is bounded from above by $d-1$.

Theorem 3.10: Let $\mathcal{C}$ be an $[n,k,d]$ linear code. Fix the parameter $a \leq d-1$, and let $r = n-k > 2$.
Then

$$T_{a,b}(\mathcal{C}) \leq \sum_{i=1}^{t} \binom{r}{i}$$

where $b \geq 2^{a-1} - \sum_{j=t+1}^{a} \binom{r}{j}$ and $t \leq a$,
i.e.

$$T_{a,b}(\mathcal{C}) \leq \sum_{i=1}^{a} \binom{r}{i} + b - 2^{a-1}. \tag{10}$$

Furthermore, the smallest number of rows in a (redundant) parity-check matrix avoiding elementary trapping sets with parameters (a,s), $s < b$, is upper bounded by

$$T^{(e)}_{a,b}(\mathcal{C}) \leq \sum_{i=1}^{a} \binom{r}{i}. \tag{11}$$

Note that both claims in Theorem 3.10 also hold for all trapping sets with parameters (t,b), where $t \leq a$.

Proof: Let $\boldsymbol{H}$ be an arbitrary parity-check matrix of the code $\mathcal{C}$ of full row rank $r = n-k$. Consider the restriction $\boldsymbol{H}_a$ of $\boldsymbol{H}$ on an arbitrary subset of a of its columns. Due to the fact that $a \leq d-1$, these columns are linearly independent, so that the row-rank of $\boldsymbol{H}_a$ must be a. Hence, there exist a rows in $\boldsymbol{H}_a$ that form a basis for $\mathbb{F}_2^a$.

From $\boldsymbol{H}$, form a new redundant parity-check matrix $\boldsymbol{H}'$ by adding all linear combinations of at least two, but not more than $t \leq a$ rows of $\boldsymbol{H}$, where t will be determined later. The total number of rows in $\boldsymbol{H}'$ in this case equals the right hand side of the expression in (10). Since adding all possible linear combinations of not more than a rows of $\boldsymbol{H}$ to $\boldsymbol{H}'$ would ensure that $b \geq 2^{a-1}$, leaving out sums involving $t+1, ..., a$ rows can reduce the number of odd-weight rows in $\boldsymbol{H}'_a$ by at most $\sum_{j=t+1}^{a} \binom{r}{j}$. This completes the proof of the first claim. The proof of the second claim represents a straightforward extension of the results in [6], and is therefore omitted. ∎

Simple inspection reveals that the bounds in (10) are loose when compared to the random bounds described in the previous section.

If $a \leq d-1$ and $b \leq r = n-k$, (10) can be substantially tightened. In this case, it reads as

$$T_{(a,b)}(\mathcal{C}) \leq r + \binom{r}{2} = \frac{r(r+1)}{2},$$

for general trapping sets, and

$$T_{(a,b)}(\mathcal{C}) \leq b \cdot r$$

for elementary trapping sets.

Let ℓ denote the number of odd-weight rows in an arbitrary restriction $\boldsymbol{H}_a$ of a columns on a parity-check matrix $\boldsymbol{H}$. Since $\boldsymbol{H}$ has full rank, $\boldsymbol{H}_a$ contains at least one odd-weight row and therefore $1 \leq \ell \leq r = n-k$ holds. It is now easy to show that adding all linear combinations of two rows to the parity-check matrix $\boldsymbol{H}$ ensures that $b \geq r$ holds for all a-sets of columns. For $1 \leq \ell \leq r$, there are $\ell \cdot (r-\ell)$ odd-weight rows among all $\binom{r}{2}$ linear combinations of pairs of rows. Adding those linear combinations to the parity-check matrix brings the number of odd-weight rows to $b = \ell + \ell \cdot (r-\ell) = \ell \cdot (r-\ell+1) \geq r$. This follows from the simple observation that the weight of the sum of one odd and one even weight word is always odd. Therefore, to avoid (a,b) trapping sets with $a \leq d-1$, $b < n-k$, at most $r + \binom{r}{2} = \frac{r(r+1)}{2}$ rows suffice.

As a final remark, note that in many applications, decoding failure caused by trapping sets can only be detected upon completion of the decoding process. In this case, one can choose to add only a small number of judiciously chosen redundant rows to the parity-check matrix of the code in order to eliminate the influence of *one particular trapping set*. How this can be accomplished is illustrated on the example of the Margulis $[2640, 1320]$ code, in Section IV-A.

C. Asymptotic Formulas for the Trapping Redundancy

Although there is no explicit formula for m as defined by Equations (1) and (5) that holds for all possible parameter values a and b, such a formula can be found in the asymptotic regime ($m, n \to \infty$, $a = O(m)$, $b = O(m)$), by using the following results from [20, p. 240] and [21].

Let

$$A_m = \sum_{0 \leq i \leq \lambda m} \binom{m}{i},$$

where $0 \leq \lambda \leq 1$, and $b = \lfloor \lambda\, m \rfloor + 1$. Since small values for the parameter b are of special interest, assume that $\lambda < 1/2$. In this case we have

$$A_m \simeq \binom{m}{\lfloor \lambda m \rfloor} \cdot \frac{1}{1 - \frac{\lambda}{1-\lambda}},$$

where the notation $c_m \simeq b_m$ describes the following relationship between two functions c_m and b_m of m: $\lim_{m\to\infty} c_m / b_m = 1$.

