Looping linear programming decoding of LDPC codes

Michael Chertkov and Mikhail G. Stepanov

***Abstract*— Considering the Linear Programming (LP) decoding of Low-Density-Parity-Check (LDPC) code [1], we describe an efficient algorithm for finding pseudo codewords with the smallest effective distance. The algorithm, coined LP-loop, starts from choosing randomly initial configuration of the noise. The configuration is modified through a discrete number of steps. Each step consists of two sub-steps. First, one applies LP decoder to the initial noise-configuration deriving a pseudo-codeword. Second, one finds configuration of the noise equidistant from the pseudo codeword and the all zeros codeword. The resulting noise configuration is used as an entry for the next step of the LP-loop algorithm. The iterations converge fast to a pseudo codeword neighboring the all zeros codeword — domain of the pseudo codeword attraction shares a common piece of error-surface with the all zeros codeword domain. Repeated many times the LP-loop procedure is characterized by the distribution function (frequency spectrum) of the pseudo codeword effective distance. Effective distance of the coding scheme is approximated by the shortest distance pseudo-codeword in the spectrum. Efficiency of the LP-loop algorithm is demonstrated on examples of the Tanner $(155, 64, 20)$ code and Margulis $p = 7$ and $p = 11$ codes (672 and 2640 bits long respectively) performing over Additive-White-Gaussian-Noise (AWGN) channel.**

***Index Terms*— LDPC codes, Linear Programming Decoding, Error-floor**

I. Introduction I: LDPC codes and their decodings.

We consider generic LDPC code of Gallager [2], described by its parity check $N \times M$ sparse matrix, $\hat{H}$, representing N bits and M checks. A codeword $\boldsymbol{\sigma} = \{\sigma_i\}$, $i = 1, \ldots, N$ and $\sigma_i = 0, 1$ satisfies all the check constraints: $\forall \alpha = 1, \ldots, M$, $\sum_i H_{\alpha i}\sigma_i = 0 \pmod 2$. We discuss the practical case of finite N and M, as opposed to the $N, M \to \infty$ (thermodynamic) limit for which Shannon capacity theorems were formulated [3]. The codeword is sent into a noisy channel. To make our consideration concrete we consider specific model for the channel — AWGN channel. (Notice that all the discussions and results of the paper can be easily generalized to other additive noise linear channel models.) Corruption of a codeword in the AWGN channel is described by the following transition probability:

$$\mathcal{P}(\boldsymbol{x}|\boldsymbol{\sigma}) \propto \prod_i \exp\left[-2s^2(x_i - \sigma_i)^2\right], \tag{1}$$

where $\boldsymbol{x}$ is the signal measured at the channel output and s is the Signal-to-Noise Ratio (SNR) of the code. Ideal, Maximum Likelihood (ML), decoding correspondent to restoration of the most probable pre-image $\boldsymbol{\sigma}'$ given the output signal $\boldsymbol{x}$,

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

is not feasible in reality since the complexity grows exponentially with the system size. Belief-propagation (BP), or sum-product, algorithm of Gallager [2] (see also [4], [6], [5]) is a popular iterative scheme often used for decoding of the LDPC codes. Another popular iterative algorithm, that can be viewed as a certain limit of the sum-product, is the min-sum algorithm. For an idealized code containing no loops (path connecting any two bits through sequence of other bits and their neighboring checks is unique) sum-product (with sufficient number of iterations) is exactly equivalent to the so-called Maximum-A-Posteriori (MAP) decoding. MAP is reduced to ML in the limit of infinite SNR. For any realistic code (with loops) both sum-product and min-sum are approximate. Sum-product can also be considered as an algorithm solving iteratively certain nonlinear equations, one refers to as the BP equations. The BP equations minimize the so-called Bethe free energy [7]. (The Bethe free energy approach originates from variational methodology developed in statistical physics [8], [9].) Minimizing the Bethe free energy, that is a nonlinear function of the probabilities/beliefs, under the set of linear (compatibility and normalizability) constraints is generally a difficult task. However in the limit of large SNR one approximates the Bethe free energy just by the self-energy part assuming that the entropy terms are inessential. Then the problem turns to minimizing a linear function under the set of linear constraints – solvable by standard and computationally feasible Linear Programming (LP) approach. This is exactly the idea behind LP decoding introduced by J. Feldman, M. Wainwright and D.R. Karger [1] in a bit different but absolutely equivalent way – as a relaxation of the ML decoding. (The authors of [1] do mention similarity of their approach to the Bethe free energy approach of [7].) In the LP approach one minimizes the Bethe self-energy,

