Analysis of Error Floors of LDPC Codes under LP Decoding over the BSC

Shashi Kiran Chilappagari, Bane Vasic
Dept. of ECE
University of Arizona
Tucson, AZ 85721, USA
Email: {shashic,vasic}@ece.arizona.edu

Mikhail Stepanov
Dept. of Mathematics
University of Arizona
Tucson, AZ 85721, USA
Email: stepanov@math.arizona.edu

Michael Chertkov
Theoretical Division
Los Alamos National Lab
Los Alamos, NM 87545, USA
Email: chertkov@lanl.gov

***Abstract*—We consider Linear Programming (LP) decoding of a fixed Low-Density Parity-Check (LDPC) code over the Binary Symmetric Channel (BSC). The LP decoder fails when it outputs a pseudo-codeword which is not a codeword. We propose an efficient algorithm termed the Instanton Search Algorithm (ISA) which, given a random input, generates a set of flips called the BSC-instanton and prove that: (a) the LP decoder fails for any set of flips with support vector including an instanton; (b) for any input, the algorithm outputs an instanton in the number of steps upper-bounded by twice the number of flips in the input. Repeated sufficient number of times, the ISA outcomes the number of unique instantons of different sizes. We use the instanton statistics to predict the performance of the LP decoding over the BSC in the error floor region. We also propose an efficient semi-analytical method to predict the performance of LP decoding over a large range of transition probabilities of the BSC.**

I. Introduction

The significance of Low-Density Parity-Check (LDPC) codes [1] is in their capacity-approaching performance when decoded using low complexity iterative algorithms, such as Belief Propagation (BP) [1], [2]. The linear programming (LP) decoding introduced by Feldman *et al.* [3], is another sub-optimal algorithm for decoding LDPC codes, which has higher complexity but is more amenable to analysis. The typical performance measures of a decoder (either LP or BP) for a fixed code are the Bit-Error-Rate (BER) or/and the Frame-Error-Rate (FER) as functions of the Signal-to-Noise Ratio (SNR). A typical BER/FER vs SNR curve consists of two distinct regions. At small SNR, the error probability decreases rapidly with the SNR, and the curve forms the so-called *waterfall* region. The decrease slows down at moderate values turning into the *error floor* asymptotic at very large SNR [4].

After the formulation of the problem by Richardson [4], a significant effort has been devoted to the analysis of the error floor phenomenon. Given that the decoding sub-optimality is expressed in the domain where the error probability is small, the troublesome noise configurations leading to decoding failures and controlling the error floor asymptotic are extremely rare, and analytical rather than simulation methods for their characterization are necessary.

In this paper, we consider pseudo-codewords [3] and instantons of the LP decoder [3] for the BSC. We define the *BSC-instanton* as a noise configuration which is decoded by the LP decoder into a pseudo-codeword distinct from the all-zero-codeword while any reduction of the (number of flips in) BSC-instanton leads to the all-zero-codeword. Finding the instantons is a difficult task which so far admitted only heuristic solutions [5], [6]. In this regard, the most successful (in efficiency) approach, coined the Pseudo-Codeword-Search (PCS) algorithm, was suggested for the LP decoding performing over the continuous channel in [7] (with Additive White Gaussian Noise (AWGN) channel used as an enabling example). Our Instanton Search Algorithm (ISA) is an extension of the PCS to the BSC, and constitutes a significantly stronger algorithm than the one of [7] due to its property that it outputs an instanton in the number of steps upper-bounded by twice the number of flips in the original configuration the algorithm is initiated with.

The main contributions of this paper are: (1) characterization of all the failures of the LP decoder over the BSC in terms of the instantons, (2) a provably efficient Instanton Search Algorithm (ISA), and (3) a semi-analytical method to predict the performance of LP decoding over a large range of transition probabilities of the BSC. An extended version of this paper with the proofs of all the lemmata and theorems (however lacking some of experimental results and analysis of the code performance in the transient regime discussed in this manuscript) has been submitted to IEEE Transactions on Information Theory (see [8]).

