An Efficient Pseudocodeword Search Algorithm for Linear Programming Decoding of LDPC Codes

Michael Chertkov, *Member, IEEE*, and Mikhail G. Stepanov

***Abstract*—In linear programming (LP) decoding of a low-density parity-check (LDPC) code one minimizes a linear functional, with coefficients related to log-likelihood ratios, over a relaxation of the polytope spanned by the codewords. In order to quantify LP decoding it is important to study vertexes of the relaxed polytope, so-called pseudocodewords. We propose a technique to heuristically create a list of pseudocodewords close to the zero codeword and their distances. Our pseudocodeword-search algorithm starts by randomly choosing configuration of the noise. The configuration is modified through a discrete number of steps. Each step consists of two substeps: one applies an LP decoder to the noise-configuration deriving a pseudocodeword, and then finds configuration of the noise equidistant from the pseudocodeword and the zero codeword. The resulting noise configuration is used as an entry for the next step. The iterations converge rapidly to a pseudocodeword neighboring the zero codeword. Repeated many times, this procedure is characterized by the distribution function of the pseudocodeword effective distance. The efficiency of the procedure is demonstrated on examples of the Tanner code and Margulis codes operating over an additive white Gaussian noise (AWGN) channel.**

***Index Terms*—Error-floor, linear programming decoding, low-density parity-check (LDPC) codes.**

I. Introduction LDPC Codes, Belief Propagation, and Linear Programming

We consider an LDPC code (cf. Gallager [2]) defined by some sparse parity-check matrix, $\hat{H} = \{H_{\alpha i}; \alpha = 1,\ldots,M; i = 1,\ldots,N\}$, of size $M \times N$. A codeword $\boldsymbol{\sigma} = \{\sigma_i = 0,1; i = 1,\cdots,N\}$ 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 over a noisy channel. To make our consideration concrete, we consider the additive white Gaussian noise (AWGN) channel. (Notice that all the discussions and results of this paper can be easily generalized to other linear channel models.) Corruption of a codeword in the AWGN channel is described by the following transition probability:

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

where $\boldsymbol{x}$ is the signal measured at the channel output and $2s^2$ is the signal-to-noise ratio (SNR) of the code, that is traditionally denoted as E_c/N_0. The maximum-likelihood (ML) decoding corresponding to the 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 its complexity grows exponentially with the system size.

LP decoding was introduced by Feldman, Wainwright, and Karger [1] as a computationally efficient approximation to the ML decoding. Following [1], let us first notice that (2) can be restated for the AWGN channel as calculating

$$\arg\min_{\boldsymbol{\sigma}'\in P} \left(\sum_i (1-2x_i)\sigma_i'\right) \tag{3}$$

where P is the polytope spanned by the codewords. Looking for $\boldsymbol{\sigma}'$ in terms of a linear combination of all possible codewords of the code, $\boldsymbol{\sigma}_v$: $\boldsymbol{\sigma}' = \sum_v \lambda_v \boldsymbol{\sigma}_v$, where $\lambda_v \geq 0$ and $\sum_v \lambda_v = 1$, one finds that ML turns into a linear optimization problem. LP decoding proposes to relax the polytope, expressing $\boldsymbol{\sigma}'$ in terms of a linear combination of the so-called local codewords, i.e., codewords of trivial codes, each associated with just one check of the original code and all the variable nodes connected to it. We will come to the formal definition of the LP decoding [1]–[9] later after discussing the belief propagation (BP) decoding of Gallager [2]–[6].

The BP, or sum–product, algorithm of Gallager [2] (see also [4]–[6]) is a popular iterative scheme often used for decoding of the LDPC codes. For an idealized code containing no loops (i.e., there is a unique path connecting any two bits through a sequence of other bits and their neighboring checks), the sum-product algorithm (with sufficient number of iterations) is exactly equivalent to the so-called maximum *a posteriori* probability (MAP) decoding, which is reduced to ML in the asymptotic limit of infinite SNR. For any realistic code (with loops), the sum-product algorithm is approximate, and it should actually be considered as an algorithm for solving iteratively certain nonlinear equations, called BP equations. The BP equations minimize the so-called Bethe free energy [10]. (The Bethe free

Manuscript received January 27, 2006; revised May 19, 2007. The work at LANL 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 DE-AC52-06NA25396, and, specifically, the LDRD Exploratory Research Grant on *Novel Physics Inspired Approach to Error Correction*.

