

# GRADIENT DECODING REVISITED

*Phillip A. Regalia*

Department of Electrical Engineering and Computer Science  
Catholic University of America  
Washington, DC 20064

## ABSTRACT

Modern coding applications, including dirty paper coding and information hiding, hinge critically on a classical ‘general decoding problem,’ known to be NP hard. Various attempts to find good solutions at reasonable complexity can be traced throughout the decades, most recently with attempts to achieve the rate-distortion bound in code word quantization. Here we take a step back to examine two computationally simple procedures in this direction: gradient decoding and a simple yet surprisingly effective variant on belief propagation that we dub truthiness propagation.

**Index Terms**— Source compression; code word quantization; rate-distortion theory; information hiding; dirty paper coding; wet paper coding; truthiness propagation.

## 1. INTRODUCTION

The general decoding problem for linear binary codes is to deduce a minimum-weight error pattern associated to a given error syndrome. The problem arose in the earliest studies of error correction codes, since under reasonable assumptions the minimum weight correction to an error-prone received block of bits gives the maximum likelihood estimate of the true code word sent. Although conceptually straightforward, the general decoding problem actually belongs to the class of NP-complete problems [1], [2], [3], thus challenging the development of efficient algorithms for its resolution. The hard aspect of this problem motivates its use in cryptography (see, e.g., [4]) and underlies heuristic search procedures for low-weight error patterns to assess cryptographic strength [5], [6], [2], [7].

The general decoding problem has met with resurgent interest in modern coding applications, including information hiding with robustness [8], [9], [10], multi-user communications using dirty paper coding [11] (which proves equivalent to information hiding), and wet paper coding in steganography [12], [13]. In these more recent applications, the error syndrome is replaced by constraints deriving from side information (typically messages to be hidden or users to be ac-

commodated), thus exposing the duality with source coding and quantization [8], [9], [10], [14]. Indeed, recent algorithmic work has focused on achieving the rate-distortion bound for source quantization (e.g., [15], [16], [17], [18], [19], [20], [21]) using variants of belief propagation combined possibly with heuristic search procedures. Although the claimed results indeed approach the theoretical rate-distortion curve, the algorithm complexity remains appreciable.

The intent of this paper is to examine two approaches of low complexity which converge to source quantization solutions, offering potentially attractive alternatives. The first is gradient decoding, proposed originally in [22] and reviewed in section 3, while the second is a variant of belief propagation that proves vastly simpler than survey propagation and related techniques, and is exposed in section 4. We first review, however, the general decoding problem.

## 2. BACKGROUND

Let  $\mathbf{H}$  be an  $(N-K) \times N$  parity check matrix comprised of ones and zeros, operating on the Galois field  $\text{GF}(2)$  (modulo-2 arithmetic over integers). The set of vectors  $\mathbf{c} \in [\text{GF}(2)]^N$  in its null space defines a linear code  $\mathcal{C}$  of rate  $K/N$ :

$$\mathbf{H}\mathbf{c} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{c} \in \mathcal{C}.$$

The set of vectors  $\mathbf{y} \in [\text{GF}(2)]^N$  which instead produce a prescribed syndrome  $\mathbf{s} \in [\text{GF}(2)]^{N-K}$  comprise the coset  $\mathcal{C}(\mathbf{s})$  for that syndrome:

$$\mathcal{C}(\mathbf{s}) = \{\mathbf{y} \in [\text{GF}(2)]^N : \mathbf{H}\mathbf{y} = \mathbf{s}\}.$$

The code  $\mathcal{C}$  induced by  $\mathbf{H}$  is simply the coset for the zero syndrome:  $\mathcal{C} = \mathcal{C}(\mathbf{0})$ . Note that with  $\mathbf{y} \in \mathcal{C}(\mathbf{s})$  and  $\mathbf{c} \in \mathcal{C}$ , their modulo-2 sum remains in the coset:  $\mathbf{y} + \mathbf{c} \in \mathcal{C}(\mathbf{s})$ .