The rest of the paper is organized as follows. In Section II, we give a brief introduction to the LDPC codes, LP decoding and pseudo-codewords. In Section III, we introduce the BSC-specific notions for the pseudo-codeword weight, medians and instantons (defined as special set of flips), their costs, and we also state some set of useful lemmata emphasizing the significance of the instanton analysis. In Section IV, we describe the ISA, state our main result concerning bounds on the number of iterations required to output an instanton, and describe how to utilize ISA for reconstructing FER vs SNR curve in the intermediate range interfacing from the waterfall to the the error floor. Section V summarizes our numerical experiments.

II. Preliminaries: LDPC Codes, LP Decoder and Pseudo-Codewords

In this section, we discuss the LP decoder and the notion of pseudo-codewords. We adopt the formulation of the LP decoder and the terminology from [3], and thus the interested reader is advised to refer to [3] for more details.

Let $\mathcal{C}$ be a binary LDPC code defined by a Tanner graph G with two sets of nodes: the set of variable nodes $V = \{1, 2, \ldots, n\}$ and the set of check nodes $C = \{1, 2, \ldots, m\}$. The adjacency matrix of G is H, a parity-check matrix of $\mathcal{C}$, with m rows corresponding to the check nodes and n columns corresponding to the variable nodes. A binary vector $\mathbf{c} = (c_1, \ldots, c_n)$ is a codeword *iff* $\mathbf{c}H^T = \mathbf{0}$. The support of a vector $\mathbf{r} = (r_1, r_2, \ldots, r_n)$, denoted by supp$(\mathbf{r})$, is defined as the set of all positions i such that $r_i \neq 0$.

We assume that a codeword $\mathbf{y}$ is transmitted over a discrete symmetric memoryless channel and is received as $\hat{\mathbf{y}}$. The channel is characterized by $\Pr[\hat{y}_i|y_i]$ which denotes the probability that y_i is received as $\hat{y}_i$. The negative log-likelihood ratio (LLR) corresponding to the variable node i is given by

$$\gamma_i = \log\left(\frac{\Pr(\hat{y}_i|y_i = 0)}{\Pr(\hat{y}_i|y_i = 1)}\right).$$

The ML decoding of the code $\mathcal{C}$ allows a convenient LP formulation in terms of the *codeword polytope* poly$(\mathcal{C})$ whose vertices correspond to the codewords in $\mathcal{C}$. The ML-LP decoder finds $\mathbf{f} = (f_1, \ldots, f_n)$ minimizing the cost function $\sum_{i=1}^{n} \gamma_i f_i$ subject to the $\mathbf{f} \in$ poly$(\mathcal{C})$ constraint. The formulation is compact but impractical, as the number of constraints is exponential in the code length.

Hence a *relaxed* polytope is defined as the intersection of all the polytopes associated with the local codes introduced for all the checks of the original code. Associating $(f_1, \ldots, f_n)$ with bits of the code we require

$$0 \leq f_i \leq 1, \quad \forall i \in V \tag{1}$$

For every check node j, let $N(j)$ denote the set of variable nodes which are neighbors of j. Let $E_j = \{T \subseteq N(j) : |T| \text{ is even}\}$. The polytope Q_j associated with the check node j is defined as the set of points $(\mathbf{f}, \mathbf{w})$ for which the following constraints hold

$$0 \leq w_{j,T} \leq 1, \qquad \forall T \in E_j \tag{2}$$

$$\textstyle\sum_{T \in E_j} w_{j,T} = 1 \tag{3}$$

$$f_i = \textstyle\sum_{T \in E_j, T \ni i} w_{j,T}, \quad \forall i \in N(j) \tag{4}$$

Now, let $Q = \cap_j Q_j$ be the set of points $(\mathbf{f}, \mathbf{w})$ such that (1)-(4) hold for all $j \in C$. (Note that Q, which is also referred to as the fundamental polytope [9], [10], is a function of the Tanner graph G and consequently the parity-check matrix H representing the code $\mathcal{C}$.) The Linear Code Linear Program (LCLP) can be stated as

$$\min_{(\mathbf{f},\mathbf{w})} \sum_{i \in V} \gamma_i f_i, \text{ s.t. } (\mathbf{f}, \mathbf{w}) \in Q.$$

For the sake of brevity, the decoder based on the LCLP is referred to in the following as the LP decoder. A solution $(\mathbf{f}, \mathbf{w})$ to the LCLP such that all f_is and $w_{j,T}$s are integers is known as an integer solution. The integer solution represents a codeword [3]. It was also shown in [3] that the LP decoder has the ML certificate, i.e., if the output of the decoder is a codeword, then the ML decoder would decode into the same codeword. The LCLP can fail, generating an output which is not a codeword.