$$E = \sum_\alpha \sum_{\sigma_\alpha} b_\alpha(\sigma_\alpha) \sum_{i\in\alpha} \sigma_i(1 - 2x_i)/k_i, \tag{3}$$

with respect to beliefs $b_\alpha(\sigma_\alpha)$ and under certain equality and inequality constraints. Here in Eq. (3) k_i is the degree of connectivity of the i bit; σ_α is a local codeword, $\sigma_\alpha = \{\sigma_i | i \in \alpha, \sum_i H_{\alpha i}\sigma_i = 0 \pmod 2\}$, associated with the check α. The equality constraints are of the two types, normalization constraints (beliefs, as probabilities, should sum to one) and

M. Chertkov [chertkov@lanl.gov] is with Theoretical Division and Center for Nonlinear Studies, LANL, Los Alamos, NM 87545, USA.

M.G. Stepanov [stepanov@cnls.lanl.gov] is with Theoretical Division and Center for Nonlinear Studies, LANL, Los Alamos, NM 87545, USA; on leave from Institute of Automation and Electrometry, Novosibirsk 630090, Russia.

compatibility constraints

$$\forall\ \alpha: \quad \sum_{\sigma_\alpha} b_\alpha(\sigma_\alpha) = 1, \tag{4}$$

$$\forall\ i\ \forall\ \alpha \ni i: \quad b_i(\sigma_i) = \sum_{\sigma_\alpha \setminus \sigma_i} b_\alpha(\sigma_\alpha), \tag{5}$$

respectively where $b_i(\sigma_i)$ is the belief (probability) to find bit i in the state σ_i. Besides, all the beliefs should be non-negative and smaller or equal than unity, thus here is the additional set of the obvious inequality constraints:

$$0 \le b_i(\sigma_i), b_\alpha(\sigma_\alpha) \le 1. \tag{6}$$

II. Introduction II: Pseudo codewords, Frame Error Rate and effective distance

As it was shown in [1], and also discussed in [10], [11], the result of the LP decoding is rarely a codeword but typically a pseudo codeword: a special configuration containing non-integers (but rational numbers) among the beliefs b_i and b_α. This configurations can be interpreted as mixed state configurations, i.e. the ones consisting of a probabilistic mixture of local (corresponding to a single check) codewords.

Important characteristics of the code/decoding performance is Frame Error Rate (FER), calculating the probability of decoding failure. FER decreases with SNR increase and the form of this dependence of FER on SNR gives the ultimate characterization of the coding scheme performance. Any decoding to a pseudo-codeword is a failure. Decoding to a codeword can also be a failure, but this would as well counts as a failure under the ML decoding. At large SNR splitting of the two FER vs SNR curves, representing ML decoding and approximate decoding (say LP decoding) is due to the pseudo codewords. Actual asymptotics of the two curves for the AWGN channel are $\mathrm{FER}_{\mathrm{ML}} \sim \exp[-d_{\mathrm{ML}} \cdot s^2/2]$ and $\mathrm{FER}_{\mathrm{LP}} \sim \exp[-d_{\mathrm{LP}} \cdot s^2/2]$, where d_{ML} is the so-called Hamming distance of the code and the d_{LP} is the effective distance of the code, specific for the LP decoding. The LP asymptotic is normally shallower than the ML one, $d_{\mathrm{LP}} < d_{\mathrm{ML}}$. This phenomenon is called error-floor [12].

For a generic linear code performed over symmetric channel it is easy to show that FER is invariant under the change of the original codeword (sent into the channel). Therefore, for the purpose of FER calculation it is enough to analyze statistics exclusively for the case of one known original codeword, say the all zeros codeword. Than calculating the effective distance of a code one makes an assumption that there exists a special configuration (or may be few special configurations) of the noise, instantons according to the terminology of [13], describing the large SNR error-floor asymptotic for FER. 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), $\boldsymbol{x}_{\mathrm{inst}}$, is found. Than finding the instanton configuration itself (i.e. respective configuration of the noise) is not a problem, one only needs to maximize the transition probability (1) with respect to the noise field, $\boldsymbol{x}$, taken at $\boldsymbol{\sigma} = 0$ and under condition 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 all zeros code word). The resulting expression for the optimal configuration of the noise (instanton) is

$$\boldsymbol{x}_{\mathrm{inst}} = \frac{\tilde{\boldsymbol{\sigma}}}{2} \frac{\sum_i \tilde{\sigma}_i}{\sum_i \tilde{\sigma}_i^2}, \tag{7}$$

and the respective effective distance is

$$d_{\mathrm{LP}} = \frac{\left(\sum_i \tilde{\sigma}_i\right)^2}{\sum_i \tilde{\sigma}_i^2}. \tag{8}$$

Eqs. (7,8) are reminiscent of the formulas derived by Wiberg and co-authors [14], [15] in the context of the computational tree analysis applied to iterative decoding with finite number of iterations.