For $b < m/2 + 1$, with $b = \lfloor \lambda m \rfloor + 1$, and $\lambda < 1/2$, (1) reduces to

$$\mathrm{e} \cdot \left(\binom{n}{a} - \binom{n-a}{a} \right) \cdot \left(\frac{1}{2} \right)^m \cdot \binom{m}{\lfloor \lambda m \rfloor} \cdot \frac{1}{1 - \frac{\lambda}{1-\lambda}} \lesssim 1.$$

By invoking the well known asymptotic formula

$$\mathrm{ld} \binom{m}{\lfloor \lambda m \rfloor} \simeq m\, \mathrm{H}_2 \left(\frac{\lfloor \lambda m \rfloor}{m} \right), \tag{12}$$

where $\mathrm{H}_2(\cdot)$ denotes Shannon's binary entropy function, and $\mathrm{ld}(\cdot)$ represents the logarithm with base two, it follows that $m \leq m'$ with

$$m' \simeq \frac{\mathrm{ld}\left(\mathrm{e} \cdot \left(\binom{n}{a} - \binom{n-a}{a}\right)\right) + \mathrm{ld}\left(\frac{1}{1-\frac{\lambda}{1-\lambda}}\right)}{1 - \mathrm{H}_2\left(\frac{\lfloor \lambda m \rfloor}{m}\right)}. \tag{13}$$

Also, for $b < m/2+1$, with $b = \lfloor \lambda m \rfloor + 1$, and $\lambda < 1/2$, the high-probability variation of LLL given in (5) reduces to

$$\left(\frac{1}{2}\right)^m \cdot \binom{m}{\lfloor \lambda m \rfloor} \cdot \frac{1}{1-\frac{\lambda}{1-\lambda}} \lesssim \frac{\epsilon}{\binom{n}{a}} \cdot \left(1-\frac{\epsilon}{\binom{n}{a}}\right)^{\left(\binom{n}{a}-\binom{n-a}{a}-1\right)}. \quad (14)$$

Using (12) once again results in an upper bound on the number of rows m in a parity-check matrix free of (a,s) trapping sets, $0 \le s < b$, i.e. $m \le m'$ with

$$m' \simeq \frac{1}{H_2\left(\frac{\lfloor \lambda m \rfloor}{m}\right)-1} \left[\mathrm{ld}\left(1-\frac{\lambda}{1-\lambda}\right) + \mathrm{ld}\left(\frac{\epsilon}{\binom{n}{a}}\right) + \left(\binom{n}{a}-\binom{n-a}{a}-1\right) \cdot \mathrm{ld}\left(1-\frac{\epsilon}{\binom{n}{a}}\right)\right]. \quad (15)$$

Note that for sufficiently large m the right-hand side is not dependent on m as $\frac{\lfloor \lambda m \rfloor}{m} \simeq \lambda$.

D. Asymptotic Formulas for the Elementary Trapping Redundancy

Likewise, it is also of interest to find an explicit formula for m given by (7) for the case of elementary trapping sets. To this end, we use the following asymptotic result taken from [21], stating that

$$\sum_{k=N}^{rN} \binom{rN}{k} p^k \, q^{rN-k} \simeq \phi(p^{-1}) \binom{rN}{N} p^N \, q^{rN-N}, \quad (16)$$

where $p, q > 0$, $p+q=1$, $r > 1$, N is a positive integer, and $\phi(y) = \frac{y-1}{y-r}$. By observing that the summands in (7) can be rewritten as

$$\sum_{j=0}^{b-1} \binom{m}{j} 2^j \, a^j \, (a^2-a+2)^{m-j} = (a^2+a+2)^m \cdot \sum_{u=m-b+1}^{m} \binom{m}{u} \left(\frac{2a}{a^2+a+2}\right)^{m-u} \left(\frac{a^2-a+2}{a^2+a+2}\right)^u,$$

it follows from substituting $m = rN$, $p = \frac{a^2-a+2}{a^2+a+2}$, and $q = \frac{2a}{a^2+a+2}$ that

$$\sum_{j=0}^{b-1} \binom{m}{j} 2^j \, a^j \, (a^2-a+2)^{m-j} \simeq \phi\left(\frac{a^2+a+2}{a^2-a+2}\right) \cdot \binom{m}{m-b+1} \left(\frac{2a}{a^2-a+2}\right)^{b-1} \left(a^2-a+2\right)^m.$$

Using the above expression, (7) reduces to

$$\mathrm{e} \cdot \left(\binom{n}{a}-\binom{n-a}{a}\right) \cdot \phi\left(\frac{a^2+a+2}{a^2-a+2}\right) \cdot \binom{m}{m-b+1} \left(\frac{2a}{a^2-a+2}\right)^{b-1} \left(\frac{a^2-a+2}{2^{a+1}}\right)^m \lesssim 1.$$