M. Chertkov is with the Theoretical Division and the Center for Nonlinear Studies, Los Alamos National Lab (LANL), Los Alamos, NM 87545 USA (e-mail: chertkov@lanl.gov).

M. G. Stepanov is with the Department of Mathematics, The University of Arizona, Tucson, AZ 85721 USA and also with the Theoretical Division and the Center for Nonlinear Studies, Los Alamos National Lab (LANL), Los Alamos, NM 87545 USA (e-mail: stepanov@math.arizona.edu).

Communicated by T. J. Richardson, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2008.917682

energy approach originates from a variational methodology developed in statistical physics [11], [12].) 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.

BP decoding becomes LP decoding in the asymptotic limit of infinite SNR. Indeed in this special limit, the entropy terms in the Bethe free energy can be neglected and the problem becomes minimization of a linear function under a set of linear constraints. The similarity between LP and BP (the latter one being understood as minimizing the Bethe Free energy [10]) was first noticed in [1] and it was also discussed in [7]–[9]. Stated in terms of beliefs, i.e., trial marginal probabilities, LP decoding minimizes the Bethe self-energy

$$E = \sum_{\alpha} \sum_{\sigma_\alpha} b_\alpha(\sigma_\alpha) \sum_{i \in \alpha} \sigma_i (1 - 2x_i)/k_i \tag{4}$$

with respect to beliefs $b_\alpha(\sigma_\alpha)$ and under certain equality and inequality constraints. Here in (4) k_i is the degree of connectivity of the i-th bit; σ_α is a local codeword, $\sigma_\alpha = \{\sigma_i | i \in \alpha, \sum_i H_{\alpha i}\sigma_i = 0 \,(\mathrm{mod}\, 2)\}$, associated with the check α. The equality constraints are of two types, normalization constraints (beliefs, as probabilities, should sum to one) and compatibility constraints

$$\forall \alpha : \sum_{\sigma_\alpha} b_\alpha(\sigma_\alpha) = 1 \tag{5}$$

$$\forall i \forall \alpha \ni i : b_i(\sigma_i) = \sum_{\sigma_\alpha \setminus \sigma_i} b_\alpha(\sigma_\alpha) \tag{6}$$

respectively where $b_i(\sigma_i)$ is the belief (trial marginal probability) to find bit i in the state σ_i, and the check belief, $b_\alpha(\boldsymbol{\sigma}_\alpha)$, stands for the trial marginal probability of finding bits, which are neighbors of the check α, in the state $\boldsymbol{\sigma}_\alpha$. Also, all the beliefs, as probabilities, should be nonnegative and smaller than or equal to unity. Thus, there is the additional set of the inequality constraints

$$0 \leq b_i(\sigma_i), b_\alpha(\sigma_\alpha) \leq 1. \tag{7}$$

II. Introduction II Pseudo Codewords, Frame Error Rate and Effective Distance

As it was shown in [1] the LP decoding has ML certificate, i.e., if the pseudocodeword obtained by the LP decoder has only integral entries then it must be a codeword, in fact it is the codeword given back by ML decoder. If LP decoding does not decode to a correct codeword then it usually yields a noncodeword pseudocodeword with some number of nonintegers among the beliefs b_i and b_α. These configurations can be interpreted as mixed state configurations consisting of a probabilistic mixture of local codewords.

An important characteristic of the decoding performance is frame error rate (FER) calculating the probability of decoding failure. FER decreases as SNR increases. The form of this dependence gives an ultimate description of the coding performance. Any decoding to a noncodeword pseudocodeword is a failure. Decoding to a codeword can also be a failure, which counts as a failure under ML decoding. For large SNR, i.e., in the so-called error-floor domain, splitting of the two (FER vs SNR) curves, representing the ML decoding and an approximate decoding (say LP decoding) is due to pseudocodewords [13]. The 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 error-floor asymptotic is normally shallower than the ML one, $d_{\mathrm{LP}} < d_{\mathrm{ML}}$. The error floor can start at relatively low values of FER, unaccessible for Monte Carlo simulations. This emphasizes importance of the pseudocodewords analysis.