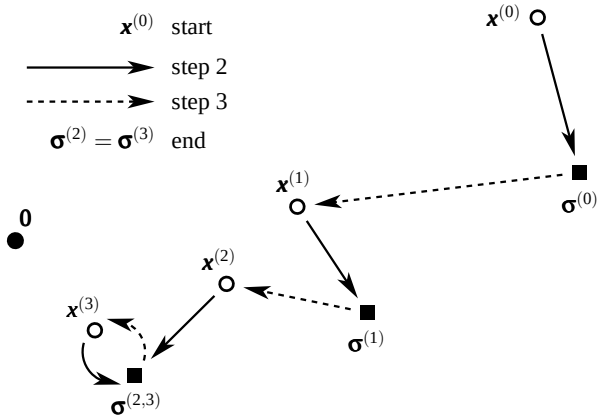

Fig. 1. Schematic illustration of the LP-loop algorithm. This example terminates at $k_* = 3$.

III. LP-loop. The algorithm.

In this Section we turn directly to describing the LP-loop algorithm. Once the algorithm is formulated, relation to the introductory material, as well as justification and motivation will become clear.

- **Start:** Initiate a starting configuration of the noise, $\boldsymbol{x}^{(0)}$.
- **Step 1:** LP decoder calculates closest pseudo codeword, $\boldsymbol{\sigma}^{(k)}$, for the given configuration of the noise

$$\{b_i^{(\mathrm{LP},k)}(\sigma_i), b_\alpha^{(\mathrm{LP},k)}(\sigma_\alpha)\} = \arg\min_{\{b_i(\sigma_i), b_\alpha(\sigma_\alpha)\}} \left\{ E\left(\boldsymbol{x}^{(k)}; \{b_i(\sigma_i), b_\alpha(\sigma_\alpha)\}\right) \right.$$
$$\left. \text{at the conditions of Eqs. (4,5,6)} \right\},$$
$$\sigma_i^{(k)} = b_i^{(\mathrm{LP},k)}(1),$$

  where the self-energy is defined according to Eq. (3).
- **Step 2:** Find the conditioned median, $\boldsymbol{y}^{(k)}$, in the noise space between the pseudo codeword, $\boldsymbol{\sigma}^{(k)}$, and the all zeros codeword

$$\boldsymbol{y}^{(k)} = \frac{\boldsymbol{\sigma}^{(k)}}{2} \frac{\sum_i \sigma_i^{(k)}}{\sum_i \left(\sigma_i^{(k)}\right)^2}.$$

- **Step 3:** If $\boldsymbol{y}^{(k)} = \boldsymbol{y}^{(k-1)}$, $k_* = k$ and the algorithm terminates, otherwise go to **Step 2** assigning $\boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)} + 0$.

- **Output:** The output configuration $\boldsymbol{y}^{(k_*)}$ is configuration of the noise that belongs to the error-surface surrounding the all zeros codeword. (The error-surface separates the domain of right LP decisions from the domain of wrong LP decisions for the original message being the all zeros codeword.) Moreover, locally, i.e. for the given part of the error-surface equidistant from the all zeros codeword and the pseudo codeword $\boldsymbol{\sigma}^{(k_*)}$, $\boldsymbol{y}^{(k_*)}$ is the closest point of the error-surface to the all zeros codeword.

The LP-loop algorithm is schematically illustrated at Fig. 1. We repeat the algorithm many times picking the initial noise configuration randomly, however guaranteing that it would be sufficiently far from the all zeros codeword so that the result of the LP decoding (first step of the algorithm) is a pseudo codeword and not the all zeros codeword. The LP-loop always converges in some relatively small number of iterations. The effective distance of the coding scheme (for given LDPC code decoded by LP decoder) is approximated by

$$d_{\rm LP} \approx \min_{\text{attempts of the LP-loop}} \left\{ \frac{\left(\sum_i \sigma_i^{(k_*)}\right)^2}{\sum_i \left(\sigma_i^{(k_*)}\right)^2} \right\}. \qquad (9)$$

It is not guaranteed that the noise configuration with the lowest possible (of all the pseudo codewords within the decoding scheme) distance is found in the result of finite number of the LP-loop iterations. However the rhs of Eq. (9) gives a very tight (if the number of attempts is sufficient) upper bound for the actual effective distance of the coding scheme.