This leads to a bound $m \le m'$ with

$$m' \simeq \frac{-\mathrm{ld}\left(\mathrm{e}\left(\binom{n}{a}-\binom{n-a}{a}\right) \cdot \phi\left(\frac{a^2+a+2}{a^2-a+2}\right)\right)}{H_2\left(\frac{\lfloor \lambda m \rfloor}{m}\right) + \mathrm{ld}(a^2-a+2) - (a+1)} - \frac{(b-1) \cdot \mathrm{ld}\left(\frac{2a}{a^2-a+2}\right)}{H_2\left(\frac{\lfloor \lambda m \rfloor}{m}\right) + \mathrm{ld}(a^2-a+2) - (a+1)}, \quad (17)$$

where we used (12) to rewrite the right hand side of the above expression. Similarly, for the high-probability variation of LLL and for elementary trapping sets, we obtain as a consequence of (9) the bound $m \le m'$ with

$$m' \simeq \frac{\mathrm{ld}\left(\frac{\epsilon}{\binom{n}{a}}\right) + \left(\binom{n}{a}-\binom{n-a}{a}-1\right) \cdot \mathrm{ld}\left(1-\frac{\epsilon}{\binom{n}{a}}\right)}{H_2\left(\frac{\lfloor \lambda m \rfloor}{m}\right) + \mathrm{ld}(a^2-a+2) - (a+1)} - \frac{\mathrm{ld}\left(\phi\left(\frac{a^2+a+2}{a^2-a+2}\right)\right) + (b-1) \cdot \mathrm{ld}\left(\frac{2a}{a^2-a+2}\right)}{H_2\left(\frac{\lfloor \lambda m \rfloor}{m}\right) + \mathrm{ld}(a^2-a+2) - (a+1)}. \quad (18)$$

Based on the results of the previous sections, it is also straightforward to see that the asymptotic formula for the trapping redundancy of Theorem 3.10 is of the form

$$T_{a,b}(\mathcal{C}) \lesssim \binom{r}{\lfloor \alpha \, r \rfloor} \frac{1}{1-\frac{\alpha}{1-\alpha}} + b - 2^{\lfloor \alpha \, r \rfloor - 1}, \quad (19)$$

where $a = \lfloor \alpha \, r \rfloor$.

Remark: Note that the matrices from the ensemble $\mathcal{M}_{\mathcal{C}}(m)$, for large m, may have highly non-uniform row and column weights. The variable- and check-node degrees of their corresponding Tanner graphs may be very large, leading to the emergence of short cycles. It is therefore important to compare the derived bounds with some benchmark values, the latter corresponding to redundant matrices that are known to have a small number of redundant rows, no short cycles, as well as no small trapping sets. Two such examples, including the aforementioned Margulis and projective geometry codes, are discussed in the next section. Other families of codes, such as codes based on Latin squares and designs, are analyzed in more detail in the companion paper [22].

IV. Trapping Redundancy: Analytical Comparisons

We perform next a numerical study of the probabilistic upper bounds derived in Section III for the Margulis $[2640, 1320]$ code and the class of projective geometry codes. The goal of the comparative study is to both assess the tightness of the bounds of Section III and to demonstrate that structured LDPC codes with redundant parity-check matrices can avoid small trapping sets in their Tanner graphs.

Note that the presented results only capture the trade-off between the smallest size of general and elementary trapping sets and the number of rows in the corresponding (redundant) parity-check matrix, without taking into consideration other important matrix properties such as variable and check nodes degree, girth, and cycle length distribution.

A. The $[2640, 1320]$ Margulis Code

Numerical values of the trapping redundancy derived in Section III for the $[2640, 1320]$ Margulis code are listed in Tables I and II. The labels *LLL (std), LLL (hp)* refer to the bounds based on LLL in standard form and its high-probability variation, respectively. The symbol m denotes the number of rows required by the LLL approach, while $\hat{m}$ refers to the number of rows of a redundant, rank $n-k$ parity-check matrix, which is an upper bound on the trapping redundancy, $T_{a,b}(\mathcal{C}) \leq \hat{m}$. As the exact minimum distance is not known for this code, a method [23] for approximating the minimum distance, proposed in [24], was used instead. The estimate at hand is $d \approx 40$, and we restrict our attention to values of the parameter a strictly (and significantly) smaller than $d/2 = 20$.

The full-rank 1320×2640 parity-check matrix $\boldsymbol{H}$ of the Margulis code, constructed in the standard manner [25], contains no cycles of length less than eight, but includes a large number of $(12, 4)$ and $(14, 4)$ elementary trapping sets [3], [16]. The LLL-based bounds reveal that there exists a matrix of full rank with at most 1336 rows that does not contain elementary $(14, s)$ trapping sets, with $s < 5$, and that adding at most 19 additional rows ensures that the matrix does not contain elementary $(12, 4)$ trapping sets either.

It can also be seen from Table I that there exists a matrix of row-rank $n-k$ with $\hat{m} = 1394$ rows that is free of trapping sets of size $(6, s)$, $s < 5$. However, for a parity-check matrix free of elementary trapping sets of the same parameters, Table II shows that only $\hat{m} = 1351$ rows are needed.