For a generic linear code performed over a symmetric channel, it is easy to show that the FER is invariant under the change of the original codeword (sent into the channel). Therefore, for the purpose of FER evaluation, it is sufficient to analyze the statistic exclusively for the case of one codeword, and the choice of zero codeword is natural. Then calculating the effective distance of a code, one makes an assumption that there exists a special configuration (or maybe a few special configurations) of the noise, instantons according to the terminology of [14], describing the large SNR error-floor asymptotic for FER. Suppose a pseudocodeword $\tilde{\boldsymbol{\sigma}} = \{\tilde{\sigma}_i = b_i(1); i = 1, \ldots, N\}$ corresponding to the most damaging configuration of the noise (instanton), $\boldsymbol{x}_{\mathrm{inst}}$, is found. Then finding the instanton configuration itself (i.e., respective configuration of the noise) is equivalent to maximizing the transition probability (1) with respect to the noise field, $\boldsymbol{x}$, taken at $\boldsymbol{\sigma} = 0$ under the condition that the self-energy calculated for the pseudocodeword in the given noise field $\boldsymbol{x}$ is zero (i.e., it is equal to the value of the self energy for the zero code word). The resulting expression for the optimal configuration of the noise (instanton) is

$$\boldsymbol{x}_{\mathrm{inst}} = \frac{\tilde{\boldsymbol{\sigma}}}{2} \frac{\sum_i \tilde{\sigma}_i}{\sum_i \tilde{\sigma}_i^2} \tag{8}$$

and the respective effective distance is

$$d_{\mathrm{LP}} = \frac{(\sum_i \tilde{\sigma}_i)^2}{\sum_i \tilde{\sigma}_i^2}. \tag{9}$$

This definition of the effective distance was first described in [15], with the first applications of this formula to the LP decoding discussed in [7] and [9]. Note also that (8), (9) are reminiscent of the formulas derived by Wiberg and coauthors in [16] and [17], in the context of the computational tree analysis applied to iterative decoding with a finite number of iterations.

III. Searching for Pseudocodewords

In this section, we turn directly to describing an algorithm which allows one to find efficiently pseudocodewords of an LDPC code performing over AWGN channel and decoded by LP. Once the algorithm is formulated, its relation to the introductory material, as well as partial justification and motivation will become clear.

The Pseudocodeword Search Algorithm

- **Start:** Initiate a starting configuration of the noise $\boldsymbol{x}^{(0)}$. Noise measures a deviation from the zero codeword and it should be sufficiently large to guarantee convergence of LP to a pseudocodeword different from the zero codeword.

- **Step 1:** The LP decoder finds the closest pseudocodeword $\boldsymbol{\sigma}^{(k)}$ for the given configuration of the noise

$$\left\{b_i^{(\mathrm{LP},k)}(\sigma_i), b_\alpha^{(\mathrm{LP},k)}(\sigma_\alpha)\right\} = \arg \min_{\{b_i(\sigma_i), b_\alpha(\sigma_\alpha)\}} \left\{ E\left(\boldsymbol{x}^{(k)}; \{b_i(\sigma_i), b_\alpha(\sigma_\alpha)\}\right) \text{ satisfying (5)–(7)} \right\}$$
$$\sigma_i^{(k)} = b_i^{(\mathrm{LP},k)}(1)$$

where the self-energy is defined according to (4). In the case of degeneracy one picks any of the closest pseudocodewords.

- **Step 2:** Find $\boldsymbol{y}^{(k)}$, the weighted median in the noise space between the pseudocodeword $\boldsymbol{\sigma}^{(k)}$, and the zero 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 $\boldsymbol{y}^{(k)} = \boldsymbol{y}^{(k-1)}$, then $k_* = k$ and the algorithm terminates. Otherwise go to **Step 2** assigning $\boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)} + \varepsilon$ for some very small ε ($+\varepsilon$ prevents decoding into the zero codeword, keeping the result of decoding within the erroneous domain.)

$\boldsymbol{y}^{(k_*)}$ **is the output** 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 pseudocodeword $\boldsymbol{\sigma}^{(k_*)}, \boldsymbol{y}^{(k_*)}$ is the nearest point of the error-surface to the zero codeword.