IV. LP-loop. Examples.

In this Section we demonstrate the power of the simple LP-loop procedure explained in the previous Section by considering three popular examples of relatively long regular LDPC codes.

A. Tanner $(155, 64, 20)$ code of [16]

For this code $N = 155$ and $M = 93$. The Hamming distance of the code is known to be $d_{\rm ML} = 20$. The authors of [10] reported a pseudo codeword with $d = 16.406$. The lowest effective distance configuration found in the result of the LP-loop procedure has $d_{\rm LP} \approx 16.4037$. These two and some number of other lower lying (in the sense of their effective distance) configurations are shown in Fig. 2. The resulting frequency spectra (derived from $3,000$ attempts of the LP-loop) is shown in Fig. 3. Some of the pseudo codewords found are actually other (than the all zero one) codewords. In particular, one finds a codeword closest to the all zeros one with $d = d_{\rm ML} = 20$.

B. Margulis code [17] with $p = 7$

This code has $N = 2 \cdot M = 672$. The set of four noise configurations with the lowest effective distance found by the LP-loop algorithm for the code are shown in Fig. 4. The lowest configuration decodes into a codeword with the Hamming distance 16. A big gap separates this configuration from the next lowest configuration corresponding to a pseudo codeword

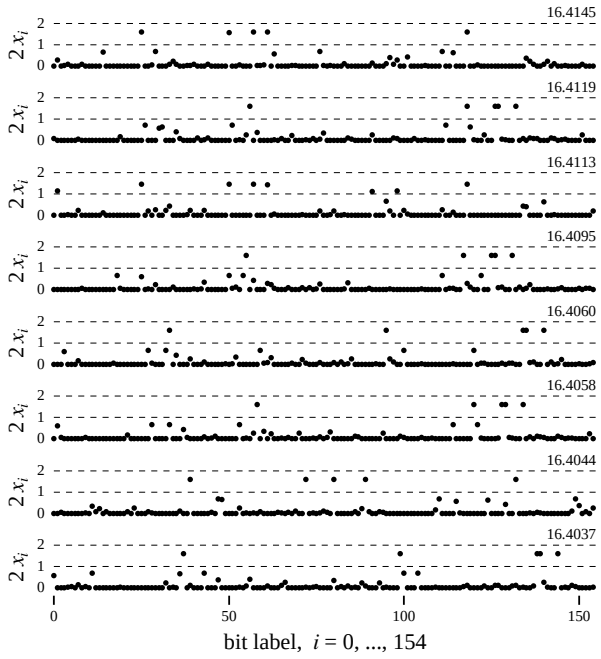

Fig. 2. 8 lowest configurations found by the LP-loop algorithm for the $(155, 64, 20)$ code. Typical number of the LP-loop iterations required to rich a stopping point is $5 \div 15$.

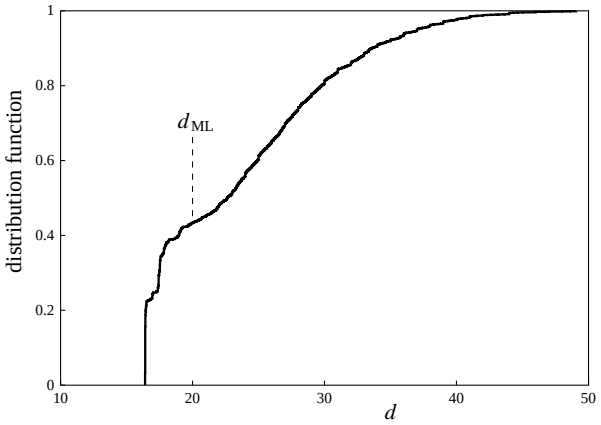

Fig. 3. Frequency spectrum (distribution function) of the effective distance constructed from 3,000 attempts of the LP-loop algorithm for the $(155, 64, 20)$ code.

that is not a codeword. Frequency spectra, characterizing performance of the LP-loop algorithm for the code, is shown in Fig. 5.

