Searching for low weight pseudo-codewords

Michael Chertkov
Theoretical Division and Center for Nonlinear Studies
LANL, MS B213, T-13, Los Alamos, NM 87545
chertkov@lanl.gov

Mikhail Stepanov
Department of Mathematics, The University of Arizona
617 N. Santa Rita Ave., Tucson, AZ 85721
stepanov@math.arizona.edu

***Abstract*— Belief Propagation (BP) and Linear Programming (LP) decodings of Low Density Parity Check (LDPC) codes are discussed. We summarize results of instanton/pseudo-codeword approach developed for analysis of the error-floor domain of the codes. Instantons are special, code and decoding specific, configurations of the channel noise contributing most to the Frame-Error-Rate (FER). Instantons are decoded into pseudo-codewords. Instanton/pseudo-codeword with the lowest weight describes the largest Signal-to-Noise-Ratio (**SNR**) asymptotic of FER, while the whole spectra of the low weight instantons is descriptive of the FER vs** SNR **profile in the extended error-floor domain. First, we describe a general optimization method that allows to find the instantons for any coding/decoding. Second, we introduce LP-specific pseudo-codeword search algorithm that allows efficient calculations of the pseudo-codeword spectra. Finally, we discuss results of combined BP/LP error-floor exploration experiments for two model codes.**

I. Introduction

Low-Density-Parity-Check (LDPC) codes [1], [2] are special, not only because they can approach virtually error-free transmission limit, but mainly because a computationally efficient iterative decoding scheme, the celebrated Belief Propagation (BP) decoding, is readily available. For an idealized code on a tree, the BP algorithm is exactly equivalent to the symbol-MAP decoding, which is reduced to block-MAP (or simply Maximum Likelihood, ML), in the asymptotic limit of infinite SNR. For any realistic code (with loops), the BP algorithm is approximate, and it should actually be considered as an algorithm solving iteratively nonlinear equations, called BP equations. The BP equations describe extrema (e.g. minima are of main interest) of the Bethe free energy [3]. Minimizing the Bethe free energy, that is a nonlinear functional of the probabilities/beliefs, under the set of linear (compatibility and normalization) constraints, is generally a difficult task.

Linear Programming decoding, introduced in [4], is a close relative of BP which can be viewed as a relaxed version of Maximum Likelihood (ML) decoding. Relation of the LP decoding to the Bethe free energy approach [3], and thus to BP equations and decoding, was noticed in [4], and the point was elucidated further in [5], [6], [7], [8], [9], [10]. In short, LP may be considered as large SNR asymptotic limit of BP, where the later is interpreted as an extremum of the Bethe free energy functional.

Both BP and LP are computationally efficient but suboptimal, i.e. incapable of matching performance of the Maximum-Likelihood (ML). Performance of an error-correcting scheme can be measured in terms of the Frame Error Rate (FER) dependence on the Signal-to-Noise Ratio (SNR). FER decreases as SNR increases. Even though BP and LP decodings are suboptimal with respect to ML at all SNRs, the difference in FER is only order one in the water-fall regime of small SNRs. The situation becomes significantly worse in the error-floor domain of moderate to large SNRs where FER for BP/LP is parametrically, i.e. orders of magnitude, larger than FER for ML. Length of the error-correction code brings another dimension into the problem. The longer the code the lower is the value of FER where the water-fall-to-error-floor transition happens. On the other hand, standard Monte-Carlo (MC) numerics is incapable to determine BER below 10^{-9}. Therefore, understanding and describing the error-floor by an alternative, and hopefully more insightful, method is in great demand [11].

One such useful insight came through recent efforts [12], [13], [14], [8], [10] to understanding error-floor in terms of the most probable of the dangerous configurations of the noise, so-called instantons, contributing most to FER. BP/LP decodes the instantons into the so-called non-codeword pseudo-codewords [15], [16], [17], [11], [5]. It was recognized that for moderate and large SNRs splitting of the two (FER vs SNR) curves, representing ML decoding and approximate BP/LP decoding, is due to the pseudo-codewords, which are confused by the suboptimal algorithm for actual codewords of the code. Describing BP/LP error-floor translates into finding pseudo-codewords with low effective distance.

We discuss the instanton/pseudo-codeword approach in this presentation. The two main themes reviewed are instanton-amoeba [13], [18] and LP-based Pseudo-Codeword Search (PCS) [8], [10] algorithms for finding low effective distance instantons/pseudo-codewords.

Instanton-amoeba, introduced in [12], [13], is an efficient numerical scheme which finds instanton/pseudo-codeword by means of a simplex (amoeba) optimization. The algorithm is initialized with a random simplex and many sequential attempts are required to built the instanton/pseudo-codeword frequency spectra of the code. The scheme is ab-initio by construction, thus it requires no additional assumptions. It is also generic, in that there are no restrictions related to the type of decoding or channel. The instanton-amoeba method is general but also computational resources consuming.

LP-based Pseudo-Codewords Search (PCS) algorithm, suggested in [8], [10], is an efficient alternative to the instanton-

amoeba. Formally, one step of the PCS algorithm constitutes sequential repetition of the LP algorithm, where the entry information, log-likelihoods, are updated according to a feedback from the previous iteration in the sequence. Like in the instanton-amoeba case, each step of the algorithm starts from a randomly selected configuration of the noise and ends at a low weight pseudo-codeword. PCS takes advantage of some special features of LP resulted in monotonicity of the procedure. We experimentally observed that effective distance of the result always decreases, or stays the same, after a single PCS circle.

The two methods, instanton-amoeba and LP-based PCS, can also be viewed as complementary ingredients of one package aiming at exploring the error-floor domain. Thus, results of the PCS algorithm can be naturally used as a starting guess for the instanton-amoeba and vice-versa.