The algorithm is schematically illustrated in Fig. 1. 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 pseudocodeword distinct from the zero codeword. Our simulations (see discussions below) show that the algorithm converges, and it does so in a relatively small number of iterations. The convergence of the algorithm is translated into the statement that the effective distance between $\boldsymbol{x}^{(n)}$ and the zero codeword does not increase, but typically decreases, with iterations. Once the algorithm converges the resulting pseudocodeword belongs to the error-surface. This observation was tested by shifting the instanton configuration of the noise correspondent to the pseudocodeword toward the zero codeword and observing that the result of decoding is the zero codeword. The effective distance of the coding scheme is approximated by

$$d_{\mathrm{LP}} \approx \min \left\{ \frac{\left(\sum_i \sigma_i^{(k_*)}\right)^2}{\sum_i \left(\sigma_i^{(k_*)}\right)^2} \right\} \tag{10}$$

where the minimum is taken over multiple evaluations of the algorithm. It is not guaranteed that the noise configuration with the lowest possible (of all the pseudocodewords within the decoding scheme) distance is found after multiple evaluations of the algorithm. Also, we do not have a formal proof of the fact that, beginning with a random $\boldsymbol{x}^{(0)}$, our algorithm explores the entire phase space of all pseudocodewords on the error-surface. However our working conjecture is that the right-hand side (RHS) of (10) gives a very tight (if the number of attempts is sufficient) upper bound on the actual effective distance of the coding scheme.

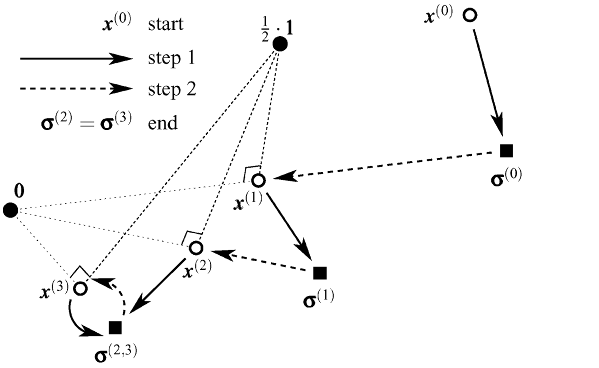
$\boldsymbol{x}^{(0)}$ start
step 1
step 2
$\boldsymbol{\sigma}^{(2)} = \boldsymbol{\sigma}^{(3)}$ end
$\frac{1}{2} \cdot \mathbf{1}$
$\boldsymbol{x}^{(0)}$
$\mathbf{0}$
$\boldsymbol{\sigma}^{(0)}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{\sigma}^{(1)}$
$\boldsymbol{x}^{(3)}$
$\boldsymbol{\sigma}^{(2,3)}$

Fig. 1. Schematic illustration of the pseudocodeword-search algorithm. This example terminates at $k_* = 3$. The point $\mathbf{1}/2 = (1,\ldots,1)/2$ is shown to illustrate that if one draws a straight line through $\mathbf{1}/2$, such that it is perpendicular to the straight line connecting $\mathbf{0} = (0,\ldots,0)$ and $\boldsymbol{\sigma}^{(k)}$, then the straight line must go ε-approximately through $\boldsymbol{x}^{(k+1)}$. [We are thankful to Referee A for making this useful $\mathbf{1}/2$-related observation.]

IV. Examples

In this section, we demonstrate the power of the simple procedure explained in the previous section by considering three popular examples of relatively long regular LDPC codes.

A. *The Tanner* $[155, 64, 20]$ *Code of [18]*

For this code $N = 155$ and $M = 93$. The Hamming distance of the code is known to be $d_{\mathrm{ML}} = 20$. The authors of [7] reported a pseudo codeword with $d = 16.406$. The lowest effective distance configuration found as a result of our search procedure is $d_{\mathrm{LP}} \approx 16.4037$. 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 3000 evaluations of the pseudocodeword-search algorithm) is shown in Fig. 3. Notice that the pseudoweight spectrum gap, defined as the difference between the pseudoweight of the noncodeword minimal pseudocodeword with smallest pseudo-weight and the minimum distance [19], is negative for the code ≈ -3.5963. Thus, the LP decoding performance is strictly worse than the ML decoding performance for SNR → ∞.

