LA-UR- 09-00307

Approved for public release;
distribution is unlimited.

Title: Analysis of Error Floor of LDPC Codes under LP Decoding over the BSC

Author(s): Michael Chertkov/149600/T-4/LANL
Shashi Chilappagari/University of Arizona
Bane Vasic/University of Arizona
Mikhail Stepanov/University of Arizona

Intended for: International Symposium on Information Theory 2009

Los Alamos
NATIONAL LABORATORY
EST. 1943

For Proceedings of ISIT (Intern. Symp. on Information Theory) '09

for LAUR cost code: X9HN

Analysis of Error Floor 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@ece.arizona.edu

Michael Chertkov
Theoretical Divison
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 a semi-analytical method to predict the performance of LP decoding over a large range of transition probabilities of the BSC.**

I. Introduction

In this paper, we consider pseudo-codewords [1] and instantons of the LP decoder [1] for the BSC. We define the *BSC-instanton* as a noise configuration which the LP decoder decodes 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. Being a close relative of the BP decoder (see [2], [3] for discussions of different aspects of this relation), the LP decoder appeals due to the following benefits: (a) it has maximum-likelihood (ML) certificate i.e., if the output of the decoder is a codeword, then the ML decoder is also guaranteed to decode into the same codeword; (b) the output of the LP decoder is discrete even if the channel noise is continuous (meaning that problems with numerical accuracy do not arise); (c) its analysis is simpler due to the readily available set of powerful analytical tools from the optimization theory; and (d) it allows systematic sequential improvement, which results in decoder flexibility and feasibility of an LP-based ML for moderately large codes [4], [5]. While slower decoding speed is usually cited as a disadvantage of the LP decoder, this potential problem can be significantly reduced, thanks to the recent progress in smart sequential use of LP constraints [6] and/or appropriate graphical transformations [5], [7], [8].

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 probabilities of transition of the BSC.

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 of the pseudo-codeword weight, medians and instantons (defined as special set of flips), their costs, and we also prove some set of useful lemmata emphasizing the significance of the instanton analysis. In Section IV, we describe the ISA and prove our main result concerning bounds on the number of iterations required to output an instanton.

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 [1], and thus the interested reader is advised to refer to [1] for more details.

Let C be a binary LDPC code defined by a Tanner graph G with two sets of nodes: the set of variable nodes $V = \{1, 2, \dots, n\}$ and the set of check nodes $C = \{1, 2, \dots, m\}$. The adjacency matrix of G is H, a parity-check matrix of C, with m rows corresponding to the check nodes and n columns corresponding to the variable nodes. A binary vector $\mathbf{c} = (c_1, \dots, c_n)$ is a codeword *iff* $\mathbf{c}H^T = \mathbf{0}$. The support of a vector $\mathbf{r} = (r_1, r_2, \dots, r_n)$, denoted by $\text{supp}(\mathbf{r})$, 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 C allows a convenient LP formulation in terms of the *codeword polytope* $\text{poly}(C)$ whose vertices correspond to the codewords in C. The ML-LP decoder finds $\mathbf{f} = (f_1, \dots, f_n)$ minimizing the cost function $\sum_{i=1}^{n} \gamma_i f_i$ subject to the $\mathbf{f} \in \text{poly}(C)$ constraint. The formulation is compact but impractical because of the number of constraints 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

$$0 \le f_i \le 1, \quad \forall i \in V \qquad (1)$$

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

$$0 \le w_{j,T} \le 1, \qquad \forall T \in E_j \qquad (2)$$

$$\sum_{T \in E_j} w_{j,T} = 1 \qquad (3)$$

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

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 [1]. It was also shown in [1] 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: [1] *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 $0 \le p_i \le 1, \forall i \in V$, 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 proofs of all the lemmata and theorems can be found in the extended version of this paper [11].

The polytope Q is symmetric and looks exactly the same from all codewords (see e.g. [1]). 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. 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

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

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 $C(\mathbf{r}, \mathbf{p}) \le 0$.

Definition 3: [12], [13, 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

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

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 $e = \lceil (w_{BSC}(\mathbf{p}) + 1)/2 \rceil$ 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. [13] 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 $e (= \lceil (w_{BSC}(\mathbf{p}) + 1)/2 \rceil)$ largest components of $\mathbf{p}$.

One observers that, $C(M(\mathbf{p}), \mathbf{p}) \le 0$. From the definition of $w_{BSC}(\mathbf{p})$, it follows that at least one median exists for every $\mathbf{p}$. Also, all medians of $\mathbf{p}$ have $\lceil (w_{BSC}(\mathbf{p}) + 1)/2 \rceil$ flips. The proofs of the following two lemmata are now apparent.