The material in the manuscript is organized as follows. Relation between LP and BP decodings is elucidated via the unifying Bethe free energy approach in Section II. We introduce the instanton-amoeba method in Section III. LP-based PCS is discussed in Section IV. Simulation results demonstrating utility of the instanton-amoeba and the PCS methods are described in Section V. Discussion of open problems in Section VI concludes the presentation.

II. Belief Propagation and Linear Programming Decodings

We consider a generic linear code, described by its parity check $N \times M$ sparse matrix, $\hat{H}$, representing N bits and M checks. The codewords are configurations, $\boldsymbol{\sigma} = \{\sigma_i = 0,1 | i = 1,\ldots,N\}$, which satisfy all the check constraints: $\forall \alpha = 1,\ldots,M$, $\sum_i H_{\alpha i}\sigma_i = 0 \pmod 2$. A codeword sent to the channel is polluted and the task of decoding becomes to restore the most probable pre-image of the output sequence, $\boldsymbol{x} = \{x_i\}$. Probability for $\boldsymbol{\sigma}$ to be a pre-image of $\boldsymbol{x}$ is

$$\mathcal{P}(\boldsymbol{\sigma}|\boldsymbol{x}) = Z^{-1}\prod_\alpha \delta\left(\prod_{i\in\alpha}(-1)^{\sigma_i},1\right)\exp\left(-\sum_i h_i\sigma_i\right), \quad (1)$$

where one writes $i \in \alpha$ if $H_{\alpha i} = 1$; Z is the normalization coefficient (so-called partition function); the Kronecker symbol, $\delta(x,y)$, is unity if $x = y$ and it is zero otherwise; and $\boldsymbol{h}$ is the vector of log-likelihoods dependent on the output vector $\boldsymbol{y}$. In the case of the AWGN channel with the SNR ratio, $\mathrm{SNR} = E_c/N_0 = s^2$, bit transition probability is, $\sim \exp(-2s^2(x_i-\sigma_i)^2)$, and the log-likelihood becomes, $h_i = s^2(1-2x_i)$. The optimal block-MAP (Maximum Likelihood) decoding maximizes $\mathcal{P}(\boldsymbol{\sigma}|\boldsymbol{x})$ over $\boldsymbol{\sigma}$

$$\arg\max_{\boldsymbol{\sigma}} \mathcal{P}(\boldsymbol{\sigma}|\boldsymbol{x}), \quad (2)$$

and symbol-MAP operates similarly, however in terms of the marginal probability at a bit

$$\arg\max_{\sigma_i} \sum_{\boldsymbol{\sigma}\setminus\sigma_i} \mathcal{P}(\boldsymbol{\sigma}|\boldsymbol{x}). \quad (3)$$

BP and LP decodings should be considered as computationally efficient but suboptimal substitutions for MAP. Both BP and LP decodings can be conveniently derived from the so-called Bethe-Free energy approach of [3] which is briefly reviewed below. (See also [19], [9], [20].) In this approach trial probability distributions, called beliefs, are introduced both for bits and checks, b_i and b_α, respectively. The set of bit-beliefs, $b_i(\sigma_i)$, satisfy equality and inequality constraints that allow convenient reformulation in terms of a bigger set of beliefs defined on checks, $b_\alpha(\boldsymbol{\sigma}_\alpha)$, where, $\boldsymbol{\sigma}_\alpha = \{\sigma_i | i \in \alpha, \sum_i H_{\alpha i}\sigma_i = 0 \pmod 2\}$, is a local codeword associated with the check α. The equality constraints are of two types, normalization constraints (beliefs, as probabilities, should sum to one) and compatibility constraints

$$\forall i,\ \forall \alpha \ni i:\ b_i(\sigma_i) = \sum_{\sigma_\alpha\setminus\sigma_i} b_\alpha(\boldsymbol{\sigma}_\alpha),\ \sum_{\boldsymbol{\sigma}_\alpha} b_\alpha(\boldsymbol{\sigma}_\alpha) = 1. \quad (4)$$

Additionally, all the beliefs should be non-negative and smaller than or equal to unity. The Bethe Free energy is defined as a difference of the self-energy and the entropy, $F = E - S$:

$$E = \sum_i h_i \sum_{\sigma_i}\sigma_i b_i(\sigma_i), \quad (5)$$

$$S = -\sum_\alpha\sum_{\boldsymbol{\sigma}_\alpha} b_\alpha(\boldsymbol{\sigma}_\alpha)\ln b_\alpha(\boldsymbol{\sigma}_\alpha) + \sum_i\sum_{\sigma_i}(q_i-1)b_i(\sigma_i)\ln b_i(\sigma_i). \quad (6)$$

Optimal configurations of beliefs minimize the Bethe Free energy subject to the equality constraints (4). Introducing the constraints as the Lagrange multiplier terms to the effective Lagrangian and looking for the extremum with respect to all possible beliefs leads to

$$b_\alpha(\boldsymbol{\sigma}_\alpha) = \frac{\exp\left(\sum_{i\in\alpha}(h_i/q_i+\eta_{\alpha i})(1-2\sigma_i)\right)}{\sum_{\boldsymbol{\sigma}_\alpha}\exp\left(\sum_{i\in\alpha}(h_i/q_i+\eta_{\alpha i})(1-2\sigma_i)\right)}, \quad (7)$$

$$b_i(\sigma_i) = \frac{\exp\left((\eta_{\alpha i}+\eta_{i\alpha})(1-2\sigma_i)\right)}{2\cosh\left(\eta_{i\alpha}+\eta_{\alpha i}\right)}, \quad (8)$$

where the set of η fields (which are Lagrange multipliers for the compatibility constraints) satisfy

$$\eta_{\alpha i} = h_i + \sum_{\beta\ni i}^{\beta\neq\alpha}\eta_{i\alpha}, \quad \eta_{i\alpha} = \tanh^{-1}\left(\prod_{j\in\alpha}^{j\neq i}\tanh\eta_{\alpha j}\right). \quad (9)$$