The performance of the LP decoder can be analyzed in terms of the pseudo-codewords, originally defined as follows:

Definition 1: [3] *Integer pseudo-codeword* is a vector $\mathbf{p} = (p_1, \ldots, p_n)$ of non-negative integers such that, for every parity check $j \in C$, the neighborhood $\{p_i : i \in N(j)\}$ is a sum of local codewords.

Alternatively, one may choose to define a *re-scaled pseudo-codeword*, $\mathbf{p} = (p_1, \ldots, p_n)$ where $0 \leq p_i \leq 1, \forall i \in V$, simply equal to the output of the LCLP. In the following, we adopt the re-scaled definition.

A given code $\mathcal{C}$ can have different Tanner graph representations and consequently potentially different fundamental polytopes. Hence, we refer to the pseudo-codewords as corresponding to a particular Tanner graph G of $\mathcal{C}$.

III. Cost and Weight of Pseudo-codewords, Medians and Instantons

Since the focus of the paper is on the pseudo-codewords for the BSC, in this section we introduce some terms, e.g. instantons and medians, specific to the BSC. We will also state here some preliminary lemmata which will enable subsequent discussion of the ISA in the next Section.

The polytope Q is symmetric and looks exactly the same from all codewords (see e.g. [3]). Hence we assume that the all-zero-codeword is transmitted. The process of changing a bit from 0 to 1 and vice-versa is known as flipping. The BSC flips every transmitted bit with a certain probability (henceforth denoted by α). We therefore call a noise vector with support of size k as having k flips.

In the case of the BSC, the likelihoods are scaled as

$$\gamma_i = \begin{cases} 1, & \text{if } y_i = 0; \\ -1, & \text{if } y_i = 1. \end{cases}$$

Two important characteristics of a pseudo-codeword are its cost and weight. While the cost associated with decoding to a pseudo-codeword has already been defined in general, we formalize it for the case of the BSC as follows:

Definition 2: The cost associated with LP decoding of a binary vector $\mathbf{r}$ to a pseudo-codeword $\mathbf{p}$ is given by

$$C(\mathbf{r}, \mathbf{p}) = \sum_{i \notin \text{supp}(\mathbf{r})} p_i - \sum_{i \in \text{supp}(\mathbf{r})} p_i. \tag{5}$$

If $\mathbf{r}$ is the input, then the LP decoder converges to the pseudo-codeword $\mathbf{p}$ which has the least value of $C(\mathbf{r}, \mathbf{p})$. The cost of decoding to the all-zero-codeword is zero. Hence, a binary vector $\mathbf{r}$ does not converge to the all-zero-codeword if there exists a pseudo-codeword $\mathbf{p}$ with $C(\mathbf{r}, \mathbf{p}) \leq 0$.

Definition 3: [11], [12, Definition 2.10] Let $\mathbf{p} = (p_1, \ldots, p_n)$ be a pseudo-codeword distinct from the all-zero-codeword. Let e be the smallest number such that the sum of the e largest p_is is at least $\left(\sum_{i \in V} p_i\right)/2$. Then, the BSC *pseudo-codeword weight* of $\mathbf{p}$ is

$$w_{\text{BSC}}(\mathbf{p}) = \begin{cases} 2e, & \text{if } \sum_e p_i = \left(\sum_{i \in V} p_i\right)/2; \\ 2e - 1, & \text{if } \sum_e p_i > \left(\sum_{i \in V} p_i\right)/2. \end{cases}$$

The minimum pseudo-codeword weight of G denoted by w_{min}^{BSC} is the minimum over all the non-zero pseudo-codewords of G. The parameter $e = \lceil (w_{\text{BSC}}(\mathbf{p}) + 1)/2 \rceil$ can be interpreted as the least number of bits to be flipped in the all-zero-codeword such that the resulting vector decodes to the pseudo-codeword $\mathbf{p}$. (See e.g. [12] for a number of illustrative examples.)