B. *The Margulis Code [20] With* $P = 7$

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

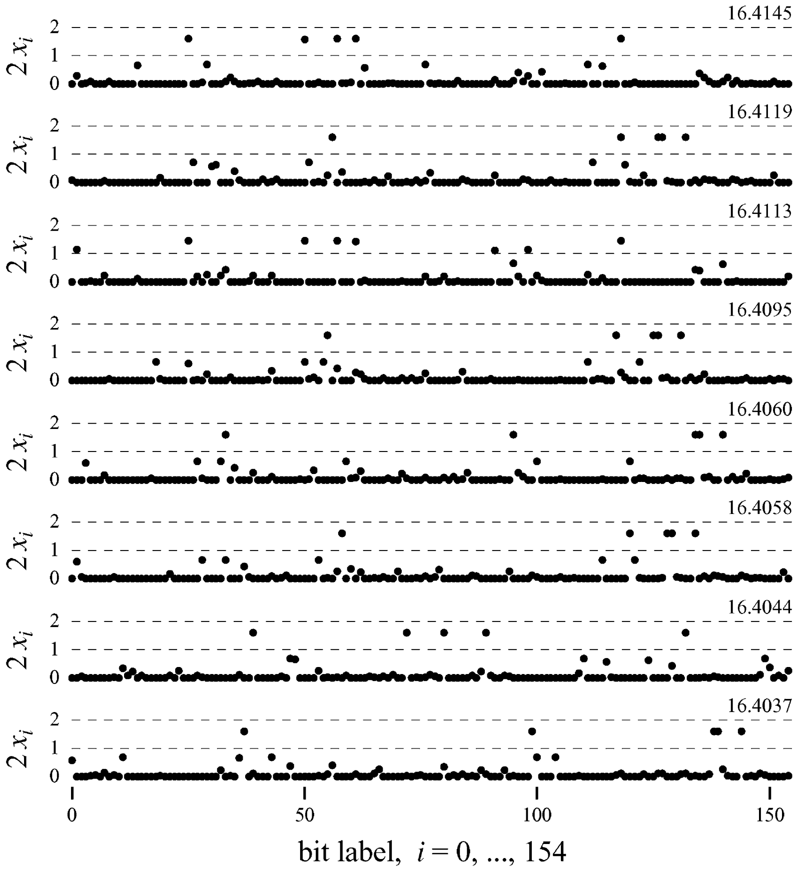

Fig. 2. The eight lowest configurations found by the pseudocodeword-search algorithm for the $[155, 64, 20]$ code. The typical number of evaluations required to reach a stopping point is $5 \div 15$.

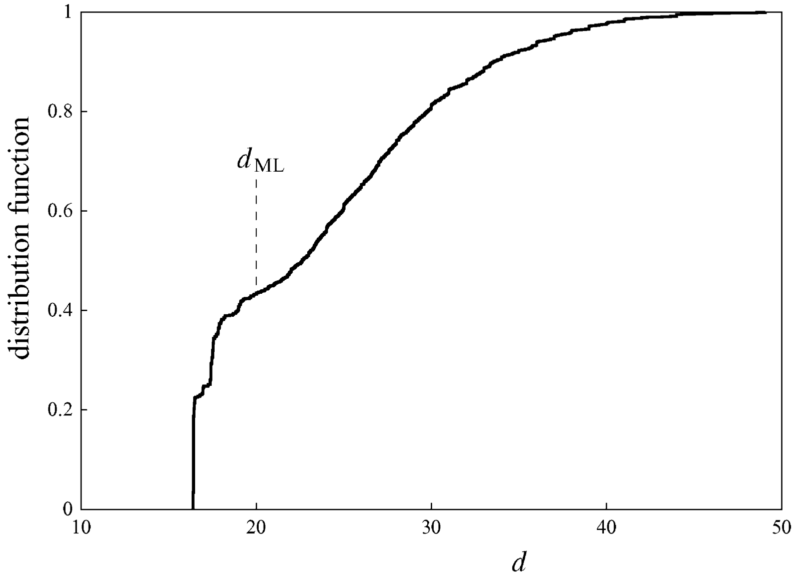

Fig. 3. The frequency spectrum (distribution function) of the effective distance constructed from 3000 attempts of our pseudocodeword search algorithm for the $[155, 64, 20]$ code.

to a pseudocodeword that is not a codeword. Since the pseudoweight spectrum gap is positive in this case, the LP decoding approaches the ML decoding performance for SNR $\to \infty$. The frequency spectra, characterizing the performance of the pseudocodeword-search algorithm for this code, is shown in Fig. 5.