These bounds cover the case of fixed a values only. Note that finding analogs of these results that cover a *range* of general trapping set sizes a instead may be desirable for certain practical applications. Due to the monotonic increase of the trapping redundancy with the value of the parameter a, one can see that if a set of rows, randomly drawn from the codewords of the dual code, does not contain trapping sets of size a with high probability, then this set is also very unlikely to support trapping sets of size smaller than a. For example, to obtain a matrix free of $(14, s)$ trapping sets, $s < 5$, according to the high-probability version of LLL with $\epsilon = 10^{-20}$ one needs $m = 216$ rows, so that $\hat{m} = 1535$. With *high probability*, this matrix does also avoid $(12, s)$ trapping sets, $s < 5$, due to the LLL-based study.

Note that for elementary trapping sets, Table I indicates that the larger the value of the parameter a, the smaller the number of redundant rows that is needed to eliminate such trapping sets. This result may seem counterintuitive, but it follows from the fact that trapping sets are deemed elementary only as their restriction does not contain rows of weight larger than two - an event that becomes less likely with the increase of the parameter a.

We complete this section by illustrating how a simple *structured method* can be used to add only one redundant parity-check equation so as to increase the value of the parameter b in any given $(a = 12, b = 4)$ or $(a = 14, b = 4)$ trapping set.

Example 4.1: The knowledge about the structure of specific trapping sets - such as the elementary $(12, 4)$ and $(14, 4)$ trapping sets - in the $[2640, 1320]$ Margulis code can be used to eliminate single instances of such sets.

TS size		$\boldsymbol{H}$	LLL (std)		LLL (hp) $\epsilon = 0.01$		LLL (hp) $\epsilon = 10^{-20}$	
a	b	$(n-k) \times n$	m	$\hat{m}$	m	$\hat{m}$	m	$\hat{m}$
6	5	1320×2640	75	1394	87	1406	150	1469
8	5	1320×2640	94	1413	105	1424	167	1486
12	5	1320×2640	129	1448	138	1457	200	1519
14	5	1320×2640	145	1464	154	1473	216	1535

TABLE I

UPPER BOUNDS ON THE (a, b) TRAPPING REDUNDANCY $T_{a,b}(\mathcal{C})$ OF THE MARGULIS CODE.

TS size		$\boldsymbol{H}$	LLL (std)		LLL (hp) $\epsilon = 0.01$		LLL (hp) $\epsilon = 10^{-20}$	
a	b	$(n-k) \times n$	m	$\hat{m}$	m	$\hat{m}$	m	$\hat{m}$
6	5	1320×2640	32	1351	39	1358	70	1389
8	5	1320×2640	26	1345	29	1348	49	1368
12	5	1320×2640	19	1338	20	1339	31	1350
14	5	1320×2640	17	1336	18	1337	26	1345

TABLE II

UPPER BOUNDS ON THE (a, b) ELEMENTARY TRAPPING REDUNDANCY $T^{(e)}_{a,b}(\mathcal{C})$ OF THE MARGULIS CODE.

First, we observe that the support of any $(14, 4)$ trapping set contains the support of a $(12, 4)$ trapping set - i.e., the sets are nested. Throughout the remainder of this section, we call the variables and checks introduced by extending a $(12, 4)$ to a $(14, 4)$ trapping set *expansion variables* and *expansion checks*, respectively. The notation B, E, and O is used to refer to the basic $(12, 4)$ trapping set, its expansion variables and checks, and the graph outside (i.e. complementary to) the $(14, 4)$ trapping set, respectively. Since the Margulis code is regular, with variable node degree $d_v = 3$ and check node degree $d_c = 6$, an elementary (a, b) trapping set with a variables and b check nodes of degree one[2] has a fixed number of checks. Therefore, in order to extend an elementary $(12, 4)$ trapping set to an elementary $(14, 4)$ trapping set, two variables and three checks have to be added. The notions of basic, expansion and outside variables and check nodes are illustrated in Figure 1

Since the Margulis code has no four-cycles, at most one expansion check can be connected to both the expansion variables. Consequently, either three or four edges emanating from the expansion variables are connected to expansion check nodes, while the remaining edges are connected to check nodes whose degree in the basic $(12, 4)$ trapping set is one. Thus there exist only two possible configurations for such trapping sets, as shown in Figure 1.

Now, based only on the knowledge of check nodes of degree one within the basic $(12, 4)$ trapping set, it is straightforward to determine the whole expansion set: in the first case, the two degree-one checks of the basic $(12, 4)$ trapping set are connected through two variable nodes to one additional check node. These two variable nodes are the expansion variables. In the second configuration, the expansion variables do not share a check node.

For the first configuration, denote the two expansion variables

[2]These check nodes, incidentally, correspond to unsatisfied check nodes, that can be readily identified upon termination of the iterative decoding process.