Suppose we are given an  $N$ -element binary vector  $\mathbf{w} \in [\text{GF}(2)]^N$  and consider the problem of finding an  $N$ -element binary vector  $\mathbf{y} \in [\text{GF}(2)]^N$  which minimizes the Hamming distance  $d(\mathbf{x}, \mathbf{y})$  subject to  $(N-K)$  parity constraints:

$$\min_{\mathbf{y}} d(\mathbf{w}, \mathbf{y}), \quad \text{subject to } \mathbf{s} = \mathbf{H}\mathbf{y} \quad (1)$$

This problem arises in information hiding [8], [9], [10], dirty paper coding [11], and wet paper coding [12, 13].

This work is supported by the National Science Foundation under grant CCF 0634757.

If the bits in  $\mathbf{w}$  hail from a uniform source, then from rate distortion theory [23], the minimum average distortion

$$D_{\min} = \frac{1}{2^N} \sum_{\mathbf{w} \in [\text{GF}(2)]^N} \min_{\mathbf{y}: \mathbf{s} = \mathbf{H}\mathbf{y}} d(\mathbf{y}, \mathbf{w})$$

relates to the code rate  $K/N$  via the rate-distortion curve [23]

$$H_2(D_{\min}) \geq 1 - (K/N)$$

where  $H_2(p) = -p \log p - (1-p) \log(1-p)$  is the binary entropy function.

Finding  $\mathbf{y}$  is equivalent to finding a minimum weight vector consistent with the syndrome:

$$\min_{\mathbf{e}} d(\mathbf{e}, \mathbf{0}) \quad \text{subject to} \quad \mathbf{s} = \mathbf{H}\mathbf{e}$$

(This is the general decoding problem in an earlier form). This vector  $\mathbf{e}$  is the *coset leader* of  $\mathcal{C}(\mathbf{s})$ , and the solution to (1) becomes  $\mathbf{y} = \mathbf{w} + \mathbf{e}$  (using the modulo-2 sum). The minimum weight problem, in turn, is known to belong to the class of NP-complete problems [1], [6], [2], [3], and thus the decoding problem (1) is hard.

Conventional belief propagation decoding applied to this problem, with  $\mathbf{w}$  a ‘‘received’’ vector,  $\mathbf{s}$  the ‘‘side information,’’ and  $\mathbf{y}$  the ‘‘hidden measurement’’ (as in [16]), does not, in general, converge to a meaningful solution, unless  $\mathbf{w}$  is fortuitously in a decoding vicinity of  $\mathcal{C}(\mathbf{s})$  (e.g., [20]). As such, other candidate approaches must be studied. This work compares gradient adaptation, proposed originally in [22] for decoding and apparent also in [24], [25], as well as a modified form of belief propagation proposed in section 4.

### 3. GRADIENT ADAPTATION ALGORITHM

Let  $\mathbf{s}$  be a prescribed syndrome, and  $\mathbf{y}$  a candidate vector that we are attempting to place within the coset  $\mathcal{C}(\mathbf{s})$ . Suppose  $\mathbf{t} = \mathbf{H}\mathbf{y}$  is the actual syndrome for  $\mathbf{y}$ , so that

$$t_i = \sum_{j=1}^N h_{ij} y_j$$

Conversion to antipodal (or  $\pm 1$ ) form is accomplished with an exponential to base  $(-1)$ :

$$\begin{aligned} \tau_i &\triangleq (-1)^{t_i} = (-1)^{\sum_j h_{ij} y_j} \\ &= \prod_{j: h_{ij}=1} (-1)^{y_j} = \prod_{j: h_{ij}=1} \eta_j \end{aligned}$$

in which we set  $\eta_j = (-1)^{y_j} \in \{-1, +1\}$ . Adjusting  $\mathbf{y}$  so that  $\mathbf{t}$  matches a prescribed syndrome is equivalent to adjusting the antipodal components in  $\boldsymbol{\eta}$  so that the antipodal syndrome  $\boldsymbol{\tau}$  matches  $\boldsymbol{\sigma} = (-1)^{\mathbf{s}}$  (with exponentiation occurring

componentwise). This can be accomplished by introducing the cost function [22]

$$J(\boldsymbol{\eta}) = \|\boldsymbol{\sigma} - \boldsymbol{\tau}(\boldsymbol{\eta})\|^2 = 4d(\mathbf{s}, \mathbf{t})$$

