An efficient instanton search algorithm for LP decoding of LDPC codes over the BSC

Shashi Kiran Chilappagari, *Member, IEEE,* Michael Chertkov, *Member, IEEE* and Bane Vasic, *Senior Member, IEEE*

***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 equal to the transmitted codeword. We design an efficient algorithm termed the Instanton Search Algorithm (ISA) which generates an error vector called the BSC-instanton. We prove that: (a) the LP decoder fails for any error pattern with support that is a superset of the support of an instanton; (b) for any input, the ISA outputs an instanton in the number of steps upper-bounded by twice the number of errors in the input error vector. We then find the number of unique instantons of different sizes for a given LDPC code by running the ISA sufficient number of times.**

***Index Terms*—Low-density parity-check codes, Linear Programming Decoding, Binary Symmetric Channel, Pseudo-Codewords, Error-floor**

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]. Iterative decoders operate by passing messages along the edges of a graphical representation of a code known as the Tanner graph [3], and are optimal when the underlying graph is a tree. However, the decoding becomes sub-optimal in the presence of cycles, and hence the asymptotic analysis methods are of limited practical use for the analysis of a fixed code. The linear programming (LP) decoding introduced by Feldman *et al.* [4], 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 *water-fall* region. The decrease slows down at moderate values turning into the *error-floor* asymptotic at very large SNR [5]. This transient behavior and the error-floor asymptotic originate from the sub-optimality of the decoding, i.e., the ideal maximum-likelihood (ML) curve would not show such a dramatic change in the BER/FER with the SNR increase.

After the initial investigation of error floors of LDPC codes on channels other than the binary erasure channel (BEC) by Richardson [5], 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. It is worth noting here that most of the analytical methods developed in the theory of iterative decoding have focused on ensembles of codes rather than a given fixed code.

The failures of iterative decoding over the BEC are well understood in terms of combinatorial objects known as stopping sets [6]. For iterative decoding on the Additive White Gaussian Noise (AWGN) channel and the BSC, the decoding failures have been characterized in terms of trapping sets [5], [7] and pseudo-codewords [8], [9], [10]. Richardson [5] introduced the notion of trapping sets and proposed a semi-analytical method to estimate the FER performance of a given code on the AWGN channel in the error floor region. The method was successfully applied to hard decision decoding over the BSC in [7]. The approach of [5] was further refined by Stepanov *et al.* [11], using *instantons*. Pseudo-codewords were first discussed in the context of iterative decoders using computation trees [8] and later using graph covers [9], [10]. Pseudo-codeword distributions were found for the special cases of codes from Euclidean and projective planes [12]. A detailed analysis of the pseudo-codewords was presented by Kelley and Sridhara [13], who discussed the bounds on pseudo-codeword size in terms of the girth and the minimum left-degree of the underlying Tanner graph. The bounds were further investigated by Xia and Fu [14]. Pseudo-codeword analysis has also been extended to the convolutional LDPC codes by Smarandache *et al.* [15]. (See also [16] for an exhaustive list of references for this and related subjects.)

Pseudo-codewords can be also used to understand the failures of the LP decoder [4]. It was shown in [4] that the LP decoding on the BEC fails if the set of erased variable nodes contain a stopping set. Hence, in this sense, the pseudo-codewords for the LP decoder are equivalent to stopping

Manuscript submitted September 2, 2008 and revised July 12, 2010 and October 25, 2010. Part of the work was presented in the IEEE International Symposium on Information Theory (ISIT), July, 2008, Seoul, Korea. Part of the work also appeared in the special issue of Journal of Selected Areas in Communications on Capacity Approaching Codes, published in August 2009.

S. K. Chilappagari [shashickiran@gmail.com] was with the ECE Department, University of Arizona, Tucson, AZ, 85721, USA. He is currently with Marvell Semiconductor Inc, Santa Clara, CA, 95054, USA.

M. Chertkov [chertkov@lanl.gov] is with Theory Division & CNLS, LANL, Los Alamos, NM, 87545 USA.

B. Vasic [vasic@ece.arizona.edu] is with the ECE Department, University of Arizona, Tucson, AZ, 85721, USA.

sets for the case of the BEC. For binary-input memoryless channels, the pseudo-codewords of the LP decoder are related to the pseudo-codewords arising from graph covers [10]. In fact, in [10] Vontobel and Koetter have also pointed out relations between pseudo-codewords arising from graph covers and trapping sets.

Closely related to the pseudo-codewords and the trapping sets are the noise configurations that lead to decoding failures which are termed as instantons [11]. Finding the instantons is a difficult task which so far admitted only heuristic solutions [7], [17]. 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 [18] (with AWGN channel used as an enabling example). Given a sufficiently strong random input, the outcome of the PCS algorithm is an instanton. The resulting distribution of the instantons (or respective pseudo-codewords) thus provides a compact and algorithmically feasible characterization of the AWGN-LP performance of the given code.