C. Margulis code [17] with $p = 11$

This code is $N = 2{\cdot}M = 2640$ bits long. We have relatively small number of configurations (30) here as it takes much longer to execute LP loop algorithm in this case. Some $30 \div 60$ steps of the LP-loop are required for a typical realization of the algorithm to rich a stopping point. Four lowest configurations are shown in Fig. 6. Obviously, with this limited statistics one cannot claim that the noise configuration with the lowest possible effective length is found. All the stopping point configurations found here correspond to pseudo codewords.

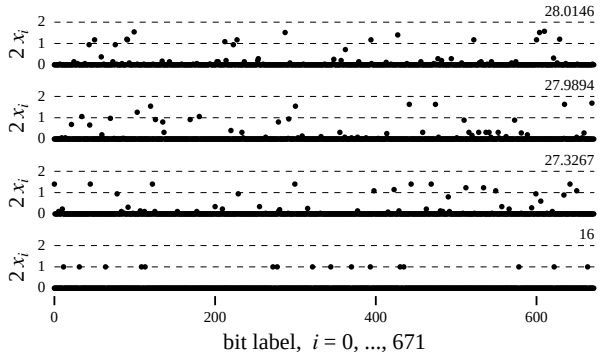

Fig. 4. 4 lowest noise configuration found by the LP-loop algorithm for the Margulis $p = 7$ code of [17]. Typical number of the LP-loop iterations required to rich a stopping point is $10 \div 20$.

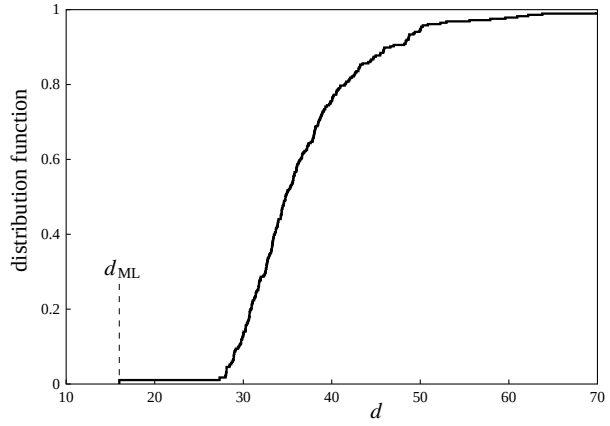

Fig. 5. Frequency spectrum (distribution function) of the effective distance found through multiple attempts of the LP-loop for the Margulis $p = 7$ code. The figure is built on 250 random attempts of the LP-loop algorithm.

(Hamming distance for the code is not known, while the upper bound mentioned in [18] is 220.)

V. Conclusions and Discussions

Let us discuss utility of the LP-loop algorithm suggested in the manuscript. The LP-loop gives an efficient way of calculating the effective distance of a code decoded by LP. It also predicts the noise configuration on the error-surface surrounding the all zeros codeword correspondent to the shortest effective distance. Efficiency of the algorithm, tested for three popular and relatively long codes, is due to fast convergence of an individual attempt of the LP-loop. (Even for the 2640 bits long code it typically takes only $30 \div 60$ steps of the LP-loop to converge.) As it was already mentioned, the LP-loop procedure applies to any additive noise linear channel. The only obvious modifications one needs to make to extend the LP-loop to other channels concern Eqs. (7,8,9) and also the basic equation of the Step 2.

One would obviously be interested to extend the looping algorithm to other (traditional) types of LDPC decoding, e.g. to finding minimal distance of sum-product and min-sum decodings. We observed, however, that at least a naive extension of the looping procedure does not work. It is guaranteed in the LP decoding case that the noise configuration found as a

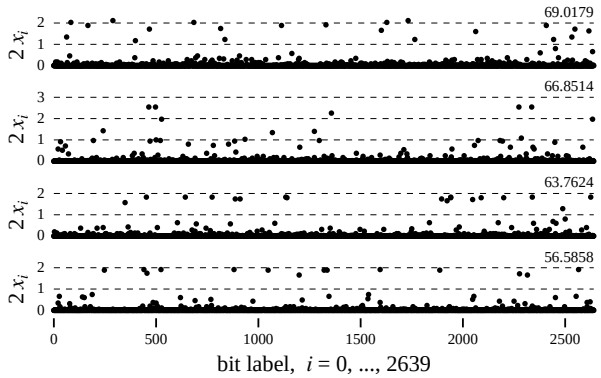

Fig. 6. 4 lowest noise configurations found by the LP-loop algorithm for the Margulis $p = 11$ code of [17]. Typical number of the LP-loop iterations required to rich a stopping point is $30 \div 60$.

median ($+0$) of the all zeros codeword and a pseudo codeword *will not* be decoded into the all zeros codeword. This allows us to explore the noise space always decreasing monotonically the effective distance, or keeping it constant, with any step of the LP-loop. It is not yet clear if this key feature of the LP decoding is expandable (hopefully with some modification of the median procedure) to other decodings. The question requires further investigation.