relating thus the Euclidean distance squared  $\|\boldsymbol{\sigma} - \boldsymbol{\tau}(\boldsymbol{\eta})\|^2$  of antipodal terms to the Hamming distance  $d(\mathbf{s}, \mathbf{t})$  of binary terms. Since both  $\boldsymbol{\sigma}$  and  $\boldsymbol{\tau}$  have  $N-K$  entries of  $\pm 1$ , we may develop the Euclidean distance squared as

$$\begin{aligned} J(\boldsymbol{\eta}) &= \|\boldsymbol{\sigma} - \boldsymbol{\tau}(\boldsymbol{\eta})\|^2 \\ &= \|\boldsymbol{\sigma}\|^2 + \|\boldsymbol{\tau}\|^2 - 2\langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle = 2(N - K - \langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle) \end{aligned}$$

From this, minimizing  $J$  is equivalent to maximizing the inner product

$$\langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle = \sum_{i=1}^{N-K} \sigma_i \tau_i$$

Maximization may be accomplished with a gradient ascent algorithm, in which each  $\eta_k$  is now allowed to become a continuous parameter, confined to the interval  $[-1, +1]$ ; the vector  $\boldsymbol{\eta}$  thus lies in an  $N$ -dimensional hypercube:  $|\eta_k| \leq 1$  for all  $k$ . The inner product  $\langle \boldsymbol{\sigma}, \boldsymbol{\tau}(\boldsymbol{\eta}) \rangle$  then becomes a multilinear function of  $\boldsymbol{\eta}$ , from which it follows that all maxima occur at vertices of the hypercube [22].

If  $l$  denotes an iteration index, a gradient algorithm appears as

$$\eta_j(l+1) = \eta_j(l) + \mu \frac{\partial \langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle}{\partial \eta_j}, \quad j = 1, 2, \dots, N,$$

where  $\mu > 0$  is a step size parameter, using the gradient

$$\frac{\partial \langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle}{\partial \eta_j} = \sum_i \sigma_i \frac{\partial \tau_i}{\partial \eta_j} \quad \text{with} \quad \frac{\partial \tau_i}{\partial \eta_j} = h_{ij} \prod_{\substack{k: h_{ik}=1 \\ k \neq j}} \eta_k(l)$$

Two modes of operation may be attempted:

- The parameters are initialized as  $\boldsymbol{\eta}(0) = (-1)^{\mathbf{w}}$  (the antipodal representation of  $\mathbf{w}$ ) and the algorithm seeks a closest member from the coset  $\mathcal{C}(\mathbf{s})$ ;
- The parameters are initialized as  $\boldsymbol{\eta}(0) = (-1)^{\mathbf{0}}$  (the all-one vector), in order to seek a minimum weight member from the coset  $\mathcal{C}(\mathbf{s})$ .

In either case, it is readily observed that the algorithm often converges to a maximum of  $\langle \boldsymbol{\sigma}, \boldsymbol{\tau} \rangle$  for which  $\boldsymbol{\tau} \neq \boldsymbol{\sigma}$ , so that the resulting  $\mathbf{y}$  is not in the prescribed coset  $\mathcal{C}(\mathbf{s})$ . It is thus preferable to modify the algorithm so that  $\mathbf{y}$  is constrained to stay in the coset.

To this end, let  $\mathbf{x}$  denote a particular vector in the coset:  $\mathbf{s} = \mathbf{H}\mathbf{x}$ ; its determination will be addressed below. Let  $\mathbf{G}$  be a  $K \times N$  generator matrix that spans the orthogonal complement to  $\mathbf{H}$ , i.e.,  $\mathbf{H}\mathbf{G} = \mathbf{0}$  in  $\text{GF}(2)$ . Each member  $\mathbf{y}$  of  $\mathcal{C}(\mathbf{s})$  may then be expressed as

$$\mathbf{y} = \mathbf{x} + \mathbf{G}\mathbf{z}$$

where  $\mathbf{z} \in [\text{GF}(2)]^K$  is comprised of  $K$  information bits. A minimum weight vector in the coset may then be found by adjusting  $\mathbf{z}$  to minimize the Hamming distance  $d(\mathbf{x}, \mathbf{Gz})$ . As above, the cost function can be put in antipodal form as