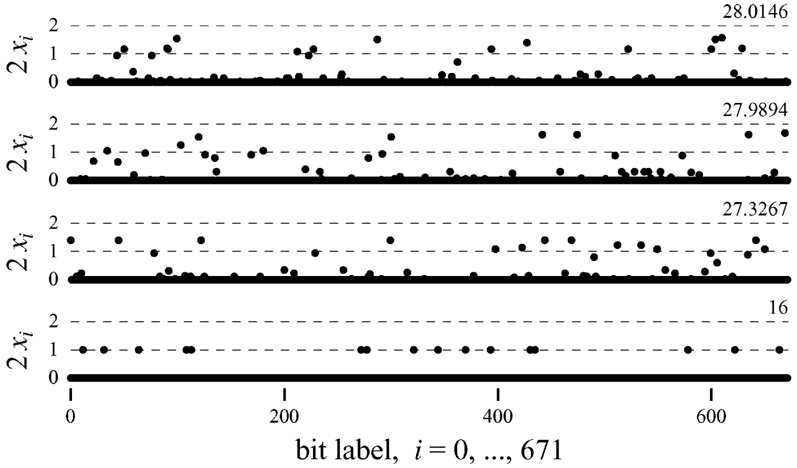

Fig. 4. The four lowest noise configuration found by our pseudocodeword search algorithm for the Margulis $p = 7$ code of [20]. The typical number of evaluations required to reach a stopping point is in between 10 and 20.

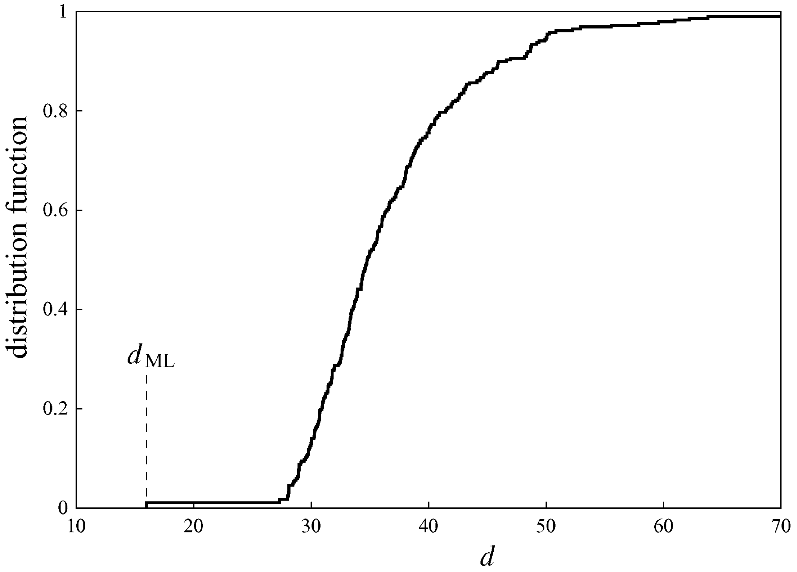

Fig. 5. The frequency spectrum (distribution function) of the effective distance found through multiple attempts of the pseudocodeword-search algorithm for the Margulis $p = 7$ code. The figure is built on 250 evaluations of the pseudocodeword-search algorithm.

C. The Margulis Code [20] With $P = 11$

This code is $N = 2 \cdot M = 2640$ bits long. We have a relatively small number of configurations (30) here because it takes much longer to execute the LP decoding in this case. Some 30 to 60 steps of the pseudocodeword search are required for a typical realization of the algorithm to reach a stopping point. The four lowest configurations are shown in Fig. 6. Obviously, with limited statistics one cannot claim that the noise configuration with the lowest possible effective length has been found. All stopping point configurations found here correspond to pseudocodewords. (The Hamming distance for this code is not known, while the pessimistic upper bound mentioned in [21] is 220.)