In this paper, we consider pseudo-codewords and instantons of the LP decoder 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 [19], [20] for discussions of different aspects of this relation), the LP decoder appeals due to the following benefits: (a) it has 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 [21], [22]. 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 [23] and/or appropriate graphical transformations [22], [24], [25] and other low complexity decoding approximations [26].

The two main contributions of this paper are: (1) characterization of all the failures of the LP decoder over the BSC in terms of the instantons, and (2) an efficient Instanton Search Algorithm (ISA). Following the idea by Chertkov and Stepanov [18], for a given a random binary n-tuple, the ISA generates a BSC-instanton, that is guaranteed to be decoded by the LP decoder into a pseudo-codeword distinct from the all-zero-codeword. Our ISA constitutes a significantly stronger algorithm than the one of [18] 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. An overview of instanton based techniques to analyze and reduce error floors of LDPC codes is presented in [27]. While the ISA has also been discussed in [27], the material in this paper presents a detailed analysis of the ISA along with the required theorems and proofs (that do not appear in [27]). Furthermore, this paper also elucidates the method to estimate the FER performance of a given LDPC code using the instanton statistics.

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. We comment on the analytical estimation of the FER using instanton statistics in Section V. We present an illustration of the ISA as applied to the $[155, 64, 20]$ Tanner code [28] and numerical results in Section VI. We summarize our results and conclude by listing some open problems in Section VII.

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 [4], and thus the interested reader is advised to refer to [4] 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 bi-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. In other words, $H_{i,j} = 1$ if and only if there is an edge between the check node i and the variable node j in the Tanner graph G. 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 $\mathrm{supp}(\mathbf{r})$, 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 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* $\mathrm{poly}(\mathcal{C})$ 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 $\mathbf{f} \in \mathrm{poly}(\mathcal{C})$ 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

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

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 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 [4]. It was also shown in [4] 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 equal to the transmitted codeword. For a more detailed description of the LCLP and the interpretation of the different variables in the above equations, the reader is referred to [4].

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

Definition 1: [4] *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 polytopes. Hence, we refer to the pseudo-codewords as corresponding to a particular Tanner graph G of $\mathcal{C}$.

It is also appropriate to mention here that the LCLP can be viewed as the zero temperature version of BP-decoder looking for the global minimum of the so-called Bethe free energy functional [19].

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 prove here some preliminary lemmata which will enable subsequent discussion of the ISA in the next Section.

When the channel and the decoder satisfy certain symmetry conditions (see [2] for details), we can assume, without loss of generality, that the all zero codeword is transmitted. The LP decoder satisfies these conditions as shown in [4]. Hence, we make the assumption of the all-zero-codeword throughout the paper. Hence, the received vector and subsequently the input to the LP decoder is the error vector. 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 an error vector with support of size k as having k flips.

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

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

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 output of the LP decoder on $\mathbf{r}$ is 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, the output of LP decoding of a binary vector $\mathbf{r}$ is not equal to the all-zero-codeword if there exists a pseudo-codeword $\mathbf{p}$ with $C(\mathbf{r}, \mathbf{p}) \leq 0$.

Definition 3: [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

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

where $E(\mathbf{p})$ is a set of e largest components of $\mathbf{p}$.

The minimum pseudo-codeword weight of G denoted by w_{BSC}^{min} is the minimum over all the non-zero pseudo-codewords of G. The parameter $e = \lfloor (w_{BSC}(\mathbf{p}) + 1)/2 \rfloor$ 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. [29] for a number of illustrative examples.)

Remark: Feldman *et al.* in [4] defined *weight* of a pseudo-codeword, the *fractional distance* and the *max-fractional distance* of a Tanner graph of a code in terms of the projected polytope $\overline{Q}$ (the interested reader is referred to [4] for explicit description of $\overline{Q}$). To differentiate the two definitions, we term the "weight" defined by Feldman *et al.* as *fractional weight* and denote it by w_{frac}. For a point $\mathbf{f}$ in $\overline{Q}$, the fractional weight of $\mathbf{f}$ is defined as the L1-norm, $w_{frac}(\mathbf{f}) = \sum_{i \in V} f_i$

and the max-fractional weight of $\mathbf{f}$ is defined as the fractional weight normalized by the maximum f_i value i.e.,

$$w_{max-frac}(\mathbf{f}) = \frac{w_{frac}(\mathbf{f})}{\max_i f_i}.$$

Also, if $\mathcal{V}_{\overline{Q}}$ denotes the set of non-zero vertices of $\overline{Q}$ the fractional distance d_{frac} of the code is defined as the minimum weight over all vertices in $\mathcal{V}_{\overline{Q}}$. The max-fractional distance d_{frac}^{max} of a Tanner graph of the code is given by

$$d_{frac}^{max} = \min_{(\mathbf{f})\in\mathcal{V}_{\overline{Q}},\mathbf{f}\neq\mathbf{0}} \left(\frac{\sum_{i\in V} f_i}{\max_i f_i}\right)$$

