MONOTONIC OPTIMIZATION BASED DECODING FOR LINEAR CODES

P. T. Khoa[†], **T. T. Son**[‡], **H. D. Tuan**[†], *and* **H. Tuy**[*]

[†]School of Electrical Engineering and Telecommunications, University of New South Wales
NSW 2051, Sydney, AUSTRALIA; Email: tran.phan@student.unsw.edu.au, h.d.tuan@unsw.edu.au

[‡]Department of Electrical and Computer Engineering, Toyota Institute of Technology,
Nagoya 468-8511, JAPAN; Email: ttson@toyota-ti.ac.jp

[*]Institute of Mathematics, Hanoi, VIETNAM; Email: htuy@math.ac.vn

ABSTRACT

A new efficient method is developed for optimal maximum likelihood (ML) decoding of an arbitrary binary linear code based on data received from a Gaussian channel. The decoding algorithm is based on minimization of a difference of two monotonic objective functions subject to the 0-1 constraint of bit variables. The iterative process converges to the global optimal ML solution after a finite number of steps. The proposed algorithm's computational complexity depends on the input sequence length k which is much less than the codeword length n, especially for codes with small code rates. The viability of the developed method is verified through simulations on different coding schemes.

1. INTRODUCTION

One of the fundamental problems in communications is to construct an encoding and a decoding systems for reliable communication over a noisy channel. In the early 1960s, Gallager invented a class of Low-Density Parity-Check (LDPC) codes [3]. Their subsequent rediscovery was made by Mackay et al in [4], which has generated considerable attention within the coding community. The most popular technique for suboptimal decoding is the belief propagation (BP) algorithm (see e.g. [5]), where messages are iteratively passed across a factor graph modeling the structure of the code. When the factor graph contains cycles, BP may not propagate the whole graph and so locates a solution which may not even be locally optimal. Yuille [9] proposed a discrete iterative algorithm called concave-convex procedure (CCCP) to find a local minimum of the Bethe free energy which corresponds to the fixed point of BP. The CCCP decomposes the free energy into concave and convex parts, which are also widely known in the global optimization community as d.c. (difference of two convex functions) representation of the free energy [6], and locally decreases the free energy by the classical Frank and Wolf algorithm [2]. Recently, Feldman et al. [1] introduced *Linear Programming* (LP) relaxation for the ML decoding. This involves minimizing a linear function over a well-constructed codeword polytope. However, the complexity of the polytope representation increases prohibitively when the size of the encoder increases moderately.

In this paper, we develop a practically efficient method to solve the exact ML decoding for an arbitrary binary linear code, using the discrete monotonic optimization [7, 8]. First, the *modulo 2* operation is re-expressed as a continuous function on the conventional linear finite dimensional space. The objective function of our global optimization problem is then re-written as a d.m. (difference of two monotonic functions) function. As far as the globally optimal solution of the discrete optimization is concerned, there are several advantages of using the monotonic optimization [7, 8] over partial convex (like d.c.) optimization [6]: (i) the monotonicity or partial monotonicity of functions are more easily recognizable than their partial convexity; (ii) over a box, since monotonic functions attain its optimal points at one of its smallest and largest vertices, by using the functions' partial monotonicity instead of partial convexity, it is less computationally consuming to compute its bounds; (iii) the monotonic optimization based algorithm is finite, yielding the exact global optimal solution [8], while partial convexity based optimization finite algorithm yields only an approximate optimal solution with a prescribed tolerance [6]. Although our approach can work well for other discrete optimizations arising from decoding and detection such as multi-user detection in CDMA and other integer least square problems, we will focus in this paper only on solution method for decoding linear codes. It is interesting to note that our d.m. representation is the most efficient for LDPC codes as shown by both theory and simulation.

The paper is organized as follows. In Section 2, the formulation of the ML problem using monotonic optimization is presented. Then, an appropriate global optimization algorithm for ML decoding is developed in Section 3. Experimental results come in Section 4, followed by the conclusion.

The notations used in the paper are as follows. An input bit sequence u and codeword x are column vectors. Throughout this paper, the generator matrix G of LDPC is of size $n \times k$ with rows $\{g_i\} \in \{0,1\}^k$ (so the transmission redundancy is $n-k$ bits). For simplicity, $G \oplus_2 u$ denotes $Gu \pmod 2$ and so the set of codewords generated by the generator matrix G is defined as

$$C = \{x \in \{0,1\}^n \,|\, x = G \oplus_2 u, u \in \{0,1\}^k\}.$$

We define the box $[a,b] \in R^k$ as the set of all $x \in R^k$ such that $a_i \le x_i \le b_i, i = 1...k$.

2. ML DECODING WITH MONOTONIC OPTIMIZATION

This section describes the formulation of the ML decoding problem for an arbitrary linear code using discrete monotonic optimization.

2.1. Optimal ML Decoding

Consider a communication system with an (n,k) block code represented by a generator matrix G and the Binary Phase Shift Keying (BPSK) modulation scheme. That is, the modulator maps the binary symbol $x_i \in \{0,1\}$ to the signal constellation $v_i \in \{+1,-1\}$

according to the rule:

$$v_i = \begin{cases} +1 & x_i = 1 \\ -1 & x_i = 0. \end{cases}$$

Let u, x be the input bit sequence and the corresponding encoder output codeword respectively, i.e. $x = G \oplus_2 u$. The codeword x is then modulated before being transmitted over the AWGN (additive white Gaussian noise) channel. Then the received vector is $\tilde{y} = v + n = (2x - 1) + n$, where n is Gaussian noise vector with zero mean and variance $\delta^2 I$.