V. Conclusion and Discussion

Let us discuss the utility of the pseudocodeword search algorithm proposed in this manuscript. The algorithm gives an efficient way of describing the LP decoding polytope and the pseudocodeword spectra of the code. It approximates the pseudocodeword and the respective noise configuration on the error-surface surrounding the zero codeword, corresponding to the shortest effective distance of the code. Our test shows that the algorithm converges very rapidly. (Even for the 2640 bits code, the longest code we considered, it typically takes only 30 to 60 steps of the pseudocodeword search algorithm to converge.) As aaforementioned, this procedure applies to any linear channel. One only needs to make modifications in (8), (9), (10), and also in the basic equation of Step 2.

One would obviously be interested in extending the pseudocodeword search algorithm to other decodings, e.g., to find the effective minimal distance of the sum-product decoding. We observed, however, that a naive extension of this procedure does not work. The very special feature of the LP-case is that the noise configuration found as a weighted median of the zero codeword and a pseudocodeword ($+\varepsilon$, as in the Step 3 of the pseudocodeword search algorithm) *is not* decoded into the zero

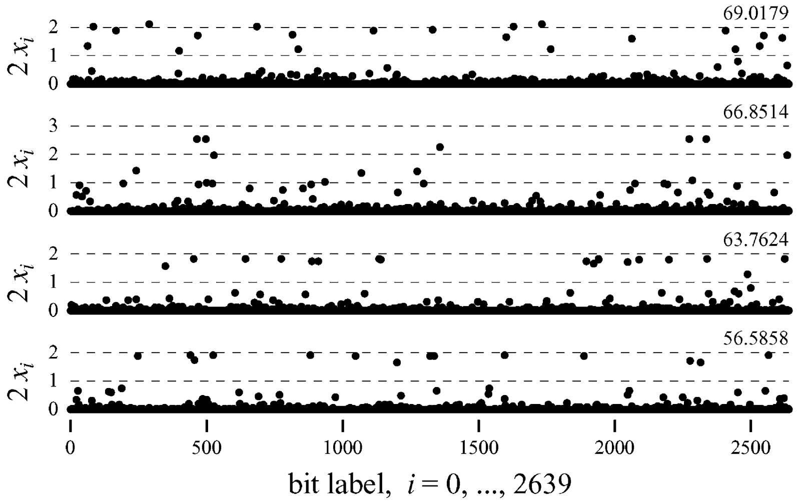

Fig. 6. The four lowest noise configurations found by our pseudocodeword search algorithm for the Margulis $p = 11$ code of [20]. The typical number of the pseudocodeword-search iterations required to reach a stopping point is in between 30 and 60.

codeword. This allows us to proceed with the search algorithm always decreasing the effective distance or at least keeping it constant. It is not yet clear if this key feature of the LP decoding is extendable (hopefully with some modification of the weighted median procedure) to iterative decoding. This question requires further investigation.

Even though the direct attempt to extend the LP-based pseudocodewords-search algorithm to the sum-product decoding failed, we still found an indirect way of using these LP results to analyze the sum-product decoding. The most damaging configuration of the noise found within the pseudocodeword-search procedure becomes a very good entry point for the instanton-amoeba method of [14], designed for finding instanton configurations (most damaging configurations of the noise) for the case of the standard iterative decoding. This hybrid method works well, sometimes resulting in the discovery of pseudocodewords (of the respective iterative scheme) with impressively small effective distance. We attribute this fact to the close relation existing between the LP decoding and the BP decoding [1], [7]–[9]. Some preliminary results of this hybrid analysis are discussed in [22]. Summarizing, the LP-based pseudocodeword search algorithm, complemented and extended by the instanton-amoeba method of [14], provides an efficient practical tool for the analysis of effective distances, most damaging configurations of the noise (instantons) describing the error-floor, and their frequency spectra for an arbitrary LDPC code performing over a linear channel and decoded by LP decoding or iteratively.

After the original version of the manuscript was submitted for publication, we have learned about some important new results concerning reducing complexity of LP-decoding [23], [24]. It is also appropriate to mention here the most recent publications exploring possibilities of LP-decoding improvement [25], [26]. These new techniques and ideas combined with the pseudocodeword-search algorithm open interesting new opportunities for exploring and improving decoding schemes of even longer LDPC codes.

Acknowledgment

The authors acknowledge very useful, inspiring and fruitful discussions with V. Chernyak, R. Koetter, O. Milenkovich, and B. Vasic. We also thank the anonymous referees for their criticisms and suggestions.