$$\begin{aligned} 4d(\mathbf{w}, \mathbf{Gz}) &= \|\boldsymbol{\xi} - \boldsymbol{\chi}\|^2 \\ &= \|\boldsymbol{\xi}\|^2 + \|\boldsymbol{\chi}\|^2 - 2\langle \boldsymbol{\xi}, \boldsymbol{\chi} \rangle \\ &= 2(N - \langle \boldsymbol{\xi}, \boldsymbol{\chi} \rangle) \end{aligned}$$

involving terms in the generator space:

$$\begin{aligned} \langle \boldsymbol{\xi}, \boldsymbol{\chi} \rangle &= \sum_{i=1}^N \xi_i \chi_i \\ \xi_i &= (-1)^{x_i} \\ \chi_i &= (-1)^{\sum_j g_{ij} z_j} = \prod_{j: g_{ij}=1} \zeta_j \\ \zeta_j &= (-1)^{z_j} \end{aligned}$$

We then allow each  $\zeta_j$  to vary continuously in the interval  $[-1, +1]$ . A gradient algorithm to maximize  $\langle \boldsymbol{\xi}, \boldsymbol{\chi} \rangle$  then appears as

$$\zeta_j(l+1) = \zeta_j(l) + \mu \frac{\partial \langle \boldsymbol{\xi}, \boldsymbol{\chi} \rangle}{\partial \zeta_j}, \quad j = 1, 2, \dots, N,$$

using the gradient

$$\frac{\partial \langle \boldsymbol{\xi}, \boldsymbol{\chi} \rangle}{\partial \zeta_j} = \sum_{i=1}^N \xi_i \frac{\partial \chi_i}{\partial \zeta_j} \quad \text{with} \quad \frac{\partial \chi_i}{\partial \zeta_j} = g_{ij} \prod_{\substack{k: g_{ik}=1 \\ k \neq j}} \zeta_k(l)$$

Note that using a low density generator matrix ensures a small number of ones in  $\mathbf{G}$ , thus reducing the number of products to form in calculating the gradient.

To determine a particular solution  $\mathbf{w}$ , a simple variant on Stern's algorithm [5] (as used also in [13]) can be employed: Choose  $N-K$  columns of  $\mathbf{H}$  randomly, to build a square matrix  $\mathbf{A}$ . If  $\mathbf{A}$  is invertible in  $\text{GF}(2)$ , we may solve the system  $\mathbf{s} = \mathbf{A}\mathbf{b}$  in  $\text{GF}(2)$  for  $\mathbf{b}$ , whose elements are copied into the appropriate positions for  $\mathbf{w}$ , with the remaining terms set to zero. The  $\mathbf{w}$  so constructed fulfills  $\mathbf{s} = \mathbf{H}\mathbf{w}$ , and has weight  $(N-K)/2$  on average (e.g., [13]). A similar algorithm has been employed previously in [6], [2], but applied to the generator matrix  $\mathbf{G}$  rather than the parity-check matrix  $\mathbf{H}$ .

#### 4. TRUTHINESS PROPAGATION

Consider the factor graph of the generator equation  $\mathbf{x} = \mathbf{Gz}$ , as in Figure 1. If  $\mathbf{x}$  is not in the column space of the generator matrix  $\mathbf{G}$ , this gives an inconsistent system; belief propagation is a candidate attempt to choose the information bits in  $\mathbf{z}$  to best reconcile  $\mathbf{Gz}$  with  $\mathbf{x}$ . The conventional belief propagation algorithm [26] passes messages along edges in the graph between variable nodes (denoted as circles) and factor nodes

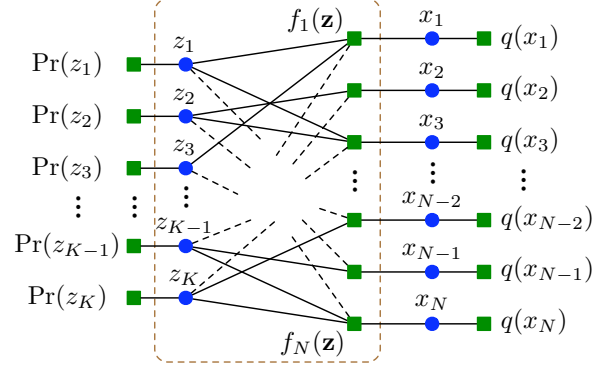


Fig. 1. Factor graph for a low density generator matrix.