It was shown in [4, Theorem 9] that the LP decoder is successful if at most $\lceil d_{frac}/2\rceil - 1$ bits are flipped by the BSC, thus making d_{frac} a potentially useful characteristic. Moreover, an efficient LP-based algorithm to calculate d_{frac} was suggested in [4]. However, LP decoding of the error pattern with the least number of flips which the LP decoder fails to correct does not necessarily output the pseudo-codeword with fractional weight d_{frac}. Hence, we adopted the definition of the pseudo-codeword weight from [13], however noticing that it was discussed there in a different but related context of the computation tree and graph covers. A slightly different definition also first appeared in [29]. The advantage of our approach will become evident in the subsequent Sections.

The following Lemma gives a relation between w_{BSC}^{min} and d_{frac}.

Lemma 1: $w_{BSC}^{min} \geq 2\lceil d_{frac}/2\rceil - 1$.

Proof: The LP decoder is successful if at most $\lceil d_{frac}/2\rceil - 1$ bits are flipped by the BSC. So, the minimum number of flips in the all-zero-codeword which can cause the LP decoder to fail is $\lceil d_{frac}/2\rceil$. If e is the minimum number of flips associated with the minimum weight pseudo-codeword, then

$$e \geq \lceil d_{frac}/2\rceil$$

Since, $w_{BSC}^{min} \geq 2e - 1$, we have $w_{BSC}^{min} \geq 2\lceil d_{frac}/2\rceil - 1$ ■

The above lemma can be generalized to any pseudo-codeword $\mathbf{p}$ as $w_{BSC}(\mathbf{p}) \geq 2\lceil w_{frac}(\mathbf{p})/2\rceil - 1$. We would like to point out that Kelley and Sridhara in [13] have derived a similar relation between $w_{BSC}(\mathbf{p})$ and $w_{max-frac}(\mathbf{p})$ and that Sridhara in [30] observed that $w_{BSC}(\mathbf{p})+1 \geq w_{max-frac}(\mathbf{p})$.

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 $e(= \lfloor (w_{BSC}(\mathbf{p})+1)/2\rfloor)$ largest components of $\mathbf{p}$.

One observes that, $C(M(\mathbf{p}), \mathbf{p}) \leq 0$. 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 $\lfloor (w_{BSC}(\mathbf{p})+1)/2\rfloor$ flips. The proofs of the following two lemmata are now apparent.

Lemma 2: Let the transmitted codeword be the all-zero-codeword and let $\mathbf{p}$ be the output of the LP decoder on an error vector with support of size k. If $\mathbf{p} \neq \mathbf{0}$, then $w_{BSC}(\mathbf{p}) \leq 2k$.

Lemma 3: 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 4: 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 $\mathbf{p}$.

Proof: Let $|S| = k$. Then by Lemma 3, $w_{BSC}(\mathbf{p}) \in \{2k-1, 2k\}$. Now, if $\mathbf{r}$ is any binary vector with support $S' \subset S$, then $\mathbf{r}$ has at most $k-1$ flips and therefore by Lemma 2, $w_{BSC}(\mathbf{p}) \leq 2(k-1)$, which is a contradiction. ■

Lemma 5: If the output of the LP decoder on $M(\mathbf{p})$ is a pseudo-codeword $\mathbf{p}_M \neq \mathbf{p}$, then $w_{BSC}(\mathbf{p}_M) \leq w_{BSC}(\mathbf{p})$. Also, $C(M(\mathbf{p}), \mathbf{p}_M) \leq C(M(\mathbf{p}), \mathbf{p})$.

Proof: According to the definition of the LP decoder, $C(M(\mathbf{p}), \mathbf{p}_M) \leq C(M(\mathbf{p}), \mathbf{p})$.

If $w_{BSC}(\mathbf{p}) = 2k$, then $M(\mathbf{p})$ has k flips and by Lemma 2, $w_{BSC}(\mathbf{p}_M) \leq 2k = w_{BSC}(\mathbf{p})$.

If $w_{BSC}(\mathbf{p}) = 2k-1$, then $M(\mathbf{p})$ has k flips and $C(M(\mathbf{p}), \mathbf{p}) < 0$. Hence, $w_{BSC}(\mathbf{p}_M) \leq 2k$ by Lemma 2. However, if $w_{BSC}(\mathbf{p}_M) = 2k$, then $C(M(\mathbf{p}), \mathbf{p}_M) = 0$, which is a contradiction. Hence, $w_{BSC}(\mathbf{p}_M) \leq 2k-1 = w_{BSC}(\mathbf{p})$. ■

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 supp($\mathbf{r}$) $\subset$ 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 $\mathbf{p} \neq \mathbf{0}$ 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 supp($\mathbf{c}+\mathbf{r}$) = 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 6: Let $\mathbf{i}$ be an instanton. Then for any binary vector $\mathbf{r}$ such that supp($\mathbf{i}$) $\subset$ supp($\mathbf{r}$), there exists a pseudo-codeword $\mathbf{p}$ satisfying $C(\mathbf{r}, \mathbf{p}) \leq 0$.