These are the BP equations for LDPC codes written in its standard form. These equations are often described in the coding theory literature as stationary point equations for the BP (also called sum product) algorithm and then η variables are called messages. The BP algorithm, initialized with $\eta_{i\alpha} = 0$, solves Eqs. (9) iterating it sequentially from right to left. The iterative solution is exact on a tree, i.e. a graph without loops. This is the iterative BP algorithm discussed in the paper. Let us also notice, for the sake of completeness, that even though significance of the BP equations for MAP decoding of actual codes (with loops) was established [21], [19], [9], the standard tree-motivated choice of the BP iterations scheduling is by no means obvious. Possible lack of the iterative algorithm convergence (to respective solution of the BP equation) is a particular concern, and some relaxation methods were recently introduced to deal with the problem [22], [23].

LP is a close relative of BP which does not have this unpleasant problem with convergence. Originally, LP decoding was introduced as a relaxation of ML decoding [4]. Eq. (2) can be restated as

$$\arg\min_{\boldsymbol{\sigma}\in\mathcal{P}}\left(\sum_i h_i\sigma_i\right), \tag{10}$$

where $\mathcal{P}$ is the polytope spanned by all the codewords of the code. Looking for $\boldsymbol{\sigma}$ in terms of a linear combination of the codewords, $\boldsymbol{\sigma}_v$: $\boldsymbol{\sigma}=\sum_v \lambda_v\boldsymbol{\sigma}_v$, where $\lambda_v\geq 0$ and $\sum_v\lambda_v=1$, one observes that block-MAP turns into a linear optimization problem. LP-decoding algorithm of [4] proposes to relax the polytope, expressing $\boldsymbol{\sigma}$ in terms of a linear combination of local codewords associated with checks, $\boldsymbol{\sigma}_\alpha$. We will not give details of this original formulation of LP here, because we prefer an equivalent formulation elucidating connection to BP decoding. One finds that BP decoding, understood as an algorithm searching for a stationary point of the BP equations, turns into LP decoding in the asymptotic limit of large SNR. Indeed in this special limit the entropy terms in the Bethe free energy can be neglected and the problem turns to minimization of a linear functional under a set of linear constraints. The similarity between LP and BP (the later one identified with a minimum of the Bethe Free energy [3]) was noticed in [4] and it was also discussed in [5], [6], [7], [9]. Stated in terms of beliefs, LP decoding minimizes the self-energy part (5) of the full Bethe Free energy functional under the set of linear equality constraints (4) and also linear inequalities guaranteeing that all the beliefs are non-negative and smaller than or equal to unity. This gives us full definition of the so-called large polytope LP decoding. One can run it as is in terms of bit- and check- beliefs, however it may also be useful to re-formulate the LP procedure solely in terms of the bit beliefs. The small polytope formulation of LP is due to [26] and [4]. Indeed, self-energy is stated only in terms of bit beliefs, and moreover one rewrites it just in terms of $f_i=b_i(1)$, excluding $b_i(0)=1-f_i$ from the consideration. Furthermore, one can also exclude check beliefs, replacing them by a set of inequality constraints imposed on f_i. The later remain the only set of variables stayed in the small polytope formulation, $\forall\alpha,\forall T\subseteq\mathcal{N}(\alpha)=\{i;i\in\alpha\},\ |T|$ is odd :

$$\sum_{i\in T} f_i+\sum_{i\in(N(\alpha)\setminus T)}(1-f_i)\leq|N(\alpha)|-1. \tag{11}$$

III. Instanton-amoeba as a general method of the noise space exploration [13], [18]

Goal of decoding is to infer the original message from the received output, $\boldsymbol{x}$. Assuming that coding and decoding are fixed and aiming to characterize performance of the scheme, one studies Frame-Error-Rate (FER) FER $=\int d\boldsymbol{x}\ \chi_{\text{error}}(\boldsymbol{x})P(\boldsymbol{x}|\mathbf{0})$, where $\chi_{\text{error}}=1$ if an error is detected and $\chi_{\text{error}}=0$ otherwise. In symmetric channel FER is invariant with respect to the original codeword, thus all-0 codeword can be assumed for the input. When SNR is large FER, as an integral over output configurations, is approximated by,

$$\text{FER}\sim\sum_{\text{inst}} V_{\text{inst}}\times P\left(\boldsymbol{x}_{\text{inst}}|\mathbf{0}\right), \tag{12}$$

where $\boldsymbol{x}_{\text{inst}}$ are the special instanton configurations of the output maximizing $P(\boldsymbol{x}|\mathbf{0})$ under the $\chi_{\text{error}}=1$ condition, and V_{inst} combines combinatorial and phase-volume factors. See Fig. 1 for illustration. Generally, there are many instantons that are all local maxima of $P(\boldsymbol{x}|\mathbf{0})$ in the noise space.

For the AWGN channel (considered as the main model example) finding the instanton means minimizing $d=\boldsymbol{x}^2$ with respect to the noise vector $\boldsymbol{x}$ in the N-dimensional space and under the condition that the decoding terminates with an error. Instanton estimation for FER at the highest SNR, $s\gg 1$, is $\sim\exp(-d_{\text{inst}}\cdot s^2/2)$, while at moderate values of SNR many terms from the right-hand-side of Eq. (12) can contribute to FER comparably.