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* $w_{BSC}(\mathbf{p}) \le 2k$.

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

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 $w_{BSC}(\mathbf{p}_M) \leq w_{BSC}(\mathbf{p})$. 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 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. It can be easily verified that if $\mathbf{c}$ is the transmitted codeword and $\mathbf{r}$ is the received vector such that $\text{supp}(\mathbf{c}+\mathbf{r}) = \text{supp}(\mathbf{i})$, where the addition is modulo two, then there exists a pseudo-codeword $\mathbf{p}'$ such that $C(\mathbf{r}, \mathbf{p}') \leq C(\mathbf{r}, \mathbf{c})$.

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.

The most general form of the above lemma can be stated as following: if $\mathbf{c}$ is the transmitted codeword and $\mathbf{r}$ is the received vector, then $\mathbf{r}$ converges to a pseudo-codeword different from $\mathbf{c}$ *iff* the $\text{supp}(\mathbf{r}+\mathbf{c})$, where the addition is modulo two, 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 illustarted in Section V). In this sense, the instantons are more fundamental than the minimal pseudo-codewords [14], [12] for the BSC (note, that this statement does not hold in the case of the AWGN channel). Two minimal pseudo-codewords of the same weight can give rise to different number of instantons. This issue was first pointed out by Forney *et al.* in [13]. (See Examples 1, 2, 3 for the BSC case in [13].) It is also worth noting that an instanton converges to a minimal pseudo-codeword.

IV. Instanton Search Algorithm and Performance Analysis

A. ISA

In this Section, we describe the Instanton Search Algorithm. The algorithm 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:

- $w_{BSC}(\mathbf{p}_{M_l}) < w_{BSC}(\mathbf{p}^l)$. Then $\mathbf{p}^{l+1} = \mathbf{p}_{M_l}$ becomes the l-th step output/$(l+1)$ step input.
- $w_{BSC}(\mathbf{p}_{M_l}) = w_{BSC}(\mathbf{p}^l)$. 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 states that the ISA terminates (i.e., outputs an instanton) in the number of steps of the order the number of flips in the initial noise configuration.

Theorem 1: $w_{BSC}(\mathbf{p}^l)$ 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

The FER at a given α can be estimated using

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

Since the channel under consideration is the BSC, we have

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

Note that the quantity $\Pr(\text{decoder failure}|k \text{ errors})$ is independent of α. It can be shown that as $\alpha \to 0$, the FER is dominated by the smallest k for which $\Pr(\text{decoder failure}|k \text{ errors}) \neq 0$ or in other words the smallest weight instanton (see [15] for a formal description of this relation). In fact, on a log-log scale, the slope of the FER curve in the asymptotic limit is equal to the size of the smallest weight instanton. Hence, the instanton statistics can be used to predict the FER performance for small values of α.

For large values of α, the FER is dominated by higher k. The values of $\Pr(\text{decoder failure}|k \text{ errors})$ for large k can be estimated using Monte-Carlo simulations with very good accuracy. These estimates can be made with a fixed complexity i.e., by running a predetrmined number of pattern with k errors. Hence, the FER for large values of α can also be predicted using such Monte-Carlo data. .

The region in which it is the most difficult to predict the performance is for intermediate values of α. The value of $\Pr(\text{decoder failure}|k \text{ errors})$ for interemediate k cannot be obtained by Monte-Carlo as it requires a very high complexity. Analytical estimates cannot be made as the instanton statistics for higher weight instantons are not complete. 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

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

Since, a decoder failure occurs if and only if an instanton is present, we have

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

where i is the size of the smallest weight instanton. For a sufficiently large value 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. Using Eq. 6, the values of T_r can be estimated approximately. These statistics can then be used to estimate $\Pr(\text{decoder failure}|k \text{ errors})$ for intermediate values of k using Eq. 7.

It should be noted that while there are a large number of instantons of large size, the error floor performance is dominated by the instnatons of smallest size which are very rare. Hence, estimates made using the above method may not be very raliable. This fact underlies the importance of the ISA which is successful in finding the smallest weight instantons.

V. Numerical Results

We first present the instanton statistics for the following two codes (1) The (3,5) regular Tanner code of length 155 [16] and (2) A (3,6) regular random code of length 204 from MacKay's webpage [17].