Proof: Since $\mathbf{i}$ is an instanton, there exists a pseudo-codeword $\mathbf{p}$ such that $C(\mathbf{i}, \mathbf{p}) \leq 0$. From Definition 2 we have,

$$\sum_{i\notin \text{supp}(\mathbf{i})} p_i - \sum_{i\in \text{supp}(\mathbf{i})} p_i \leq 0.$$

Since, supp($\mathbf{i}$) $\subset$ supp($\mathbf{r}$) and $p_i \geq 0, \forall i$, we have

$$\sum_{i\notin \text{supp}(\mathbf{r})} p_i - \sum_{i\in \text{supp}(\mathbf{r})} p_i \leq 0,$$

thus yielding

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

■

The above lemma implies that if the all-zero-codeword is transmitted over the BSC and the support of the received vector is a superset of an instanton, then the LP decoder fails to decode the received vector 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 LP decoding of all binary vectors with support $S' \subset S$ such that $|S'| = |S| - 1$, outputs $\mathbf{0}$, 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 7: Let the transmitted codeword be the all-zero-codeword and let $\mathbf{r}$ be a binary error vector such that the output of LP decoding on $\mathbf{r}$ is a pseudo-codeword different from the all-zero-codeword. Then, 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 result of LP decoding on $\mathbf{r}$ is a pseudo-codeword different from $\mathbf{c}$ *iff* the 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 7 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. In this sense, the instantons are more fundamental than the minimal pseudo-codewords [12], [13] 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 [29]. (See Examples 1, 2, 3 for the BSC case in [29].) It is also worth noting that the result of LP decoding on an instanton is a minimal pseudo-codeword.

It should be noted that finding pseudo-codewords with fractional weight d_{frac} is not equivalent to finding minimum weight pseudo-codewords. The pseudo-codewords with fractional weight d_{frac} can be used to derive some instantons, but not necessarily the ones with the least number of flips. However, as d_{frac} provides a lower bound on the minimum pseudo-codeword weight, it can be used as a test if the ISA actually finds an instanton with the least number of flips. In other words, if the number of flips in the lowest weight instanton found by the ISA is equal to $\lceil d_{frac}/2 \rceil$, then the ISA has indeed found the smallest size instanton.

IV. Instanton Search Algorithm and its Analysis

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 5, 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 4 guarantees that any $\mathbf{p}_{i_t} \neq \mathbf{p}^l$, thus preventing the ISA from entering into an infinite loop.)

Fig. 1 illustrates different scenarios arising in the execution of the ISA. Here, the squares represent pseudo-codewords and the circles represent binary vectors (noise configurations). Two squares of the same color have identical pseudo-codeword weight and two circles of the same color consist of same number of flips. Fig. 1(a) shows the case where the result of LP decoding of a median, $M(\mathbf{p}^l)$, of a pseudo-codeword $\mathbf{p}^l$ is a pseudo-codeword $\mathbf{p}_{M_l}$ of a smaller weight. In this case, $\mathbf{p}^{l+1} = \mathbf{p}_{M_l}$. Fig. 1(b) illustrates the case where the LP decoding of a median, $M(\mathbf{p}^l)$, of a pseudo-codeword $\mathbf{p}^l$ outputs a pseudo-codeword $\mathbf{p}_{M_l}$ of the same weight. Fig. 1(c) illustrates the case where the LP decoding of a median, $M(\mathbf{p}^l)$, of a pseudo-codeword $\mathbf{p}^l$ outputs the pseudo-codeword $\mathbf{p}^l$ itself. In the two latter cases, we consider all the binary vectors whose support sets are subsets of the support set of $M(\mathbf{p}^l)$ and the vectors contain one flip less. We run the LP decoder with the vectors as inputs and find their corresponding pseudo-codewords. One of the non-zero pseudo-codewords found is chosen at random as $\mathbf{p}^{l+1}$. This is illustrated in Fig. 1(d). Fig. 1(e) shows the case when LP decoding of all the subsets of $M(\mathbf{p}^l)$ (reduced by one flip) outputs to the all-zero-codeword. LP decoding of $M(\mathbf{p}^l)$ itself could output $\mathbf{p}^l$ or some other pseudo-codeword of the same weight. In this case, $M(\mathbf{p}^l)$ is an instanton constituting the output of the algorithm.

We now prove 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.

Proof: If $\mathbf{p}^{l+1} = \mathbf{p}_{M_l}$, then $w_{BSC}(\mathbf{p}^{l+1}) < w_{BSC}(\mathbf{p}^l)$. Consequently, $|\text{supp}(M(\mathbf{p}^{l+1}))| \leq |\text{supp}(M(\mathbf{p}^l))|$.