The interpretation of BSC pseudo-codeword weight motivates the following definition of the *median noise vector* corresponding to a pseudo-codeword:

Definition 4: The median noise vector (or simply the median) $M(\mathbf{p})$ of a pseudo-codeword $\mathbf{p}$ distinct from the all-zero-codeword is a binary vector with support $S = \{i_1, i_2, \ldots, i_e\}$, such that $p_{i_1}, \ldots, p_{i_e}$ are the $e(= \lceil (w_{\text{BSC}}(\mathbf{p}) + 1)/2 \rceil)$ largest components of $\mathbf{p}$.
One observers that, $C(M(\mathbf{p}), \mathbf{p}) \leq 0$. From the definition of $w_{\text{BSC}}(\mathbf{p})$, it follows that at least one median exists for every $\mathbf{p}$. Also, all medians of $\mathbf{p}$ have $\lceil (w_{\text{BSC}}(\mathbf{p}) + 1)/2 \rceil$ flips. The following lemmata characterize some important properties of the median.

Lemma 1: The LP decoder decodes a binary vector with k flips into a pseudo-codeword $\mathbf{p}$ distinct from the all-zero-codeword *iff* $w_{\text{BSC}}(\mathbf{p}) \leq 2k$.

Lemma 2: Let $\mathbf{p}$ be a pseudo-codeword with median $M(\mathbf{p})$ whose support has cardinality k. Then $w_{\text{BSC}}(\mathbf{p}) \in \{2k-1, 2k\}$.

Lemma 3: Let $M(\mathbf{p})$ be a median of $\mathbf{p}$ with support S. Then the result of LP decoding of any binary vector with support $S' \subset S$ and $|S'| < |S|$ is distinct from $\mathbf{p}$.

Lemma 4: If $M(\mathbf{p})$ converges to a pseudo-codeword $\mathbf{p}_M \neq \mathbf{p}$, then $w_{\text{BSC}}(\mathbf{p}_M) \leq w_{\text{BSC}}(\mathbf{p})$. Also, $C(M(\mathbf{p}), \mathbf{p}_M) \leq C(M(\mathbf{p}), \mathbf{p})$.

Definition 5: The BSC *instanton* $\mathbf{i}$ is a binary vector with the following properties: (1) There exists a pseudo-codeword $\mathbf{p}$ such that $C(\mathbf{i}, \mathbf{p}) \leq C(\mathbf{i}, \mathbf{0}) = 0$; (2) For any binary vector $\mathbf{r}$ such that $\text{supp}(\mathbf{r}) \subset \text{supp}(\mathbf{i})$, there exists no pseudo-codeword with $C(\mathbf{r}, \mathbf{p}) \leq 0$. The size (or weight) of an instanton is the cardinality of its support.
In other words, the LP decoder decodes $\mathbf{i}$ to a pseudo-codeword other than the all-zero-codeword or one finds a pseudo-codeword $\mathbf{p}$ such that $C(\mathbf{i}, \mathbf{p}) = 0$ (interpreted as the LP decoding failure), whereas any binary vector with flips from a subset of the flips in $\mathbf{i}$ is decoded to the all-zero-codeword.

The following lemma follows from the definition of the cost of decoding (the pseudo-codeword cost):

Lemma 5: Let $\mathbf{i}$ be an instanton. Then for any binary vector $\mathbf{r}$ such that $\text{supp}(\mathbf{i}) \subset \text{supp}(\mathbf{r})$, there exists a pseudo-codeword $\mathbf{p}$ satisfying $C(\mathbf{r}, \mathbf{p}) \leq 0$.