(denoted as squares). These messages consist of two-element vectors containing pseudo-probabilities that sum to one. We review first the standard belief propagation algorithm to grok its deficiency for code word quantization, and then present a modification which corrects this defect.

Belief propagation updates at the variable nodes as

$$m_{z_i \rightarrow f_j}(z_i) = \beta_j \Pr(z_i) \prod_{\substack{k \in F(i) \\ k \neq j}} m_{f_k \rightarrow z_i}(z_i) \quad (2)$$

where  $m_{z_i \rightarrow f_j}(z_i)$  denotes the message sent from node  $z_i$  to node  $f_j$ ,  $F(i)$  contains the indices  $k$  whose factor nodes  $f_k$  connect to variable node  $z_i$ , and the scale factor  $\beta_j$  is chosen to ensure evaluations sum to one:

$$m_{z_i \rightarrow f_j}(0) + m_{z_i \rightarrow f_j}(1) = 1.$$

The factor node  $\Pr(z_i)$  reflects any *a priori* information on the information bit  $z_i$ ; when none is available, this can be set to  $\Pr(z_i = 1) = 0.5 + d_i$  where  $d_i$  is a small dithering noise.

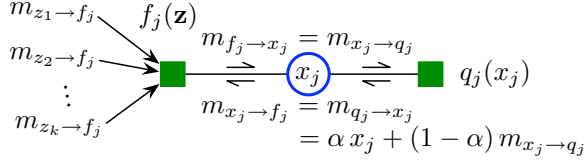
The update equations at the parity check nodes read as

$$\begin{aligned} m_{f_j \rightarrow z_i}(z_i) &= \frac{1}{2} \left\{ 1 + (-1)^{z_i} (1 - 2m_{x_j \rightarrow f_j}(1)) \right. \\ &\quad \left. \times \prod_{\substack{k \in V(j) \\ k \neq i}} (1 - 2m_{z_k \rightarrow f_j}(1)) \right\} \quad (3) \end{aligned}$$

where  $V(j)$  contains the indices  $k$  of the variable nodes  $z_k$  which enter into the  $j$ -th parity check. By conventional belief propagation, the message  $m_{x_j \rightarrow f_j}$  is that sent from the factor node  $q_j(x_j)$  (right-most squares in Figure 1) to the variable node  $x_j$ . Setting  $q_j(1)$  to some "soft" probability skewed towards the hard value  $x_j$  gives an algorithm which, in general, converges to essentially meaningless probabilities. Setting  $q_j(1) = x_j$  (hard values) gives, in general, an inconsistent system, resulting in numerical singularities.

A more intelligent choice is to set

$$q_j(1) = \alpha x_j + (1 - \alpha) m_{x_j \rightarrow q_j}(1) \quad (4)$$



**Fig. 2.** Messages at  $j$ -th parity-check node  $f_j(\mathbf{z})$ , to illustrate modified algorithm. The constraint node  $q_j$  sends back a convex combination of its hard constraint  $x_j$  and its incoming message  $m_{x_j \rightarrow q_j}$ , controlled by a “truthiness” factor  $\alpha$  in (4).

where  $0 < \alpha < 1$  and

$$m_{x_j \rightarrow q_j}(x_j) = m_{f_j \rightarrow x_j}(x_j) = \frac{1}{2} \left\{ 1 + (-1)^{x_j} \prod_{k \in V(j)} (1 - 2m_{z_k \rightarrow f_j}(1)) \right\}$$

is the message sent from factor node  $f_j$  to variable node  $x_j$ ; cf. Figure 2. The value  $m_{f_j \rightarrow x_j}(1)$  is the probability that the information bits  $\{z_k\}$  which impinge on the  $j$ -th parity check produce an odd parity [27], giving thus the “natural parity” (i.e., without the constraint from  $x_j$ ) at that check node.

The rationale for choosing the convex combination in (4) is that if the natural parity agrees with the constraint  $x_j$ , the function node maintains this “hard” constraint. If instead the natural parity is opposite to the constraint  $x_j$ , the probability fed from  $q_j$  is softened, thus allowing further refinement in the message passing algorithm. The operation at the constraining factor node is akin to a system which feeds back not the actual “truth” (as strict belief propagation would do), but rather what it “wishes” to be true, via injection of the natural parity. Accordingly, we dub the algorithm *truthiness<sup>1</sup> propagation*, obtained by running (2), (4), and then (3), and iterating. Interestingly, this simple modification gives an algorithm that is observed to converge to a quantizing solution, yet features negligible overhead compared to standard belief propagation. It is thus an order of magnitude simpler than the survey propagation methods developed in [15], [18].