In our instanton-amoeba numerical scheme instanton with the smallest effective distance, d_{inst}, was found by a downhill simplex method also called "amoeba", with accurately tailored (for better convergence) annealing. We repeat the instanton-amoeba evaluation many times, always starting from a new set for initial simplex chosen randomly. d, as a function of noise configuration inside the area of unsuccessful decoding, has multiple minima each corresponding to an instanton. Multiple attempts of the instanton-amoeba evaluations gives us not only the instanton with the minimal d_{inst} but also the whole spectra of higher valued d_{inst}.

Instanton is a highly probable configuration of the noise leading to an error. Decoding applied to the instanton configuration results in the so-called pseudo-codeword [15], [16], [17], [11], [5]. Effective distance, d_{inst}, characterizing an instanton and its respective pseudo-codeword, should be compared with the Hamming distance of the code, d_{ML}, which measures minimal number of flips (from 0 to 1 and vice

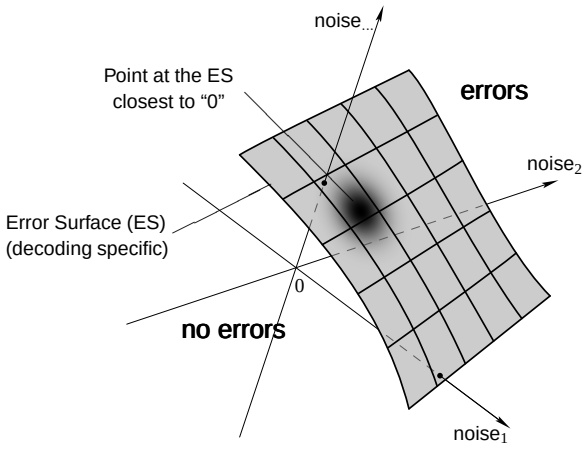

Fig. 1. Illustration for the instanton method. The noise space is divided into areas of successful and erroneous decoding by error surface. The point at the error surface closest (in the appropriate metrics) to the point of zero noise is the most probable configuration of the noise causing the decoding error. Contribution from the special configuration of the noise, the instanton, and its close vicinity estimates the noise integral for FER.

versa) required for changing from the all zero codeword to another codeword of the code. Instanton/pseudo-codeword with $d < d_{\text{ML}}$ will completely screen contribution of the respective codeword into FER at the largest SNRs. We will see below in Section V) that this situation is actually realized for one of the codes discussed.

We develop two different versions of "amoeba", "soft" and "hard". In "soft amoeba" the minimization function decreases with noise probability density in erroneous area of the noise, while in area of successful decoding the function is made artificially big (to guarantee that the actual minimum is achieved inside the erroneous domain). In the "hard amoeba" case minimization is performed only over all orientations of the noise vector, while the length of the vector corresponds exactly to respective point at the error surface, that is the surface separating domains of errors from the domain of correct decoding. (See Fig. 1 for illustration.) This special point at the error surface is found numerically by bisection method. In [13], [14] the "hard amoeba" was used. In [18] we found that even though the "hard amoeba" outperforms the "soft amoeba" for relatively short codes, the later one has clear advantage in the computational efficiency for mid-size and long codes.

Once an instanton is found, its validity can be verified against a theoretical evaluation. This theoretical approach, introduced in [13], [14], is based on the notion of the computational tree (CT) of Wiberg [16] built by unwrapping the Tanner graph of a given code into a tree from a bit for which one determines the probability of error. The concept of CT is useful because the result of iterative decodings at a bit of an LDPC code and at the tree center of the respective CT are equal by construction [16]. The initial messages at any bit of the tree are log-likelihoods and, therefore, the result obtained in the tree center is a linear combination of the log-likelihoods with integer coefficients, so the error surface condition becomes $\sum_i n_i h_i = 0$ with integer n_i that depend on CT structure. For AWGN channel the instanton length is equal to $d_{\text{inst}} = (\sum_i n_i)^2/(\sum_i n_i^2)$ [16]. The definition of n_i was generalized in [13]. This CT approach was further developed in [14], discussing an example of non-Gaussian (Laplacian) channel and, thus, explicitly demonstrating that the instantons are channel dependent. In spite of its clear utility the CT approach becomes impractical for larger number of iterations. Thus, we actually use the CT approach only to verify validity of the instanton-amoeba results for relatively small number of iterations.

IV. Accelerated Pseudo-codeword search for LP decoding [8]

Suppose a pseudo codeword, $\tilde{\boldsymbol{\sigma}} = \{\tilde{\sigma}_i = b_i(1);\ i = 1, \ldots, N\}$, corresponding to the most damaging configuration of the noise (instanton) counted from the all zero codeword, $\boldsymbol{x}_{\text{inst}}$, is found. Then finding the instanton configuration itself (i.e. respective configuration of the noise) is not a problem, one only needs to maximize the transition probability with respect to the noise field, $\boldsymbol{x}$, taken at $\boldsymbol{\sigma} = 0$ under the condition

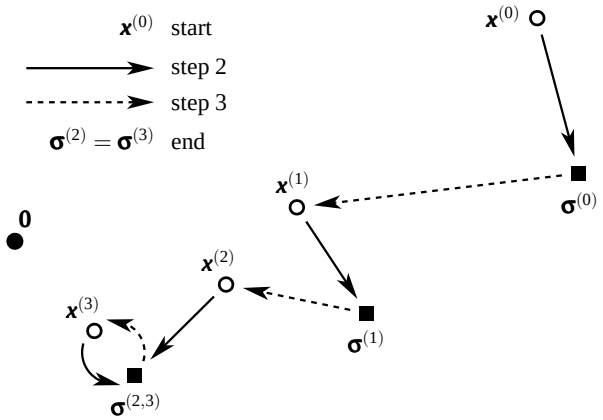

Fig. 2. Schematic illustration of the pseudo-codeword-search algorithm. This example terminates at $k_* = 3$.