If $\mathbf{p}^{l+1} = \mathbf{p}_{i_t}$, then $w_{BSC}(\mathbf{p}_{i_t}) \leq 2(|\text{supp}(M(\mathbf{p}^l))| - 1) < w_{BSC}(\mathbf{p}^l)$. Consequently, $|\text{supp}(M(\mathbf{p}^{l+1}))| \leq |\text{supp}(M(\mathbf{p}^l))|$.

Since $w_{BSC}(\mathbf{p}^j)$ is strictly decreasing, the weight of pseudo-codeword at step l decreases by at least one compared

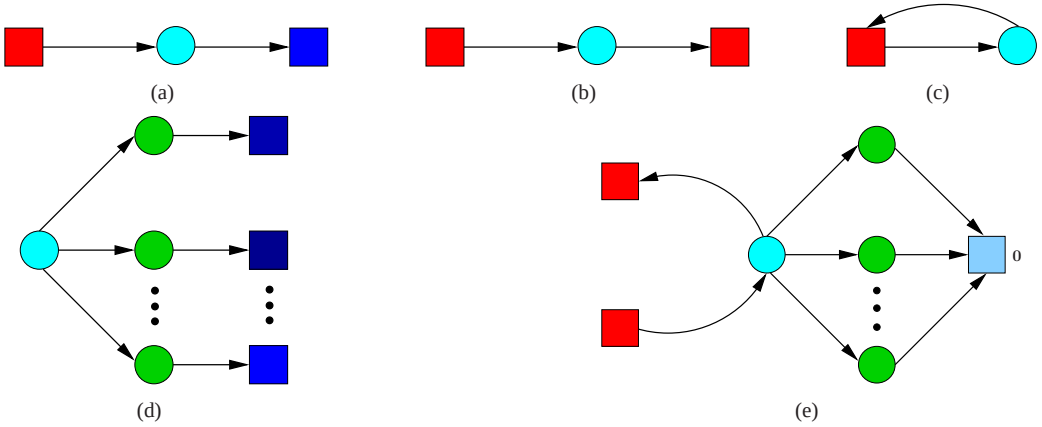

Fig. 1. Squares represent pseudo-codewords and circles represent medians or related noise configurations (a) LP decodes median of a pseudo-codeword into another pseudo-codeword of smaller weight (b) LP decodes median of a pseudo-codeword into another pseudo-codeword of the same weight (c) LP decodes median of a pseudo-codeword into the same pseudo-codeword (d) Reduced subset (three different green circles) of a noise configuration (e.g. of a median from the previous step of the ISA) is decoded by the LP decoder into three different pseudo-codewords (e) LP decodes the median (blue circle) of a pseudo-codeword (low red square) into another pseudo-codeword of the same weigh (upper red square). Reduced subset of the median (three configurations depicted as green circles are all decoded by LP into all-zero-codeword. Thus, the median is an instanton.

to the weight of the pseudo-codeword at step $l-1$. Since by Lemma 2, $w_{BSC}(\mathbf{p}^1) \leq 2k_0$, the algorithm can run for at most $2k_0$ steps. ■

Remarks: (1) By "sufficient number of flips", we mean that the initial binary vector should be noisy enough to for the LP decoder to output a pseudo-codeword other than the all-zero-codeword. While LP decoding of any binary vector with a large number of flips is almost guaranteed to output a pseudo-codeword different from the all-zero-codeword, such a choice might also lead to a longer running time of the ISA (from Theorem 1). On the other hand, choosing a binary vector with a few number of flips might lead to decoding to the all-zero-codeword very often, thereby necessitating the need to run the ISA for a large number of times.

(2) Theorem 1 does not claim that the algorithm finds the minimum weight pseudo-codeword or the instanton with the smallest number of flips. However, it is sometimes possible to verify if the algorithm has found the minimum weight pseudo-codeword. Let w_{ISA}^{min} denote the weight of the minimum weight pseudo-codeword found by the ISA. If $w_{ISA}^{min} = 2\lceil d_{frac}/2 \rceil - 1$, then $w_{ISA}^{min} = w_{BSC}^{min}$.

(3) At some step l, it is possible to have $w_{BSC}(\mathbf{p}_{M_l}) = w_{BSC}(\mathbf{p}^l)$ and incorporating such pseudo-codewords into the algorithm could lead to lower weight pseudo-codewords in the next few steps. However, this inessential modification was not included in the ISA to streamline the analysis of the algorithm.

(4) While we have shown that $w_{BSC}(\mathbf{p}^l)$ decreases by at least unity at every step, we have observed that in most cases, it decreases by at least two. This is due to the fact that the pseudo-codewords with odd weights outnumber pseudo-codewords with even weights. As a result, in most cases, the algorithm converges in less than k_0 steps. (For illustration of this point see example discussed in the next Section.)