Even thought this direct attempt to extend the looping algorithm to iterative decoding did not work, we did find a way to apply the LP-loop for analysis of an iterative decoding. We used the most damaging configuration of the noise found within the LP-loop as an entry point for the instanton-amoeba method of [13], which is designed for finding instanton configurations (most damaging configurations of the noise) in the case of a standard iterative decoding. This hybrid method works well, sometimes resulting in discovery of pseudo codewords (of respective iterative scheme) with impressively small effective distance. We attribute this fact to the close relation existed between the LP decoding and standard iterative decodings. The results of this hybrid LP-loop-instanton-amoeba analysis and also detailed evaluation of the relation between LP and iterative decodings will be discussed elsewhere [19]. Summarizing, the LP-loop algorithm, complemented and extended by the instanton-amoeba method of [13], provides an efficient practical tool for analysis of the effective distance and the most damaging configuration of the noise (instanton) describing the error-floor for an arbitrary LDPC code performing over linear channel and decoded by LP or iteratively.

Continuing this discussion and turning to the Generalized Belief Propagation (GBP) of [7] as yet another type of decoding, one can also consider a Generalized Linear Programming (GLP) decoding simply combining checks in super-checks (called valid regions in [7]) and introducing respective set of extra constraints into the LP minimization procedure. GLP will obviously be an improvement against LP showing up in the effective distance increase. On the other hand, GLP will also inherit the convergence of the LP, which is important for success of the looping procedure. Therefore, to find the most damaging configuration of the noise for the GLP decoding we

suggest using the GLP-loop procedure, that is similar to the LP one and only requires to change from LP to GLP at an individual iteration step of the LP-loop algorithm.

Regarding possible application of the LP-loop algorithm to other areas of information science and statistical physics, let us note that the whole approach obviously applies to analysis of the high SNR limit in many other important inference problems. One particularly interesting example laying outside of the coding theory, where BP, GBP and thus LP, GLP, LP-loop and GLP-loop may find very interesting applications, is from the $2d$ inter-symbol interference (detection in $2d$ channels with memory), where GBP is claimed to be remarkably efficient [20].

VI. Acknowledgements

The authors acknowledge very useful, inspiring and fruitful discussions with Vladimir Chernyak, Ralf Koetter, Olgica Milenkovich and Bane Vasic. This work was supported by DOE under LDRD ER program at LANL.

References

[1] J. Feldman, M. Wainwright, D.R. Karger, *Using linear programming to decode linear codes*, 2003 Conference on Information Sciences and Systems, The John Hopkins University, March 12-14, 2003.

[2] R.G. Gallager, *Low density parity check codes* (MIT Press, Cambridge, MA, 1963).

[3] C.E. Shannon, Bell. Syst. Tech. J. **27**, 379 (1948); **27**, 623 (1948).

[4] R.G. Gallager, *Information theory and reliable communication* (Wiley, New York, 1968).

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

[6] J. Pearl, *Probabilistic reasoning in intelligent systems: network of plausible inference* (Kaufmann, San Francisco, 1988).

[7] J.S. Yedidia, W.T. Freeman, Y. Weiss, *Constructing free energy approximations and generalized belief propagation algorithms*, IEEE Trans. Inf. Theory **51**, 2282 (2005).

[8] H.A. Bethe, *Statistical theories of superlattices*, Proc. Roy. Soc. London A **150**, 552 (1935).

[9] R. Kikuchi, *A theory of cooperative phenomena*, Phys. Rev. **81**, 988 (1951).

[10] 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.

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

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

[13] M.G. Stepanov, V. Chernyak, M. Chertkov, B. Vasic, *Diagnosis of weakness in error correction: 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] N. Wiberg, H-A. Loeliger, R. Kotter, *Codes and iterative decoding on general graphs*, Europ. Transaction Telecommunications **6**, 513 (1995).

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

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

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

[18] 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 .

[19] M.G. Stepanov, M. Chertkov, *Dynamics of iterative decoding*, in preparation.

[20] O. Shental, A. J. Weiss, N. Shental and Y. Weiss, *Generalized belief propagation receiver for near-optimal detection of two-dimensional channels with memory*, IEEE Workshop on Information Theory, San Antonio, 2004.