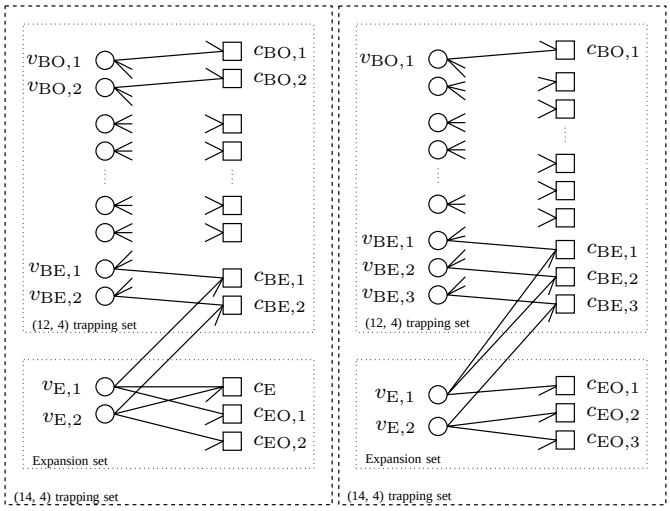

Fig. 1. Trapping set structure of a $(12, 4)$ trapping set and its expansion. (a) first configuration; (b) second configuration.

by $v_{\mathrm{E},1}$ and $v_{\mathrm{E},2}$. Furthermore, denote the check node connected to both these variables by c_{E}. The degree-one check nodes in the basic trapping set neighboring the expansion variables are denoted by $c_{\mathrm{BE},1}$ and $c_{\mathrm{BE},2}$, while the variable nodes in the basic trapping set connected to check nodes $c_{\mathrm{BE},1}$ and $c_{\mathrm{BE},2}$ are denoted by $v_{\mathrm{BE},1}$ and $v_{\mathrm{BE},2}$, respectively. The check nodes of degree one in the expansion of the trapping set are named $c_{\mathrm{EO},1}$ and $c_{\mathrm{EO},2}$. The two remaining check nodes of degree one in the basic trapping set are termed $c_{\mathrm{BO},1}$ and $c_{\mathrm{BO},2}$, and the variable nodes within the basic $(12, 4)$ trapping set connected to them are $v_{\mathrm{BO},1}$ and $v_{\mathrm{BO},2}$, respectively (see Figure 1(a)). The configuration involving all the aforementioned checks and variables is illustrated in Table III.

	Expansion Variables		Basic Trapping Set Variables			
	$v_{\mathrm{E},1}$	$v_{\mathrm{E},2}$	$v_{\mathrm{BE},1}$	$v_{\mathrm{BE},2}$	$v_{\mathrm{BO},1}$	$v_{\mathrm{BO},2}$
c_{E}	1	1	0	0	0	0
$c_{\mathrm{EO},1}$	1	0	0	0	0	0
$c_{\mathrm{EO},2}$	0	1	0	0	0	0
$c_{\mathrm{BE},1}$	1	0	1	0	0	0
$c_{\mathrm{BE},2}$	0	1	0	1	0	0
$c_{\mathrm{BO},1}$	0	0	0	0	1	0
$c_{\mathrm{BO},2}$	0	0	0	0	0	1

TABLE III

RESTRICTION OF THE BASIC TRAPPING SET AND THE EXPANSION VARIABLES

The parity-check equations containing the restriction described in Table III can be linearly combined to generate a redundant parity-check equation that has a restriction of odd weight in both the $(12, 4)$ and the $(14, 4)$ trapping set. There are three different methods for linearly combining the parity-check equations with restrictions as shown in Table III.

Method S_1 refers to adding the rows indexed by (c_{E}, $c_{\mathrm{EO},1}$, $c_{\mathrm{BE},2}$), (c_{E}, $c_{\mathrm{EO},2}$, $c_{\mathrm{BE},1}$), ($c_{\mathrm{EO},1}$, $c_{\mathrm{BE},1}$), and ($c_{\mathrm{EO},2}$, $c_{\mathrm{BE},2}$). Method S_2 refers to adding the rows indexed by (c_{E}, $c_{\mathrm{EO},1}$) and (c_{E}, $c_{\mathrm{EO},2}$). Observe that all these combinations have a restriction of weight one within the expansion variables. Method S_3 differs from the previous methods in so far that it may generate redundant parity-check equations with odd-weight restrictions that are not necessarily of weight one. Candidate equations are obtained by adding the rows indexed by (c_{E}, $c_{\mathrm{BO},1}$), (c_{E}, $c_{\mathrm{BO},2}$), (c_{E}, $c_{\mathrm{BE},1}$, $c_{\mathrm{BE},2}$, $c_{\mathrm{BO},1}$), (c_{E}, $c_{\mathrm{BE},1}$, $c_{\mathrm{BE},2}$, $c_{\mathrm{BO},2}$), ($c_{\mathrm{BO},1}$, $c_{\mathrm{BO},2}$, $c_{\mathrm{EO},1}$, $c_{\mathrm{BE},2}$), and ($c_{\mathrm{BO},1}$, $c_{\mathrm{BO},2}$, $c_{\mathrm{EO},2}$, $c_{\mathrm{BE},1}$). The Hamming weight of the constructed rows is an even integer between 10 and 24. The lower bound 10 is obtained if the supports of two added parity-checks share exactly one element. The intersection of the supports cannot have more than one element, since the code has no four-cycles. The upper bound 24 is met when four parity-check equations are added and none of the variables listed in Table III occur in the support of more than one parity-check.