(5) At any step, there can be more than one median, and the ISA does not specify which one to pick. Our current implementation suggests to pick a median at random. Also, the algorithm does not provide clarification on the choice of the pseudo-codeword for the case when more than one noise configurations from the subset $\mathbf{r}_{i_t}$ decode to pseudo-codewords distinct from the all-zero-codeword. In this degenerate case, we again choose a pseudo-codeword for the next iteration at random. Note that one natural deterministic generalization of the randomized algorithm consists of exploring all the possibilities at once. In such a scenario, a tree of solutions can be built, where the root is associated with one set of initiation flips, any branch of the tree relates to a given set of randomized choices (of medians and pseudo-codewords), and any leaf corresponds to an instanton.

V. Analytical Prediction of the FER

In [11], [31], 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 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 very good accuracy by Monte-Carlo simulations. The FER estimates in this region can be made with a fixed complexity (the details of which will be explained subsequently). The region in which it is the most difficult to predict the performance is for intermediate values of α. 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 instantiations 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

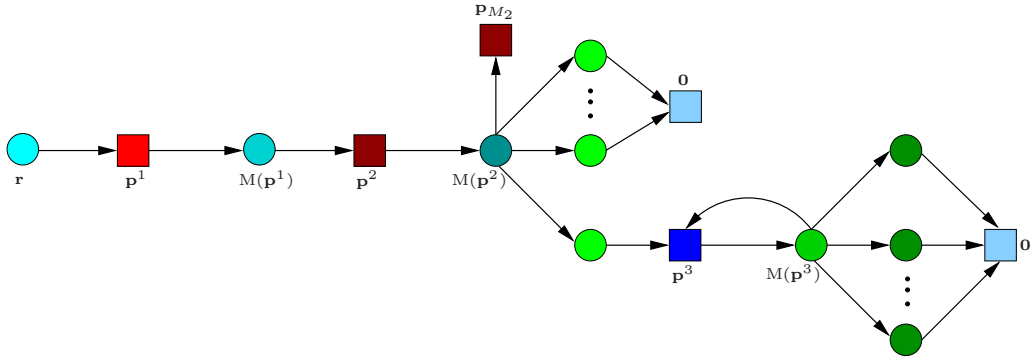

Fig. 2. Illustration for the example of ISA execution on the $[155, 65, 20]$ Tanner code discussed in Section VI.

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

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

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

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

where i is the size of the smallest weight instanton. For 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.

The FER at a given α can then 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 FER for large values of α is dominated by higher k. The values of $\Pr(\text{decoder failure}|k \text{ errors})$ for large k can be estimated with a fixed complexity by running a predetermined number of pattern with k errors and recording the number of failures. Hence, the FER over a large range of α can be estimated by the above approach.

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

VI. Numerical Results

In this Section, we present results illustrating different aspects and features of the ISA.

A. Illustration of the ISA

We use the $[155, 64, 20]$ Tanner code [28] for illustration purposes. We begin with an actual (and rather typical) example. The reader is advised to follow this example with an eye on Fig. 2.

Example 1: The algorithm is initiated with a binary vector $\mathbf{r}$ whose support set has cardinality 12. In this case, Lp decoding of $\mathbf{r}$ outputs a pseudo-codeword $\mathbf{p}^1$ of weight 17 (Lemma 2 guarantees that $w_{BSC}(\mathbf{p}^1) \leq 24$). The Median $M(\mathbf{p}^1)$ of the pseudo-codeword $\mathbf{p}^1$ has 9 flips. The output of LP decoding on $M(\mathbf{p}^1)$ is a pseudo-codeword $\mathbf{p}_{M_1}$ of weight 11, marked as $\mathbf{p}^2$, whose median $M(\mathbf{p}^2)$ contains 6 flips. $M(\mathbf{p}^2)$ decodes to a pseudo-codeword $\mathbf{p}_{M_2}$ of weight 11 and hence we consider all vectors whose support sets consist of one flip less than in the support set of $M(\mathbf{p}^2)$. There are 6 such vectors and 5 of them decode to the all-zero-codeword (we do not show all the six vectors in Fig .2). The remaining vector decodes to a pseudo-codeword of weight 9, marked as $\mathbf{p}^3$. The pseudo-codeword $\mathbf{p}^3$ has only one median $M(\mathbf{p}^3)$ which is decoded to the same pseudo-codeword $\mathbf{p}^3$. Hence, we consider all (five) vectors built from the median $M(\mathbf{p}^3)$ removing a single flip and observe that the LP decoder decodes all these vectors into the all-zero-codeword. We conclude that the median is actually an instanton of size 5.