The above lemma implies that the LP decoder fails to decode every vector $\mathbf{r}$ whose support is a superset of an instanton to the all-zero-codeword. We now have the following corollary:

Corollary 1: Let $\mathbf{r}$ be a binary vector with support S. Let $\mathbf{p}$ be a pseudo-codeword such that $C(\mathbf{r}, \mathbf{p}) \leq 0$. If all binary vectors with support $S' \subset S$, such that $|S'| = |S| - 1$, converge to $\mathbf{0}$, then $\mathbf{r}$ is an instanton.

The above lemmata lead us to the following lemma which characterizes all the failures of the LP decoder over the BSC:

Lemma 6: A binary vector $\mathbf{r}$ converges to a pseudo-codeword different from the all-zero-codeword *iff* the support of $\mathbf{r}$ contains the support of an instanton as a subset.

From the above discussion, we see that the BSC instantons are analogous to the minimal stopping sets for the case of iterative/LP decoding over the BEC. In fact, Lemma 6 characterizes all the decoding failures of the LP decoder over the BSC in terms of the instantons and can be used to derive analytical estimates of the code performance given the weight distribution of the instantons (this will be illustrated in Section V).

IV. Instanton Search Algorithm and Performance Analysis

In this section, we first describe the Instanton Search Algorithm and then proceed to describe a semi-analytical method to predict the FER performance of a given code.

A. The Instanton Search Algorithm

The ISA starts with a random binary vector with some number of flips and outputs an instanton.

Instanton Search Algorithm

Initialization (l=0) step: Initialize to a binary input vector $\mathbf{r}$ containing sufficient number of flips so that the LP decoder decodes it into a pseudo-codeword different from the all-zero-codeword. Apply the LP decoder to $\mathbf{r}$ and denote the pseudo-codeword output of LP by $\mathbf{p}^1$.

$l \geq 1$ *step*: Take the pseudo-codeword $\mathbf{p}^l$ (output of the $(l-1)$ step) and calculate its median $M(\mathbf{p}^l)$. Apply the LP decoder to $M(\mathbf{p}^l)$ and denote the output by $\mathbf{p}_{M_l}$. By Lemma 4, only two cases arise:

- $w_{\text{BSC}}(\mathbf{p}_{M_l}) < w_{\text{BSC}}(\mathbf{p}^l)$. Then $\mathbf{p}^{l+1} = \mathbf{p}_{M_l}$ becomes the l-th step output/$(l+1)$ step input.
- $w_{\text{BSC}}(\mathbf{p}_{M_l}) = w_{\text{BSC}}(\mathbf{p}^l)$. Let the support of $M(\mathbf{p}^l)$ be $S = \{i_1, \ldots, i_{k_l}\}$. Let $S_{i_t} = S \backslash \{i_t\}$ for some $i_t \in S$. Let $\mathbf{r}_{i_t}$ be a binary vector with support S_{i_t}. Apply the LP decoder to all $\mathbf{r}_{i_t}$ and denote the i_t-output by $\mathbf{p}_{i_t}$. If $\mathbf{p}_{i_t} = \mathbf{0}, \forall i_t$, then $M(\mathbf{p}^l)$ is the desired instanton and the algorithm halts. Else, $\mathbf{p}_{i_t} \neq \mathbf{0}$ becomes the l-th step output/$(l+1)$ step input. (Notice, that Lemma 3 guarantees that any $\mathbf{p}_{i_t} \neq \mathbf{p}^l$, thus preventing the ISA from entering into an infinite loop.)

Theorem 1 below addresses the bounds on the number of steps in which the ISA terminates.

Theorem 1: $w_{\text{BSC}}(\mathbf{p}^l)$ and $|\text{supp}(M(\mathbf{p}^l))|$ are monotonically decreasing. Also, the ISA terminates in at most $2k_0$ steps, where k_0 is the number of flips in the input.