References

[1] J. Feldman, M. Wainwright, and D. R. Karger, "Using linear programming to decode linear codes," in *2003 Conf. Inf. Sci. Syst., The John Hopkins Univ.*, Baltimore, MD, Mar. 12–14, 2003.

[2] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[3] C. E. Shannon, "A mathematical theory of communications," *Bell. Syst. Tech. J. 27. 379*, vol. 27, no. 623, 1948.

[4] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[5] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference. San Francisco, CA, Kaufmann, 1988.

[6] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, p. 399, 1999.

[7] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. 3rd Int. Symp. Turbo Codes Related Topics*, Brest, France, Sept. 1–5, 2003, pp. 75–82.

[8] P. O. Vontobel and R. Koetter, "On the relationship between LP decoding and min-sum algorithms decoding," in *Proc. ISITA*, Parma, Italy, 2004.

[9] P. O. Vontobel and R. Koetter, Graph-Cover Decoding and Finite-Length Analysis of Message-Passing Iterative Decoding of LDPC Codes arXiv:cs.IT/0512078..

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

[11] H. A. Bethe, "Statistical theories of superlattices," in *Proc. R. Soc. London A*, 1935, vol. 150, p. 552.

[12] R. Kikuchi, "A theory of cooperative phenomena," *Phys. Rev.*, vol. 81, p. 988, 1951.

[13] T. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annu. Allerton Conf. Communication, Control and Computing*, Monticello, IL, Oct. 2003, pp. 1426–1435.

[14] M. G. Stepanov, V. Chernyak, M. Chertkov, and B. Vasic, "Diagnosis of weakness in error correction codes: A physics approach to error floor analysis," *Phys. Rev. Lett.*, vol. 95, p. 228701, 2005 [Online]. Available: http://www.arxiv.org/cond-mat/0506037

[15] G. D. Forney, R. Koetter Jr., F. R. Kschischang, and A. Reznik, "On the effective weights of pseudocodewords for codes defined on graphs with cycles," in *Codes, Syst., Graphical Models of IMA Vol. Math. Appl.*, B. Marcus and J. Rosenthal, Eds., Minneapolis, MN, 1999/2001, vol. 1233, pp. 101–112, Springer-Verlag.

[16] N. Wiberg, H.-A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs," *Europ. Transaction Telecommunications*, vol. 6, p. 513, 1995.

[17] N. Wiberg, "Codes and decoding on general graphs," Ph.D. thesis, Linköping Univ., Linköping , 1996.

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

[19] P. O. Vontobel, R. Smarandache, N. Kiyavash, J. Teutsch, and D. Vukobratovic, "On the minimal pseudo-codewords of codes from finite geometries," in *ISIT 2005*, Adelaide, Australia, Sep. 4–9, 2005, pp. 980–984 [Online]. Available: http://www.arxiv.org/abs/cs.IT/0508019

[20] G. A. Margulis, "Explicit construction of graphs without short circles and low-density codes," *Combinatorica*, vol. 2, p. 71, 1982.

[21] D. J. C. MacKay and M. J. Postol, "Weaknesses of margulis and ramanujan-margulis low-density parity-check codes," in *Proc. MFCSIT2002*, Galway [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/abstracts/margulis.html

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

[23] M. H. N. Taghavi and P. H. Siegel, "Adaptive linear programming decoding," in *Proc. IEEE ISIT*, Seattle, WA, 2006, arxiv:cs.IT/0601099.

[24] P. O. Vontobel and R. Koetter, "Toward low-complexity linear-programming decoding," in *Proc. 4th Int. Symp. Turbo Codes and Related Topics*, Munich, Germany, Apr. 3–7, 2006, arxiv:cs/0602088.

[25] A. G. Dimakis and M. J. Wainwright, "Guessing facets: Polytope structure and improved LP decoder," in *Proc. IEEE ISIT*, Seattle, WA, 2006 [Online]. Available: arxiv:cs/0608029

[26] M. Chertkov and V. Chernyak, "Loop calculus helps to improve belief propagation and linear programming decodings of low-density-parity-check codes," in *Proc. 44th Allerton Conf.*, Sep. 2006 [Online]. Available: arXiv:cs.IT/0609154