If a trapping set of the form shown in Figure 1(b) is present in the code graph, then the two expansion variables cannot have a common check node. Consequently, the expansion variable $v_{\mathrm{E},1}$ is connected to two of the degree-one check nodes of the basic trapping set.[3] If there is only one variable node connected to two degree-one checks, denoted by $c_{\mathrm{BE},1}$ and $c_{\mathrm{BE},2}$, the variable of interest is the expansion variable $v_{\mathrm{E},1}$, which is also connected to expansion check node $c_{\mathrm{EO},1}$. Observe that the expansion variable $v_{\mathrm{E},2}$ is strongly influenced by its two neighboring check nodes connected to the outside graph and it cannot be uniquely determined. Due to this limited knowledge of the expansion set, there are only two possible ways to generate a redundant parity-check with an odd restriction weight on the basic trapping set, involving the sums of the rows ($c_{\mathrm{EO},1}$, $c_{\mathrm{BE},1}$) as well as ($c_{\mathrm{EO},1}$, $c_{\mathrm{BE},2}$). A similar analysis can be conducted for $(14, 4)$ trapping sets. Details regarding this procedure are omitted.

As illustrated by the example, knowledge about the structure of trapping sets allows one to exactly determine the choice of the redundant row to be added to the parity check matrix. Unfortunately, since there are at least 1320 trapping sets of each such form in the code graph, adding this many rows to the code matrix is undesirable. Nevertheless, as already pointed out, only one row can be added upon detecting the presence of a decoding failure caused by a given trapping set.

B. Projective Geometry Codes

Projective geometry codes are linear block codes with many well known combinatorial parameters and properties. As will be shown next, it is also straightforward to characterize a large sub-family of trapping sets in these codes.

We start our derivations by introducing the relevant terminology.

Definition 4.1: [26] A finite projective geometry $\mathrm{PG}(M, q)$ of dimension M, over a finite field $\mathrm{GF}(q)$, for some prime power q, is a set of points and subsets thereof, called lines. The following axioms hold for the points and lines of a finite geometry:

- Two distinct points determine a unique line.
- Every line consists of more than two points.
- For every pair of distinct lines L_1 and L_2, intersecting at some point r, there exist two pairs of points $(p_1, q_1) \in L_1$ and $(p_2, q_2) \in L_2$ that differ from r, such that the lines determined by (p_1, p_2) and (q_1, q_2) intersect as well.

[3]One must keep in mind that the choice for such a variable may not be unique.

- For each point and for each line, there exist at least two lines and two points that are not incident to them, respectively.

The points of a projective geometry $\mathrm{PG}(M,q)$ can be represented by non-zero $(M+1)$-tuples $(a_0,a_1,a_2,...,a_M)$ such that $a_i \in \mathrm{GF}(q)$. Points of the form $(a_0,a_1,a_2,...,a_M)$ and $(\delta a_0,\delta a_1,\delta a_2,...,\delta a_M)$, $\delta \in \mathrm{GF}(q)\backslash\{0\}$, are considered equivalent. A line through two distinct points $(a_0,a_1,a_2,...,a_M)$ and $(b_0,b_1,b_2,...,b_M)$ consists of all points that can be expressed as $(\alpha\, a_0+\beta\, b_0,...,\alpha\, a_M+\beta\, b_M)$, where $\alpha,\beta \in \mathrm{GF}(q)$ and are not both simultaneously zero. Consequently, a projective geometry $\mathrm{PG}(M,q)$ has $(q^{M+1}-1)/(q-1)$ points, and each line in the geometry contains $q+1$ points. The number of lines in a projective geometry is given by

$$(q^M+...+q+1)(q^{M-1}+...+q+1)/(q+1). \tag{20}$$

It is straightforward to see that the number of lines and points coincide for $M=2$, since $(q^2+q+1)=(q^3-1)/(q-1)$ holds.

A type-I projective geometry code is defined in terms of a parity-check matrix representing the *line-point* incidence matrix of a projective geometry $\mathrm{PG}(M,q)$ [27]. Throughout the remainder of the paper, we consider projective plane codes, $M=2$, and codes based on projective geometries with $M=3$ only.

Definition 4.2: An s-arc in $\mathrm{PG}(2,q)$ is a collection of s points such that no three of them are collinear. The lines incident to an s-arc $\mathcal{K}$ are either unisecants (they intersect the arc in exactly one point) or bisecants (they intersect the arc in exactly two points). Similarly, an s-cap in $\mathrm{PG}(3,q)$ is a set of s points, no three of which are collinear.

The following results pertaining to unisecants and bisecants are taken from [28, Ch. 8] and [29, Ch. 16].

Lemma 4.3: Let n_1 and n_2 denote the number of unisecants and bisecants of an s-arc $\mathcal{K}$ in $\mathrm{PG}(2,q)$, respectively. Then

$$n_1 = s(q+2-s), \quad \text{and} \quad n_2 = \frac{1}{2}s(s-1). \tag{21}$$

Similarly, for an s-cap $\mathcal{K}$ in $\mathrm{PG}(3,q)$ it holds that

$$n_1 = s(q^2+q+2-s), \tag{22}$$

where n_1 denotes the number of unisecants of $\mathcal{K}$.