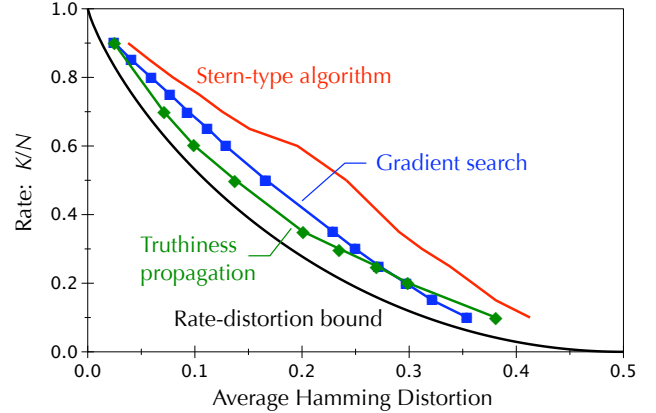
## 5. SIMULATION RESULTS

Figure 3 shows the average distortion versus code rate  $K/N$  for the various methods, using  $N \approx 300$  bits, along with the theoretical performance bound  $H_2(D) = 1 - K/N$ . For the gradient and truthiness propagation algorithms, the average distortion is

$$D = \frac{E[d(\mathbf{x}, \mathbf{Gz})]}{N}$$

in which the expectation is over syndromes  $\mathbf{s}$  which generate the coset behind the particular solution  $\mathbf{x}$ . For Stern’s algorithm, the average Hamming weight of  $\mathbf{x}$  is instead plotted;

<sup>1</sup>**truthiness** (noun): “The quality of preferring concepts or facts one wishes to be true, rather than concepts or facts known to be true” (Merriam-Webster).



**Fig. 3.** Rate-distortion performance for the gradient and truthiness propagation algorithms, using  $N = 300$ .

this  $\mathbf{x}$  serves as the target that  $\mathbf{Gz}$  attempts to approximate in the other two methods. The generator matrices were low density varieties obtained from the methods of [28]<sup>2</sup>.

For the truthiness propagation algorithm, we observe that reducing  $\alpha$  generally gives lower quantization distortion, until a lower limit on  $\alpha$  is reached, below which the algorithm fails to converge. The critical lower value for  $\alpha$  decreases with the code rate, but is not yet available in analytic form.

A performance gap is still visible, although we emphasize the short block length employed ( $N \approx 300$ ), which is more realistic of streaming applications in which latency and queuing considerations favor shorter packet lengths. In such cases, the luxury of using long block lengths is not an option. In addition, although the design methods of [28] claim “good” low density matrices for long block lengths, it is unknown what quality those designs furnish with shorter block lengths. Thus some performance loss compared to the theoretical rate-distortion curve is to be expected in the short block length case. One expects, however, that message passing algorithms should be more sensitive to connectivity patterns among nodes, in contrast to the gradient search procedure which depends instead on the column space spanned by the generator matrix.

## 6. CONCLUDING REMARKS

We have examined two computationally efficient procedures for code word quantization: gradient decoding and truthiness propagation. Both algorithms are iterative, offer linear complexity in the block length per iteration, and converge to meaningful solutions, although a performance gap is still observed with respect to the theoretical rate-distortion curve. The algorithms are much simpler in implementation and complexity compared to survey propagation [15], [17], [18], and

<sup>2</sup>See <http://lthcwww.epfl.ch/research/ldpcopt/>

are devoid of the cumbersome pruning and decimation steps required of that procedure. They also avoid the exponential complexity growth of heuristic search procedures [5], [7], [6], [2], [21] (albeit with controlled exponents), and are thus viable alternatives for real-time applications.