B. Performance Prediction Using Instanton Statistics

In [13], [14], it was shown that the slope of the (log-log) FER curve in the asymptotic limit of $\alpha \to 0$ is equal to the size of the smallest weight instanton. In other words, most of the decoding failures in the error floor region are due to the low-weight instantons. Hence, the instanton statistics can be used to predict the FER performance for small values of α. For large values of α (near the threshold), the FER performance can be estimated with a very good accuracy by Monte-Carlo simulations. FER estimates in this region can be made with a fixed complexity (the details of which will be explained subsequently). The region of intermediate α, interfacing from waterfall to error floor, is the most difficult one for predicting performance. Analytical estimates cannot be made as the instanton statistics for higher weight instantons are not complete. This is due to the fact that the number of instantons grows with the size and the ISA needs to run for a large number of initiations to gather reliable statistics about higher weight instantons. On the other hand Monte-Carlo estimates cannot be made due to prohibitive complexity. Hence, we make use of an approach that is a combination of Monte-Carlo simulations and analytical approach.

Observe that a decoder failure for a pattern with k errors can occur due to the presence of an instanton (or instantons) of size less than or equal to k. Let $\Pr(r|k)$ denote the probability that an instanton of size r is present in an error pattern of size k. If the number of instantons of size r is denoted by T_r then, it can be seen that, whenever this probability is small,

$$\Pr(r|k) \approx \frac{\binom{k}{r} T_r}{\binom{n}{r}}. \tag{6}$$

Now, let $\Pr(\text{decoder failure}|k \text{ errors})$ denote the probability that the decoder fails when the channel makes k errors. Since, a decoder failure occurs if and only if an instanton is present, we have (again assuming that the probability is small)

$$\Pr(\text{decoder failure}|k \text{ errors}) \approx \sum_{r=i}^{k} \Pr(r|k), \tag{7}$$

where i is the size of the smallest weight instanton. For sufficiently large values of k, using Monte-Carlo simulations, the relative frequencies of different instantons can be found and consequently $\Pr(r|k)$ for different r can be estimated. In fact, Eq. 6 does suggest such an approximate estimation, which subsequently can be used, with the help of Eq. 7, to estimate $\Pr(\text{decoder failure}|k \text{ errors})$ for intermediate values of k. This suggests the following estimation for FER at a given α

$$\text{FER}(\alpha) = \sum_{k=1}^{n} \Pr(\text{decoder failure}|k \text{ errors}) \Pr(k \text{ errors}),$$

and since the channel under consideration is the BSC, we arrive at

$$\Pr(k \text{ errors}) = \binom{n}{k} \alpha^k (1-\alpha)^{n-k}$$

Note that FER for large values of α is dominated by higher k. At large k fixed complexity estimate for $\Pr(\text{decoder failure}|k \text{ errors})$ can be found by recording the number of failures based on comparison with a set of predetermined patterns with k errors. This concludes explanation of how to estimate FER vs α in the transient regime.

Remark: Notice that while the number of instantons grows with size, the error floor performance is actually dominated by the instantons of the smallest size, these which are very rare. Hence, estimates made using the method described above are reliable only if the number of found small-size instantons is sufficiently large. This underlies importance of the ISA as an efficient method of small-size instanton discovery.

V. Numerical Results

We present instanton statistics for the following two codes: (1) (3,5) regular Tanner code of length 155 [15]; and (2) (3,6) regular random code of length 204 from MacKay's webpage [16]. Table I shows the number of instantons found by the ISA initiated with 20 flips and run for 10000 instances. The total number of instantons of each size as well as the total number of distinct instantons of each size are recorded [1]. It can be seen that the size of the smallest instanton is 5 for the Tanner code and 4 for the random MacKay code. Hence, the slope of the FER curve in the error floor region for the two codes is 5 and 4 respectively.

Table II shows the data corresponding to $\Pr(\text{decoder failure}|k \text{ errors})$ for the Tanner code and the MacKay code in the $k = [8; 20]$ range. For $k > 20$, we assume that $\Pr(\text{decoder failure}|k \text{ errors}) = 1$. Table III shows the relative frequencies of various weight instantons for the two codes. The results are obtained by simulating 10^7 error patterns with 8 errors for the Tanner code resulting in 331 decoder failures. Contributions of various instantons is found by comparing with the subsets of the 8 error patterns. Note that some error patterns may consist of multiple instantons and hence the estimates made are only approximate. For the Tanner code, one finds that there are approximately 2300 instantons of size 6, 6.4×10^5 instantons of size 7 and 3.8×10^7 instantons of size 8. For the MacKay code (analyzing the 87 error events obtained by simulating 623385 error patterns), one finds approximately 1120 instantons of size 5, 1.9×10^5 instantons of size 6, 1.2×10^7 instantons of size 7 and 1.66×10^9 instantons of size 8.