Lemma 4.4: The largest arc in $\mathrm{PG}(2,q)$ contains at most $q+2$ points, for q even, and $q+1$ points, for q odd. Arcs with $s=q+1$ and $s=q+2$ are called ovals and hyperovals, respectively. The size of any s-cap in $\mathrm{PG}(3,q)$ satisfies $s \leq q^2+1$. For $q>2$, a (q^2+1)-cap is called an ovaloid.

The results of Lemmas 4.3 and 4.4 can be used to establish the following simple results regarding trapping sets in the Tanner graph of type-I projective geometry codes. Note that all stated results restrict the parameter sets for which trapping sets may potentially exist, although they do not imply the existence of such sets.

Corollary 4.5: All elementary trapping sets of a $\mathrm{PG}(2,q)$, type-I, projective geometry code have parameters $(s, s(q+2-s))$. Consequently, the number of degree-one check nodes of such trapping sets for q odd is necessarily larger than or equal to the number of variables in the trapping set. For even values of q, an exception to the aforementioned rule is a hyperoval, which represents a codeword. The trapping sets with the smallest ratio b/a have parameters $(q+1,q+1)$ (q odd) and $(q+2,0)$ (q even), respectively, and those with the largest ratio $(3,3(q-1))$.

Proof: Note that the parity-check matrix $\boldsymbol{H}$ of a $\mathrm{PG}(2,q)$ code is the line-point incidence matrix of the underlying PG. Arcs correspond to a collection of columns, the restriction of which has rows of weight at most two only, and exactly n_1 of these rows have weight one. This is equivalent to the definition of an (s,n_1) trapping set, where $s \geq 3$ and also $s \leq (q+1)$ (q odd) or $s \leq (q+2)$ (q even), respectively, according to Lemma 4.4. The smallest and largest ratios of b/a are defined by the limits of s. Hyperovals have $s=q+2$ points and $n_1 = (q+2)\cdot(q+2-(q+2))=0$ unisecants, and therefore correspond to $(q+2,0)$ trapping sets, which are codewords. ■

Corollary 4.6: All elementary trapping sets of a $\mathrm{PG}(3,q)$, type-I projective geometry code have parameters $(s, s(q^2+q+2-s))$. Provided that the PG contains an s-arc with $n_1 = s(q^2+q+2-s)$, the trapping sets with the smallest and largest ratio b/a have parameters $(q^2+1,(q^2+1)(q+1))$ and $(3,3(q^2+q-1))$, respectively.

Proof: The proof follows along the lines of the proof of Corollary 4.5, with $n_1 = s(q^2+q+2-s)$ for $\mathrm{PG}(3,q)$. ■

A complete classification of trapping sets in projective geometry codes is probably an impossible task. This is due to the fact that very little is known about the number and existence of arcs and caps of different sizes in projective spaces. One aspect of this problem that is better understood is the existence and enumeration of *complete arcs* (and *caps*) - i.e. arcs and caps not contained in any larger arc or cap. The interested reader is referred to [28], [29] for more information regarding the problem of complete arc enumeration.

We compare next the upper bounds derived in Section III with the results of the study presented in this section. We consider elementary trapping sets only.

In order to apply the results of the lemmas in this section, one has to consider a parity-check matrix that represents a complete line-point incidence structure [27]. For this reason, the standard parity-check matrices of PG codes contain exactly n rows, and are therefore redundant.

Table IV list the number of rows required to avoid elementary trapping sets of a given size, computed according to LLL and its high-probability variation. The values for $\hat{m}$ are derived using the minimum distance and the rank results taken from [27].

Observe that Table IV indicates that there exists a parity-check matrix for the $\mathrm{PG}(2,16)$ code with at most 175 rows and no $(3,s)$, $s<45$, elementary trapping sets. This is significantly less than $n=273$ as for the PG code, but might also include rows of large weight. On the other hand, to obtain a matrix without such trapping sets with probability larger than $1-10^{-20}$, at most 309 rows are required, a number clearly larger than n.

Although there exist parity-check matrices for the PG codes that have smaller row-redundancy and do not contain elementary trapping sets of small sizes, the matrix defined by the PG construction presents a way to generate a parity-check matrix with limited redundancy, small row-weight, and no four-cycles, which also performs well in the waterfall region. As can be seen from the comparison table, PG codes represent an attractive

Code	TS size		compl. $\boldsymbol{H}$	min. $\boldsymbol{H}$	LLL (std)		LLL (hp) $\epsilon = 0.01$		LLL (hp) $\epsilon = 10^{-20}$	
	a	b	$\overline{m} = n$	$\underline{m} = n - k$	m	$\hat{m}$	m	$\hat{m}$	m	$\hat{m}$
$\mathrm{PG}(2, 16)$	3	45	273	82	94	175	125	206	228	309
$\mathrm{PG}(2, 16)$	8	80	273	82	80	161	80	161	80	161
$\mathrm{PG}(2, 32)$	3	93	1057	244	115	358	178	421	338	581
$\mathrm{PG}(2, 32)$	16	288	1057	244	288	531	288	531	288	531

TABLE IV

UPPER BOUNDS ON THE (a, b) ELEMENTARY TRAPPING REDUNDANCY OF PG CODES.

systematic construction of LDPC codes without small trapping sets, whose redundancy lies between the bounds based on the standard case and the high-probability version of LLL.