## 7. REFERENCES

- [1] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Information Theory*, vol. 24, no. 3, pp. 384–386, May 1978.
- [2] T. Johansson and F. Jönsson, "On the complexity of some cryptographic problems based on the general decoding problem," *IEEE Trans. Information Theory*, vol. 48, no. 10, pp. 2669–2678, Oct. 2002.
- [3] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of Reed–Solomon codes is NP-hard," *IEEE Trans. Information Theory*, vol. 51, no. 7, pp. 2249–2256, July 2005.
- [4] W. Trappe and L. C. Washington, *Introduction to Cryptography with Coding Theory*, Prentice-Hall, Upper Saddle River, NJ, 2nd edition, 2006.
- [5] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, G. Cohen and J. Wolfman, Eds., vol. 338, pp. 106–113. Springer, Berlin, 1989.
- [6] A. Canteaut and F. Chabaud, "A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511," *IEEE Trans. Information Theory*, vol. 44, no. 1, pp. 367–378, Jan. 1998.
- [7] C.-C. Shih, C. R. Wulff, C. R. P. Hartmann, and C. K. Mohan, "Efficient heuristic search algorithms for soft-decision decoding of linear block codes," *IEEE Trans. Information Theory*, vol. 44, no. 7, pp. 3023–3038, Nov. 1998.
- [8] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Information Theory*, vol. 49, no. 3, pp. 563–593, Mar. 2003.
- [9] R. J. Barron, B. Chen, and G. W. Wornell, "The duality between information embedding and source coding with side information and some applications," *IEEE Trans. Information Theory*, vol. 49, no. 5, pp. 1159–1180, May 2003.
- [10] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Information Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.
- [11] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Information Theory*, vol. 29, no. 3, pp. 439–441, May 1983.
- [12] J. Fridrich, M. Goljan, and D. Soukal, "Wet paper codes with improved coding efficiency," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 1, pp. 102–110, Mar. 2006.
- [13] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Processing*, vol. 10, no. 53, pp. 3923–3935, Oct. 2005.
- [14] A. Khisti, E. Martinian, and G. W. Wornell, "Information embedding with distortion side information," in *Int. Symp. Information Theory*, Seattle, WA, July 2006, pp. 183–187.
- [15] M. Mézard and R. Zecchina, "Random  $K$ -satisfiability problem: From an analytic solution to an efficient algorithm," *Physical Review E*, vol. 66, no. 5, 2002.
- [16] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 1491–1513, Oct. 2002.
- [17] E. Martinian and J. S. Yedidia, "Iterative quantization using codes on graphs," in *Allerton Conf. Communications, Control and Computing*, Oct. 2003.
- [18] M. J. Wainwright and E. Maneva, "Lossy source coding via message-passing and decimation over generalized codewords of LDGM codes," in *Int. Symp. Information Theory*, Adelaide, Australia, Sept. 2005.
- [19] E. Martinian and M. J. Wainwright, "Low-density constructions for lossy compression, binning, and coding with side information," in *IEEE Information Theory Workshop*, Punta del Este, Uruguay, 2006, pp. 263–264.
- [20] M. J. Wainwright, "Sparse graph codes for side information and binning," *IEEE Signal Processing Mag.*, vol. 24, no. 7, pp. 47–57, Sept. 2007.
- [21] A. Gupta and S. Verdú, "Nonlinear sparse-graph codes for lossy compression of discrete nonredundant sources," in *Proc. Information Theory Workshop*, Lake Tahoe, CA, Sept. 2007, pp. 541–546.
- [22] K. H. Farrell, L. D. Rudolph, C. R. P. Hartmann, and L. D. Nielsen, "Decoding by local optimization," *IEEE Trans. Information Theory*, vol. 29, no. 5, pp. 740–743, Sept. 1983.
- [23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, Hoboken, NJ, 2nd edition, 2007.
- [24] A. Hämmäläinen and J. Henriksson, "Convolutional decoding using recurrent neural networks," in *Proc. Joint Conf. Neural Networks*, San Diego, CA, 1999, vol. 5, pp. 3323–3327.
- [25] K. Hueske, J. Götze, and E. Coersmeier, "Optimization approaches for a recurrent neural network convolutional decoder," in *Proc. Int. Workshop on Machine Learning for Signal Processing*, Thessaloniki, Greece, Aug. 2007.
- [26] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498–519, February 2001.
- [27] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Information Theory*, vol. 2, pp. 21–28, 1962.
- [28] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.