Fig. 1(a) shows comparison of the FER curves obtained using the semi-analytical approach described above and the Monte-Carlo simulations for the Tanner code. It is clear from the plots that our method is very accurate. Fig. 1(b) shows the corresponding plots for the MacKay code. The predicted performance curve labeled 1 is based on the statistics shown in Table II, while the curve labeled 2 uses instanton statistics obtained by analyzing a total of 713 error patterns of weight 8. The close agreement of the two curves suggest that even approximate statistics for higher weight instantons are sufficient. The plots also show the predicted performance at the values of α which are beyond the reach of the Monte-Carlo simulations. The FER in this region is dominated by the smallest weight instantons and the calculated slopes agree with the theoretical prediction. Finally, we note that Tables I, II, III are sufficient to estimate FER for the Tanner code and the MacKay code.

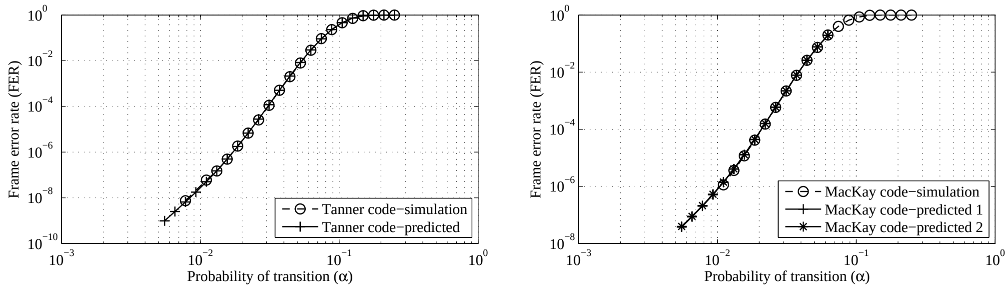
[1] The standard way to find out whether our instanton search exhausted all the unique configurations is as follows. Assume that there are N unique instantons of a given weight and in each trial ISA finds all of them with equal probability. To estimate the number of ISA runs required for finding all the N instantons, one notice that if $N-1$ instantons are already found the number of trails required to find to the last instanton is $\approx N$. If all but two instantons are already found the number of ISA trials required is $N/2$. Therefore, the average number of ISA trials required to find all the instantons is estimated as $N + N/2 + N/3 + \cdots N/(N-1) + 1 = N(1+1/2+1/3+\cdots+1/N)$ turning to $N \ln N$ at $N \to \infty$, i.e. $N \ln N$ trials ISA reliably finds N instantons. From this discussion, it is clear that the statistics for smallest size instantons for both the codes are very reliable.

TABLE I
INSTANTON STATISTICS OBTAINED BY RUNNING THE ISA WITH 20 RANDOM FLIPS AND 10000 INITIATIONS FOR THE TANNER CODE AND THE MACKAY CODE.

Code		Number of instantons of weight									
		4	5	6	7	8	9	10	11	12	13
Tanner code	Total		3506	1049	1235	1145	1457	1024	369	66	7
	Unique		155	675	1028	1129	1453	1024	369	66	7
MacKay code	Total	213	749	2054	2906	2418	1168	332	55	6	
	Unique	26	239	1695	2864	2417	1168	332	55	6	

TABLE II
$\Pr($DECODER FAILURE$|k$ ERRORS$)$ OBTAINED BY MONTE-CARLO SIMULATIONS.