V. CONCLUSION

We introduced the notion of the (a, b) trapping redundancy of a code, representing the smallest number of rows in any parity-check matrix of the code that avoids (a, s) trapping sets with $1 \leq s < b$. Upper bounds on these combinatorial numbers were derived using Lovász Local Lemma and variations thereof. Also presented were numerical results for the trapping redundancy of the Margulis $[2640, 1320]$ and type-I PG codes.

Acknowledgment: The authors are grateful to the anonymous reviewers for their constructive comments that significantly improved the exposition of the work. In addition, they would like to thank Dr. Richardson for carefully handling the manuscript.

REFERENCES

[1] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.

[2] D. MacKay and M. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes," *Electronic Notes in Theoretical Computer Science*, vol. 74, 2003. [Online]. Available: http://www.cs.toronto.edu/∼mackay/margulis.pdf

[3] T. Richardson, "Error-floors of LDPC codes," in *Proceedings of the 41st Annual Allerton Conference on Communication, Control and Computing*, Monticallo, Illinois, USA, September 2003, pp. 1426–1435.

[4] J. Han and P. Siegel, "Improved upper bounds on stopping redundancy," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 90–104, January 2007.

[5] R. Koetter, W.-C. W. Li, P. Vontobel, and J. Walker, "Characterizations of pseudo-codewords of LDPC codes," *accepted for Advances in Mathematics*, August 2006.

[6] M. Schwartz and A. Vardy, "On the stopping distance and stopping redundancy of codes," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 922–932, March 2006.

[7] J. Weber and K. Abdel-Ghaffar, "Stopping and dead-end set enumerators for binary Hamming codes," in *Proceedings of the Twenty-sixth Symposium on Information Theory in the Benelux*, Brussels, Belgium, May 2005, pp. 165–172.

[8] A. Hedayat, N. Sloane, and J. Stufken, *Orthogonal Arrays: Theory and Applications*. New York, USA: Springer Verlag, 1999.

[9] O. Milenkovic, E. Soljanin, and P. Whiting, "Stopping and trapping sets in generalized covering arrays," in *Proceedings of the 40th annual Conference on Information Sciences and Systems (CISS)*, Princeton, New Jersey, USA, March 2006, pp. 259–264.

[10] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[11] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, February 2001.

[12] T. Richardson and R. Urbanke, "The capacity of LDPC codes under message passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, February 2001.

[13] J. Rosenthal and P. Vontobel, "Constructions of regular and irregular LDPC codes using Ramanujan graphs and ideas from Margulis," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Washington, District of Columbia, USA, June 24–29 2001, p. 4.

[14] A. McGregor and O. Milenkovic, "On the hardness of approximating stopping and trapping sets in LDPC codes," in *Proceedings of the IEEE Information Theory Workshop (ITW)*, Lake Tahoe, California, USA, September 2007, pp. 248–253.

[15] O. Milenkovic, E. Soljanin, and P. Whiting, "Asymptotic spectra of trapping sets in regular and irregular LDPC code ensembles," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 39–55, January 2007.

[16] S. Laendner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Proceedings of the International Conference on Wireless Networks, Communications, and Mobile Computing (WirelessComm)*, Maui, Hawaii, June 2005, pp. 630–635.

[17] H. Hollmann and L. Tolhuizen, "On parity-check collections for iterative erasure decoding that correct all correctable erasure patterns of a given size," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 823–828, February 2007.

[18] T. Hehn, O. Milenkovic, S. Laendner, and J. Huber, "Permutation decoding and the stopping redundancy hierarchy of cyclic and extended cyclic codes," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5308–5331, December 2008.

[19] N. Alon and J. Spencer, *The Probabilistic Method*, ser. Interscience Series in Discrete Mathematics and Optimization. John Wiley, 2000.

[20] M. Hofri, *Analysis of Algorithms*. Oxford, U.K.: Oxford University Press, 1995.

[21] P. Brockwell, "An asymptotic expansion for the tail of the binomial distribution and its application in queuing theory," *Journal of Applied Probability*, vol. 1, no. 1, pp. 163–169, June 1964.

[22] S. Laendner and O. Milenkovic, "Codes based on latin squares: Cycle structure, stopping set, and trapping set analysis," *IEEE Transaction on Communications*, vol. 55, no. 2, pp. 303–312, February 2007.

[23] X.-Y. Hu. Source code for approximating the MinDist problem of LDPC codes. Error-Correcting Codes Website by D. MacKay. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/MINDIST_ECC.html

[24] X.-Y. Hu, M. Fossorier, and E. Eleftheriou, "On the computation of the minimum distance of low-density parity-check codes," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Paris, France, June 2004, pp. 767–771.

[25] G. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, vol. 2, no. 1, pp. 71–78, March 1982.

[26] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland Publishing Company, 1977.

[27] Y. Kou, S. Lin, and M. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711–2736, November 2001.

[28] J. Hirschfeld, *Projective geometeries over finite fields*. Oxford Mathematical Monographs, 1979.

[29] ——, *Finite projective spaces of three dimensions*. Oxford, U.K.: Oxford Mathematical Monographs, 1985.