B. Performance Prediction Results

We first present the instanton statistics for the following two codes (1) The (3,5) regular Tanner code of length 155 [28] and (2) A (3,6) regular random code of length 204 from MacKay's webpage [32]. Table I shows the number of instantons found when the ISA is initiated with 20 flips and run for 10000 different instantiations. The total number of instantons of each size as well as the total number of unique instantons of each

TABLE I
INSTANTON STATISTICS OBTAINED BY RUNNING THE ISA WITH 20 RANDOM FLIPS FOR 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
$\Pr(\text{DECODER FAILURE}|k \text{ ERRORS})$ 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

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 these codes is 5 and 4 respectively.

Note that the smallest weight instanton found by the ISA for the Tanner code is 5 (We have observed that all the instantons of size 5 are in fact the $(5,3)$ trapping sets described in [7]. Further investigation of the topological structure of instantons will be dealt with in future work). The accuracy of this estimate can be verified (indirectly) by finding the d_{frac} of the code. Using the method outlined in [4], we observed that d_{frac} of the Tanner code is 8.3498. This implies that $w_{BSC}^{min} \geq 9$ (by Lemma 1), which in turn implies that the size of any instanton cannot be less than 5. This proves that here 5 is, indeed, the smallest instanton size, and respective minimum pseudo-codeword weight is 9. Note also that the fractional weight of all the 155 pseudo-codewords of BSC weight 9 is 9.95, while the weight of the pseudo-codeword with the minimal fractional weight of 8.3498 is 19. The remark illustrates that minimality of the fractional weight does not imply minimality of the pseudo-codeword weight (and thus minimality of the respective instanton size).

Table II shows the data corresponding to $\Pr(\text{decoder failure}|k \text{ errors})$ 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 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 instantons 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 hence the estimates made are only approximate. For the Tanner 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 instantons 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. 3(a) and Fig. 3(b) show the comparison 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. 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 dominate by the smallest weight instantons and the calculated slopes agree with the theoretical prediction. It is worth noting that Tables I,II,III are sufficient to obtain the FER estimate for the Tanner and MacKay codes.

A Note on Other Possible Algorithms: One can imagine other simple algorithms that converge to an instanton in a finite number of steps. One such algorithm can be formulated in the following way [2].

1) Start with a binary input vector $\mathbf{r}$ with many bit flips, and let $\mathbf{p}$ be the output of the LP decoder applied to $\mathbf{r}$. If $\mathbf{p} = 0$, then start again.
2) For any i, let $\mathbf{s} = \mathbf{s}(i)$ be such that $\mathbf{s}_i = 0$ and $\mathbf{r}_i = 1$ for some i, and $\mathbf{s}_j = \mathbf{r}_j$ for all $j \neq i$. Let $\mathbf{p}(i)$ be the result of application of the LP decoder to $\mathbf{s}(i)$.
3) If there is i such that $\mathbf{p}(i) \neq 0$, then set $\mathbf{r} = \mathbf{s}(i)$ and go to step 2. Otherwise, $\mathbf{r}$ is minimal and so it is an instanton.

The above simple algorithm can also find an instanton in a finite number of steps. However, such an algorithm finds only instantons that are subsets of the support of the initial error vector. This implies that the algorithm needs to be run

[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 (this is also known as the "coupon collector's problem"). From this discussion, it is clear that the statistics for smallest size instantons for both the codes are very reliable.

[2]We would like to thank the anonymous reviewer for pointing out this simple alternative algorithm.

for a large number of instantiations to find a small weight instanton. As an illustration, consider the Tanner code that contains 155 instantons of weight five. The probability that an error pattern with $t \geq 5$ errors contains a weight-five instanton is approximately $155 \times \binom{t}{5}/\binom{155}{t}$. It follows that if $t = 20$, the above simple algorithm needs to be run about $\binom{155}{20}/\binom{155}{15}$ times (in the order of 10's of thousands) to find a weight-five instanton. In contrast, the ISA finds weight-five instantons 3506 times in 10000 trials, which implies that the ISA needs to be run only three times to find the lowest weight instanton. This clearly illustrates the advantage of the ISA in finding low weight instantons. The advantage of the ISA comes from the fact that given a pseudo-codeword, the median step finds the lowest weight error vector that could decode to the pseudo-codeword.

VII. Summary and Open Problems

In this paper, we characterized failures of the LP decoder over the BSC in terms of the instantons and respective pseudo-codewords. We then provided an efficient algorithm for finding the instantons. The ISA is guaranteed to terminate in the number of steps upper bounded by twice the number of flips in the original input (Theorem 1). Repeated sufficient number of times, the ISA outcomes the Instanton-Bar-Graph showing the number of unique instantons of different sizes. We also proved that the LP decoding of any configuration of the input noise which includes an instanton leads to a failure (Lemma 7). This Lemma arguably suggests to use the Instanton-Bar-Graph derived with the ISA algorithm as a metric for code optimization.

Finally, we conclude with an incomplete list of open problems and directions for future research following from this study:

(1) One would like to understand how to choose initiation of the ISA which guarantees convergence to the smallest size instanton.