that the self-energy calculated for the pseudo-codeword in the given noise field $\boldsymbol{x}$ is zero (i.e. equal to the value of the self energy for the zero code word). The resulting expression for the optimal configuration of the noise (instanton) in the case of the AWGN channel is $\boldsymbol{x}_{\text{inst}} = (\tilde{\boldsymbol{\sigma}} \sum_i \tilde{\sigma}_i)/(2\sum_i \tilde{\sigma}_i^2)$, and the respective effective distance is $d_{\text{LP}} = (\sum_i \tilde{\sigma}_i)^2/\sum_i \tilde{\sigma}_i^2$. This definition of the effective distance was first described in [17], with the first applications of this formula to LP decoding discussed in [5] and [7]. Note also that the expressions are reminiscent of the formulas derived by Wiberg and co-authors in [15] and [16], in the context of the computational tree analysis applied to iterative decoding with a finite number of iterations.

Let us now introduce the pseudo-codeword-search algorithm [8] inspired by the aforementioned median procedure.

- **Start:** Initiate a starting configuration of the noise, $\boldsymbol{x}^{(0)}$. Noise is counted from zero codeword and it should be sufficiently large to guarantee convergence of LP to a pseudo-codeword different from the zero codeword.
- **Step 1:** LP decodes $\boldsymbol{x}^{(k)}$ to a codeword $\boldsymbol{\sigma}^{(k)}$.
- **Step 2:** Find $\boldsymbol{y}^{(k)}$, the weighted median in the noise space between the pseudo codeword, $\boldsymbol{\sigma}^{(k)}$, and the zero codeword. The AWGN expression for the weighted median is $\boldsymbol{y}^{(k)} = (\boldsymbol{\sigma}^{(k)} \sum_i \sigma_i^{(k)})/(2\sum_i \left(\sigma_i^{(k)}\right)^2)$.
- **Step 3:** If $\boldsymbol{y}^{(k)} = \boldsymbol{y}^{(k-1)}$, then $k_* = k$ and the algorithm terminates. Otherwise go to Step 2, assigning $\boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)} + 0$. ($+0$ prevents decoding into the zero codeword, keeping the result of decoding within the erroneous domain.)
- **Output** configuration $\boldsymbol{y}^{(k_*)}$ is the configuration of the noise that belongs to the error-surface surrounding the zero codeword. (The error-surface separates the domain of correct LP decisions from the domain of incorrect LP decisions.) Moreover, locally, i.e. for the given part of the error-surface equidistant from the zero codeword and the pseudo codeword $\boldsymbol{\sigma}^{(k_*)}$, $\boldsymbol{y}^{(k_*)}$ is the nearest point of the error-surface to the zero codeword.

We repeat the algorithm many times picking the initial noise

configuration randomly, however guaranteeing that it would be sufficiently far from the zero codeword so that the result of the LP decoding (first step of the algorithm) is a pseudo-codeword distinct from the zero codeword. We showed in [8] that the PCS algorithm converges in a relatively small number of iterations.

For the sake of completeness, let us also notice that there are some LP-specific limitations which are carried over to the bare PCS algorithm. Thus, LP decoding operates with the local codewords while their number grows exponentially with check degrees, q_α. However, this undesirable complexity of LP can actually be dealt with. It was noticed in [4] that only relatively few of the LP constraints are actually used in decoding. Some suggestions were introduced to overcome the problem [4], [24], [25], [10]. Even though any of the complexity reduction method can probably be used to improve performance of PCS, our only tests so far were based on the dendro-LDPC method of [10]. The dendro-LDPC approach suggests to change the graphical representation of the model by replacing all checks of high degree by dendro-subgraphs (trees) with appropriate number of auxiliary checks of degree three and number of punctured, i.e. not transmitted, bits of degree two. We showed in [10] that the dendro-code and the original code have identical set of codewords and pseudo-codewords. Moreover, for any configuration of the channel output the results of MAP decodings are identical for the two codes. Another result, reported in [10], is that the described above PCS algorithm works flawlessly for the dendro-codes. The dendro version of the algorithm is actually identical to the one described above under exception of what concerns the punctured nodes. First, one should always zero the log-likelihoods at all the punctured nodes and, second, calculating the weighted medians one should exclude punctured nodes from the sum.

Our direct attempts to extend the PCS algorithm to BP decoding did not succeed. In this regards, we attribute success of the PCS in the LP case to the fact that the weighted median $(+0)$ of the zero codeword and a pseudo codeword *is not* decoded into the zero codeword, generating a new pseudo-codeword with effective distance smaller or equal to (but never larger than!) the one of the initial pseudo-codeword. This special feature of LP which allows to find a median of the pseudo-codeword and the codeword is apparently lacking in the standard iterative BP. In spite of that we still found an indirect way of using the PCS LP results for analysis of the BP decoding. One simply uses result of the LP-PCS as entry guess for the BP instanton-amoeba search. The hybrid method works well, often resulting in discovery of BP instantons/pseudo-codewords with small effective distance.

V. Pseudo-codeword spectra and FER vs SNR performance curve

In this Section, aimed to illustrate utility of the instanton-amoeba and PCS methods described above, we analyze two codes, the Tanner $[155, 64, 20]$ code introduced in [27], and the $p = 7$ Margulis code $[672, 336, 16]$ introduced in [28] and also discussed in [29]. The two codes are selected for demonstration, in part, because they show qualitatively different behavior in the error-floor domain. The results discussed in this Section were partially presented before in [18] and [8] for BP and LP decodings respectively.