Code	Number of Errors												
	8	9	10	11	12	13	14	15	16	17	18	19	20
Tanner code	3.3 e-5	1.2 e-4	5.3 e-4	2.2 e-3	7.7 e-3	2.6 e-2	7.5 e-2	0.178	0.358	0.582	0.806	0.932	0.985
MacKay code	1.4 e-4	5.1 e-4	1.9 e-3	6.2 e-3	1.9 e-2	5.5 e-2	0.124	0.265	0.449	0.674	0.853	0.947	0.991



(a) (b)

Fig. 1. Comparison between FER curves obtained using the semi-analytical approach and the Monte-Carlo simulations for (a) the Tanner code and (b) the MacKay code.

TABLE III
RELATIVE FREQUENCIES OF DIFFERENT SIZE INSTANTONS OBTAINED BY ANALYZING ERROR PATTERNS WITH 8 ERRORS.

Code	# error events	# instantons of weight				
		4	5	6	7	8
Tanner code	331		130	37	139	58
MacKay code	87	10	14	36	24	16

ACKNOWLEDGMENT

Part of the work by S. K. Chilappagari was performed when he was a summer GRA at LANL. The work at LANL, by S. K. Chilappagari and M. Chertkov, was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. B. Vasic and S. K. Chilappagari would like to acknowledge the financial support of the NSF (Grants CCF-0634969 and ECCS-0725405) and Seagate Technology. M. Stepanov would like to acknowledge the support of NSF grant DMS-0807592.

REFERENCES

[1] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

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

[3] J. Feldman, M. Wainwright, and D. Karger, "Using linear programming to decode binary linear codes," *Information Theory, IEEE Transactions on*, vol. 51, no. 3, pp. 954–972, March 2005.

[4] T. J. Richardson, "Error floors of LDPC codes," in *41st Annual Allerton Conf. on Communications, Control and Computing*, 2003, pp. 1426–1435. [Online]. Available: http://www.hpl.hp.com/personal/Pascal_Vontobel/pseudocodewords/papers

[5] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic, "Error floors of LDPC codes on the binary symmetric channel," in *International Conference on Communications*, vol. 3, June 11-15 2006, pp. 1089–1094.

[6] M. Chertkov, "Reducing the error floor," *Information Theory Workshop, 2007. ITW '07. IEEE*, pp. 230–235, Sept. 2007.

[7] M. Chertkov and M. Stepanov, "An efficient pseudocodeword search algorithm for linear programming decoding of LDPC codes," *Information Theory, IEEE Transactions on*, vol. 54, no. 4, pp. 1514–1520, April 2008.

[8] S. K. Chilappagari, M. Chertkov, and B. Vasic, "Provably efficient instanton search algorithm for LP decoding of LDPC codes over the BSC," 2008, submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: http://arxiv.org/abs/0808.2515

[9] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. of the 3rd Intern. Conf. on Turbo Codes and Related Topics*, Sept. 1-5 2003, pp. 75–82.

[10] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite length analysis of message-passing iterative decoding of LDPC codes," Dec. 2005. [Online]. Available: http://arxiv.org/abs/cs.IT/0512078

[11] C. Kelley and D. Sridhara, "Pseudocodewords of Tanner graphs," *Information Theory, IEEE Transactions on*, vol. 53, no. 11, pp. 4013–4038, Nov. 2007.

[12] G. D. Forney, R. Koetter, F. R. Kschischang, and A. Reznik, "On the effective weights of pseudocodewords for codes defined on graphs with cycles," in *In Codes, systems and graphical models*. Springer, 2001, pp. 101–112.

[13] M. G. Stepanov, V. Chernyak, M. Chertkov, and B. Vasic, "Diagnosis of weaknesses in modern error correction codes: A physics approach," *Phys. Rev. Lett.*, vol. 95, p. 228701, Nov. 2005.

[14] M. Ivkovic, S. K. Chilappagari, and B. Vasic, "Eliminating trapping sets in low-density parity-check codes by using Tanner graph covers," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3763–3768, 2008. [Online]. Available: http://dx.doi.org/10.1109/TIT.2008.926319

[15] R. M. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," in *ISCTA*, 2001. [Online]. Available: http://www.soe.ucsc.edu/~tanner/isctaGrpStrLDPC.pdf

[16] D. J. C. Mackay, "Encyclopedia of sparse graph codes." [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html