(2) When can one be reasonably certain that all instantons of a given weight are found? Or stating it differently, how many trials of the ISA are required to find all the instantons of the given size? Does the number of trials scales linearly with the size of the code?

(3) We have noticed that difficulty of finding an instanton grows with its size. Once the ISA finds all the instantons of certain weight, can one optimize initiation strategy for the algorithm to find instantons of larger size more efficiently?

(4) Can one utilize knowledge of the code structure (e.g. for highly structured codes) to streamline discovery of the Instanton-Bar-Graph, especially in the part related to the larger size instantons?

(5) Some studies have explored connections between pseudo-codewords and stopping sets (see e.g. [13]). Are there any (similar?) relationships between trapping sets of the BSC (for Gallager like algorithms) and BSC-LP instantons?

(6) Are instantons of a code performing over the BSC related to instantons of the same code over the AWGN channel (or other soft channels)? Can we use one to deduce the other?

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

The authors would like to thank M. G. Stepanov for his help with the numerical simulation results. The authors would like to thank P. Vontobel for his comments and suggestions and D. Sridhara for his clarifications regarding Lemma 1. 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 would like to acknowledge the financial support of the NSF and Seagate Technology.

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] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.

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

[5] 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

[6] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke, "Finite length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1570–1579, June 2002.

[7] 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.

[8] N. Wiberg, "Codes and decoding on general graphs," Ph.D., Univ. Linköping, Sweden, Dept. Elec. Eng., 1996.

[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] 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.

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

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

[14] S.-T. Xia and F.-W. Fu, "Minimum pseudoweight and minimum pseudocodewords of LDPC codes," *IEEE Trans. Inform. Theory*, vol. 54, no. 1, pp. 480–485, Jan. 2008.

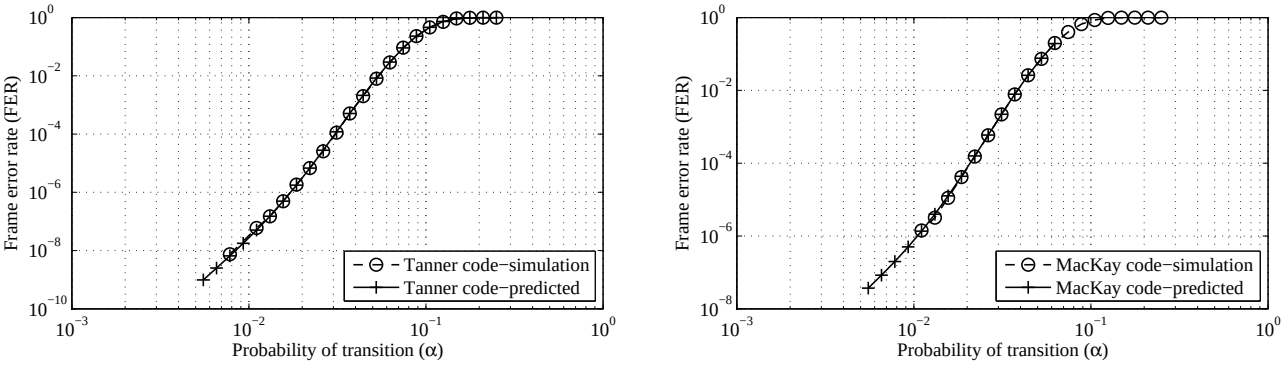

Fig. 3. Comparison between the FER curves obtained using the semi-analytical approach and the Monte-Carlo simulations for (a) the Tanner code and (b) the MacKay code

[15] R. Smarandache, A. E. Pusane, P. O. Vontobel, and D. J. Costello Jr, "Pseudo-codeword performance analysis for LDPC convolutional codes," 2006. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0609148

[16] P. O. Vontobel, "Papers on pseudo-codewords." [Online]. Available: http://www.pseudocodewords.info

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

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

[19] 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

[20] 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

[21] 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.

[22] 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.

[23] M. H. Taghavi and P. Siegel, "Adaptive methods for linear programming decoding," *IEEE Trans. Inform. Theory*, vol. 54, no. 12, pp. 5396–5410, Dec. 2008.

[24] 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.

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

[26] D. Burshtein, "Iterative approximate linear programming decoding of LDPC codes with linear complexity", in *Proc. IEEE International Symposium on Information Theory*, Toronto, Canada, pp. 1498-1502, July 2008.

[27] S. K. Chilappagari, M. Chertkov, M. G. Stepanov, and B. Vasic, "Instanton-based techniques for analysis and reduction of error floors of LDPC codes," *IEEE J. Sel. Areas in Commun.*, vol. 27, no. 6, pp. 855–865, Aug. 2009.

[28] R. M. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," in *ISCTA*, 2001.

[29] 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.

[30] D. Sridhara, Personal communication, August 2008.

[31] 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.

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