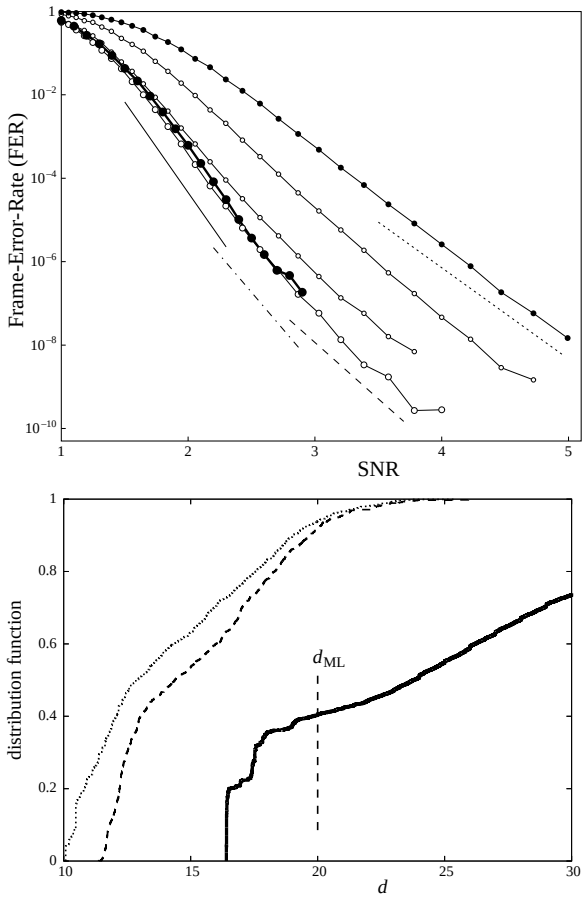

Fig. 3. FER vs SNR $(= s^2)$ curves for the $[155, 64, 20]$ code decoded by iterative BP decoding with $4, 8, 128, 1024$ iterations and LP decoding are shown on the top Figure A. Filled dots for the data points correspond to 4 iterations BP decoding. Improvement of the code performance with the number of iterations is monotonic. Filled dots over bold line correspond to LP decoding. Solid straight line corresponds to the Hamming distance asymptotics, FER $\sim \exp(-20 \cdot s^2/2)$. Dotted straight line corresponds to the minimal distance instanton for BP with 4 iterations, FER $\sim \exp(-d_4 \cdot s^2/2)$, where $d_4 = 46^2/210$. Dashed straight line correspond to a special instanton configuration with $d = 12.5$ that withstand 400 BP iterations. Dash-dotted straight line corresponds to the minimal distance instanton for LP decoding, FER $\sim \exp(-d_{LP} \cdot s^2/2)$, where $d_{LP} \approx 16.4037$.
Figure B, on the bottom, shows pseudo-codeword spectra found for iterative BP and LP decodings by the instanton-amoeba and the PCS methods respectively. Solid, dotted and dashed curve show results for the LP and BP decodings with four and eight iterations respectively.

We perform numerical simulations of three distinct and complementary types. First of all we study FER vs SNR curve for iterative BP decoding (with fixed number of iterations) and LP decoding by direct Monte-Carlo simulations. These MC results, shown in Fig. 3A and Fig 4A for the two codes respectively, provide a test ground for the two other instanton-amoeba and PCS methods aimed at exploring efficiently the error-floor domain. The outcome of the error-floor exploration