Table II shows the data for the Tanner and the MacKay code from $k = 8$ to $k = 20$. For $k < 20$, we can assume that $\Pr(\text{decoder failure}|k \text{ errors}) = 1$. Table III shows the relative frequencies of various weight instantons for the

TABLE III
INSTANTON STATISTICS

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

Tanner code and the MacKay code. The results are obtained by simulating 10^7 error patterns with 8 errors for the Tanner code resulting in 331 decoder failures. The contributions of various insstantons is found by examining the subsets of the 8 error patterns and finding the instantons. Note that some error patterns can consist multiple instantons and hene the estimates made are only approximate. For the Taner code, it is found that there are approximately 2300 instantons of size 6, 6.4×10^5 instantons of size 7 and 3.8×10^7 instantosn of size 8. For the MacKay code, it is found that there are approximately 1120 instantons of size 5, 1.6×10^5 instantons of size 6, 9.2×10^6 instantons of size 7 and instantons of size 8. Fig. 1(a) and Fig. 1(b) show the comparision between the FER curves obtained using the semi-analytical approach described above and the Monte-Carlo simulations. It is clear from the plots that the proposed method predicts the performance accurately.

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. The authors would like to thank A. R. Krishnan for providing the modified code.

References

[1] 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.

[2] M. J. Wainwright and M. I. Jordan, "Variational inference in graphical models: the view from the marginal polytope," in *Proc. of the 40th Allerton Conf. on Communications, Control, and Computing*, October 1-3, 2003 2003. [Online]. Available: http://www.hpl.hp.com/personal/Pascal_Vontobel/pseudocodewords/papers

[3] M. Chertkov and V. Chernyak, "Loop calculus helps to improve belief propagation and linear programming decodings of low-density-parity-check codes," in *Proc. of the 44th Annual Allerton Conf. on Communications, Control and Computing*, 2006. [Online]. Available: http://arxiv.org/abs/cs/0609154

[4] S. C. Draper, J. S. Yedidia, and Y. Wang, "ML decoding via mixed-integer adaptive linear programming," in *Proc. of IEEE International Symposium on Information Theory*, June 2007, pp. 1656–1660.

[5] K. Yang, X. Wang, and J. Feldman, "Fast ML decoding of SPC product code by linear programming decoding," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, Nov. 2007, pp. 1577–1581.

[6] M. H. Taghavi and P. Siegel, "Adaptive methods for linear programming decoding." [Online]. Available: http://arxiv.org/abs/cs/0703123v1

TABLE I

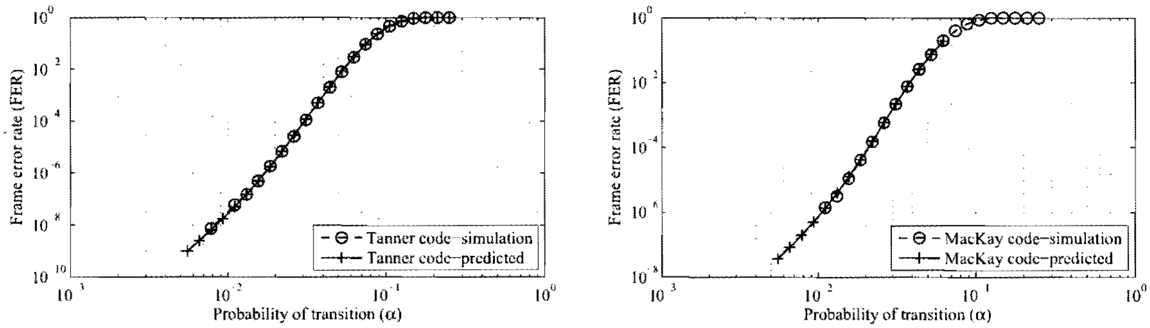
DIFFERENT CODES AND THEIR PARAMETERS

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

DIFFERENT CODES AND THEIR PARAMETERS

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

[7] A. G. Dimakis and M. J. Wainwright, "Guessing facets: Polytope structure and improved LP decoder," in *Proc. of IEEE International Symposium on Information Theory*, July 2006, pp. 1369–1373.

[8] M. Chertkov and M. Stepanov, "Pseudo-codeword landscape," in *Proc. of IEEE International Symposium on Information Theory*, June 2007, pp. 1546–1550.

[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] 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

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

[13] 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.

[14] R. Smarandache and P. Vontobel, "Pseudo-codeword analysis of Tanner graphs from projective and Euclidean planes," *Information Theory, IEEE Transactions on*, vol. 53, no. 7, pp. 2376–2393, July 2007.

[15] 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

[16] 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

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