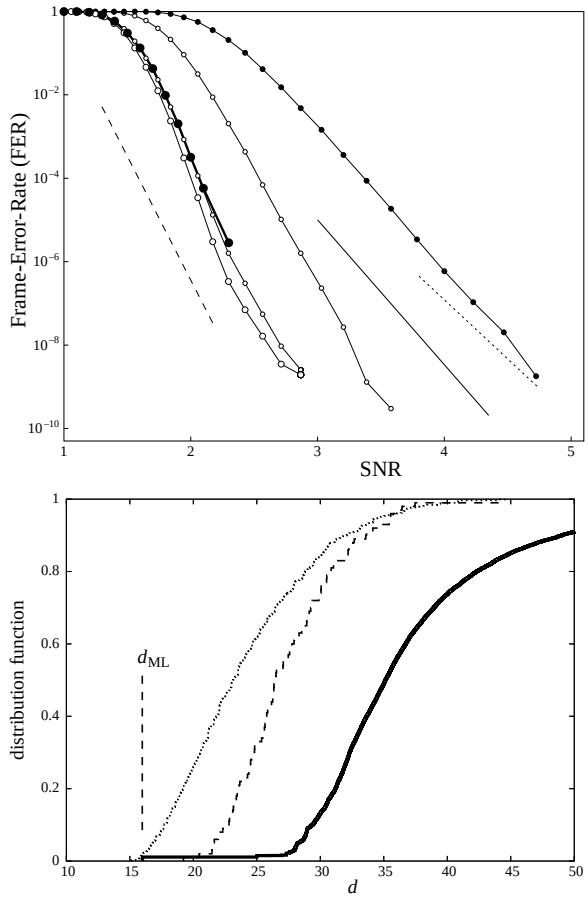

Fig. 4. FER vs SNR ($= s^2$) curves for the $[672, 336, 16]$ code decoded by iterative BP and LP decodings are shown on the top Figure A. Different curves correspond to BP decoding with 4, 8, 128 and 1024 iterations. Filled dots for the data points correspond to 4 iterations decoding. Improvement of the code performance with the number of iterations is monotonic. Filled dots over bold line correspond to LP decoding. Solid straight line corresponds to the Hamming distance asymptotics, FER $\sim \exp(-16 \cdot s^2/2)$. Dotted straight line corresponds to the minimal distance instanton for BP with 4 iterations, FER $\sim \exp(-d_{4;\text{inst}} \cdot s^2/2)$, where $d_{4;\text{inst}} = 46^2/162$. Dashed straight line corresponds to a non-codeword LP-instanton with the minimal distance, FER $\sim \exp(-27.33 \cdot s^2/2)$.
Figure B, on the bottom, shows pseudo-codeword spectra, i.e. probability to observe pseudo-codeword of a given or smaller effective distance, found for iterative BP and LP decodings by the instanton-amoeba and the PCS methods respectively. Solid, dotted and dashed curve show results for the LP and BP decodings with four and eight iterations respectively.

experiment is contained in a list of observed low-weight pseudo-codeword configurations, that is compactly presented in the form of the effective-distance spectra of the codes, shown in Fig. 3B and Fig 4B respectively.

In the case of the $[155, 64, 20]$ code the pseudo-codeword spectrum of LP starts at $d_{\text{inst}} \approx 16.404$ and the pseudo-codeword frequency increases with the effective distance passing though $d_{\text{ML}} = 20$ without any visible anomaly. The growth starting immediately at d_{inst} is fast, indicating that the frequency of the low-effective distance configurations is considerable, i.e. $O(1)$. This form of the pseudo-codeword spectra is fully consistent with what is seen in the MC simulations: the error-floor asymptotic of FER, $\sim \exp(-d_{\text{inst}} \cdot s^2/2)$, correspondent to the pseudo-codeword with the lowest effective weight, sets early. The pseudo-codeword spectra of BP are qualitatively similar to the one of LP. We anticipate that the BP spectra gets closer to the LP one with the number of iterations increased.

The behavior demonstrated by the $[672, 336, 16]$ code is different. Looking, first, at the pseudo-codeword LP spectra we find that configuration with the lowest effective distance is actually a codeword, $d_{\text{ML}} = 16$. We also find in the spectrum two other codewords correspondent to $d = 24$ and $d = 25$. Even thought the special low distance configurations were observed, their frequencies were orders of magnitude smaller than of other pseudo-codeword configurations found at $d \gtrsim 27.33$. Emergence of the gap suggests that, in spite of the fact the relatively small Hamming distance will certainly dominate the largest SNR asymptotic of FER, the moderate SNR asymptotic should actually be controlled by continuous part of the pseudo-codeword spectra above the gap. This prediction is indeed consistent with MC results shown in Fig. 4A where the early set intermediate asymptotic, with the slop steeper than the one given by the non-codeword instanton with the smallest effective distance, $\sim \exp(-27.33 \cdot s^2/2)$, changes to a shallower curve with the SNR increase. Like in the case of the $[155, 64, 20]$ code, the pseudo-codeword spectra of BP are qualitatively similar to the respective one of LP.

VI. Conclusions

We conclude with some general remarks highlighting directions for future research.

One important result of this work is that analyzing LP and BP algorithms simultaneously is helpful. The two algorithms are asymptotically equivalent at large SNR. Currently, BP is thought of as an algorithm of a greater practical value, however as this work suggests, LP is easier for analysis. Therefore, developing a more flexible, coding specific, and hopefully distributed, LP decoding may be one fruitful research direction. This future research should benefit from recently developed approaches towards reducing LP-complexity [24], [25], [10] and improving performance of LP decoding [30], [9]. On the BP side of the problem, one would, first of all, like to develop more efficient ways of the error-floor analysis. One conjecture here is that BP, understood as a fixed point of BP equations, may actually be suitable for the accelerated PCS-style analysis of the instanton spectra. In this regards, relaxing iterative BP, e.g. through the method proposed in [18], may constitute possible resolution to the PCS failure in the case of iterative BP caused by its irregular, cyclic dynamics.

It was shown recently [11], [31], [32] that codes within expurgated (properly designed) ensembles show very good convergence (practically identical FER vs SNR dependence) in the water-fall domain. However the behavior is qualitatively different in the error-floor domain where a widely spread distribution (over codes within an ensemble) is observed. This observation emphasizes importance of an individual code analysis in the error-floor domain discussed in this presentation. We anticipate that the ensemble averaged approaches,

e.g. of the type discussed in [31], [32], and the individual code instanton approach employed together could actually be very useful in designing new efficient and application specific coding schemes.

Let us conclude by noticing that instanton analysis of the BP/LP decoding can be easily extended to variety of practically important correlated channels, e.g. inter-symbol-interference channel. The approach can also be tuned to other problems in communications, storage, operational research and network science, wherever it is necessary to analyze algorithms of statistical inference in an extreme, low probability domain unaccessible to standard Monte-Carlo methods.

This work 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.

REFERENCES

[1] R.G. Gallager, *Low density parity check codes* (MIT PressCambridhe, MA, 1963).

[2] D.J.C. MacKay, *Good error-correcting codes based on very sparse matrices*, IEEE Trans. Inf. Theory **45** (2) 399-431 (1999).

[3] J.S. Yedidia, W.T. Freeman, Y. Weiss, *Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms*, IEEE IT**51**, 2282 (2005).

[4] J. Feldman, M. Wainwright, D.R. Karger, *Using linear programming to decode binary linear codes*, IEEE IT**51**, 954 (2005).

[5] R. Koetter, P.O. Vontobel, *Graph covers and iterative decoding of finite-length codes*, Proc. 3rd International Symposium on Turbo Codes & Related Topics, Brest, France, p. 75–82, Sept. 1–5, 2003.

[6] P.O. Vontobel, and R. Koetter, *On the relationship between LP decoding and Min-Sum Algorithms Decoding*, ISITA 2004, Parma Italy.

[7] P.O. Vontobel, R. Koetter, *Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes*, arXiv:cs.IT/0512078.

[8] M. Chertkov, M. Stepanov, *An Efficient Pseudo-Codeword-Search Algorithm for Linear Programming Decoding of LDPC Codes*, arXiv:cs.IT/0601113, submitted to IEEE Transactions on Information Theory.

[9] M. Chertkov, V. Chernyak, *Loop Calculus Helps to Improve Belief Propagation and Linear Programming Decodings of Low-Density-Parity-Check Codes*, invited talk at 44th Allerton Conference (September 27-29, 2006, Allerton,IL), arXiv:cs.IT/0609154.

[10] M. Chertkov, M. Stepanov, *Pseudo-codeword Landscape*, submitted for proceeding of ISIT 2007, June 2007, Nice, cs.IT/0701084.

[11] T. Richardson, *Error floors of LDPC codes*, 2003 Allerton conference Proccedings.

[12] V. Chernyak, M. Chertkov, M.G. Stepanov, B. Vasic, *Error correction on a tree: an instanton approach*, Phys. Rev. Lett. **93**, 198702 (2004).

[13] M.G. Stepanov, V. Chernyak, M. Chertkov, B. Vasic, *Diagnosis of weakness in error correction codes: a physics approach to error floor analysis*, Phys. Rev. Lett. **95**, 228701 (2005) [See also http://www.arxiv.org/cond-mat/0506037 for extended version with Supplements.]

[14] M. Stepanov and M. Chertkov, *The error-floor of LDPC codes in the Laplacian channel*, Proceedings of 43rd Allerton Conference (September 28-30, 2005, Allerton, IL), arXiv:cs.IT/0507031.

[15] N. Wiberg, H-A. Loeliger, R. Kotter, *Codes and iterative decoding on general graphs*, Europ. Transaction Telecommunications **6**, 513 (1995).

[16] N. Wiberg *Codes and decoding on general graphs*, Ph.D. thesis, Linköping University, 1996.

[17] G.D. Forney, Jr., R. Koetter, F.R. Kschischang, and A. Reznik, *On the effective weights of pseudocodewords for codes defined on graphs with cycles*, in Codes, Systems, and Graphical Models (Minneapolis, MN, 1999)(B. Marcus and J. Rosenthal, eds.) vol. 1233 of IMA Vol. Math. Appl.., pp. 101-112 Springer Verlag, New York, Inc., 2001.

[18] M.G. Stepanov, M. Chertkov, *Instanton analysis of Low-Density-Parity-Check codes in the error-floor regime*, arXiv:cs.IT/0601070, Proceeding of ISIT 2006, July 2006 Seattle.

[19] M. Chertkov, V. Chernyak, *Loop series for discrete statistical models on graphs*, J. Stat. Mech. (2006) P06009, cond-mat/0603189.

[20] V. Chernyak, M. Chertkov, *Loop Calculus and Belief Propagation for q-ary Alphabet: Loop Tower*, submitted for proceeding of ISIT 2007, June 2007, Nice, cs.IT/0701086.

[21] M. Chertkov, V. Chernyak, Loop Calculus in Statistical Physics and Information Science, Phys. Rev. E **73**, 065102(R) (2006); cond-mat/0601487.

[22] M. Stepanov, M. Chertkov, *Improving convergence of belief propagation decoding*, cs.IT/0607112, Proceedings of 44th Allerton Conference (September 27-29, 2006, Allerton, IL).

[23] S. Laendner, T. Hehn, O. Milenkovic, J. B. Hubercs, *Two Methods for Reducing the Error-Floor of LDPC Codes*, it/0701006.

[24] M.H. Taghavi N., P.H. Siegel, *Adaptive Linear Programming Decoding*, Proceedings of the IEEE ISIT, Seattle 2006, arxiv:cs.IT/0601099.

[25] P.O. Vontobel and R. Koetter, *Towards Low-Complexity Linear-Programming Decoding*, Proc. 4th Int. Symposium on Turbo Codes and Related Topics, Munich, Germany, April 3-7, 2006, arxiv:cs/0602088.

[26] M. Yannakakis, *Expressing combinatorial optimization problems by linear programs*, J. of Computer and System Sciences, **43**(3) 441-466 (1991).

[27] R.M. Tanner, D. Srkdhara, T. Fuja, *A class of group-structured LDPC codes*, Proc. of ISCTA 2001, Ambleside, England.

[28] G.A. Margulis, *Explicit construction of graphs without short circles and low-density codes*, Combinatorica **2**, 71 (1982).

[29] D.J.C. MacKay and M.J. Postol, *Weaknesses of Margulis and Ramanujan-Margulis Low-Density Parity-Check codes*, Proceedings of MFCSIT2002, Galway, http://www.inference.phy.cam.ac.uk/mackay/ abstracts/margulis.html .

[30] A.G. Dimakis, M.J. Wainwright, *Guessing Facets: Polytope Structure and Improved LP Decoder*, Proceedings of the IEEE ISIT, Seattle 2006, arxiv:cs/0608029.

[31] A. Amraoui, A. Montanari and R. Urbanke, *How to find Good Finite-Length Codes: From art Towards Science*, in Proc. of 4-th International Symposium on Turbo codes and related topics, Munich 3-7 Apr 2006; cs.IT/0607064.

[32] A. Amraoui, *Asymptotic and finite length optimization of LDPC codes*, Ph.D. Thesis, Lausanne 2006.