Given the received signal $\tilde{y}$, the ML detector attempts to estimate the codeword $\tilde{x}$ which maximizes the likelihood function $\mathcal{P}(\tilde{y}|x)$. Based on the generator matrix G, the direct ML criterion for detection of $\tilde{u}$ is the following optimization problem

$$\tilde{u} = \arg \min_{u \in \{0,1\}^k} \{||(2(G \oplus_2 u) - 1) - \tilde{y}||^2\} \tag{1}$$

which is an integer least squares problem under modulo 2 constraints. Note that, in general, an integer least squares problem alone is *NP-hard*, i.e. an exhaustive search to evaluate all the 2^k possible bit sequences may be needed to detect the optimal solution $\tilde{u}$. Our aim is to develop a *practically* efficient algorithm for the optimal solution of this difficult optimization problem. Details of our algorithm are introduced next.

2.2. Discrete Monotonic Optimization

The central idea of our proposed method is to optimize a suitable d.m. objective function. Therefore, firstly, the involved *modulo 2* operation needs to be properly re-arranged as a conventional one, acting on the linear finite dimensional space, making it possible to write the objective function $f(u)$ as a d.m. function. One way to handle the *modulo 2* constraints is to express

$$G \oplus_2 u = \frac{1 - \cos(\pi G u)}{2}$$

which is a continuous function for $u \in [0, 1]^k$.

Now, the objective function $f(u) = ||(2(G \oplus_2 u) - 1) - \tilde{y}||^2$ in (1) can be readily written as

$$f(u) = || - \tilde{y} - \cos(\pi G u)||^2.$$

Consequently, the problem (1) is equivalent to

$$\tilde{u} = \arg \min_{u \in \{0,1\}^k} \{\sum_{i=1}^{n} \tilde{y}_i \cos(\pi g_i u)\}.$$

With the definition $I_- = \{\, i \mid \tilde{y}_i < 0\}$ and $I_+ = \{\, i \mid \tilde{y}_i > 0\}$, we express $f(u)$ by

$$\begin{aligned} f(u) &= \sum_{i \in I_+} |\tilde{y}_i| \cos(\pi g_i u) - \sum_{i \in I_-} |\tilde{y}_i| \cos(\pi g_i u) \\ &= f^+(u) - f^-(u) \end{aligned} \tag{2}$$

where

$$\begin{aligned} f^+(u) &= \sum_{i=1}^{n} |\tilde{y}_i| 2^{\lceil \frac{g_i u}{2} \rceil} + \sum_{i \in I_+} |\tilde{y}_i| \cos(\pi g_i u) \\ f^-(u) &= \sum_{i=1}^{n} |\tilde{y}_i| 2^{\lceil \frac{g_i u}{2} \rceil} + \sum_{i \in I_-} |\tilde{y}_i| \cos(\pi g_i u) \end{aligned} \tag{3}$$

which are increasing functions with respect to the variable $u \in \{0, 1\}^k$.

We now reduce the original modulo 2 constrained integer least square optimization problem (1) to the following optimization problem

$$\min_u \; f^+(u) - f^-(u) \text{ s.t. } u \in \{0, 1\}^k \tag{4}$$

which belongs to the family of d.m. optimization [7, 8].

Our next section is devoted to an effective algorithmic development toward the solution of (4) by exploring more combinatoric nature of $0 - 1$ variables.

3. GLOBAL OPTIMIZATION FOR ML DETECTION

Note that the constraint in (4) is the same as

$$u \in [a, b] \cap S^* \tag{5}$$

where $a = 0 \in \mathbb{R}^k_+$, $b = [1, \ldots, 1]^T \in \mathbb{R}^k_+$ and $S^* = \{0, 1\}^k$ the vertex set of $[a, b]$.

To solve the optimization problem (4) under the constraint (5) globally, we adapt the Branch-Reduce-and-Bound (BRB) strategy of [8]. It involves in each iteration three basic operations: **branching, reduction, and bounding** which are developed in three subsequent subsections.

3.1. Branching

A partition set has the form $M = [p, q]$ such that for some $I \subset \{1, \ldots, k\}$ and $J \subset \{1, \ldots, k\} \setminus I$:

$$\begin{aligned} M \;=\; & \{u \in \mathbb{R}^k_+ |\, 0 \le u_i \le 1 \; (i \in I), \\ & u_i = 0 \; (i \in J), \; u_i = 1 \; (i \notin I \cup J)\}. \end{aligned}$$

Therefore, it is convenient to write $M = [p, q]^{I,J}$, where $I \subset \{1, \ldots, k\}, J \subset \{1, \ldots, k\} \setminus I$. The partition of M is performed as follows: select an index $k \in I$ and divide M into $M^- = \{u \in M |\, u_k = 0\}$ and $M^+ = \{u \in M |\, u_k = 1\}$.

It can be easily seen that the branching step does not leave out any potential solution for (4) since $M \cap S^* = M^- \cap S^* + M^+ \cap S^*$.

3.2. Reduction

If $M = [p, q]^{I,J}$ and the Current Best Value (CBV) is γ, then we are interested in only those $u \in M \cap S^*$ satisfying

$$f^+(u) - f^-(u) \le \gamma \tag{6}$$

because those not satisfying (6) are definitely not the optimal solution.

Proposition 1. *There exists a point $u \in M$ satisfying (6) only if*

$$f^+(p) \le f^-(q) + \gamma. \tag{7}$$

Any such x is contained in the subset $red_\gamma[p, q] := [p', q'] \subset [p, q]$ where

- *For box $[p, q]$, and $I = \{i \in I : p_i = 0, q_i = 1\}$,*

$$p'_i = \begin{cases} p_i = q_i & i \notin I \\ 0 & f^-(q - e^i) + \gamma \ge f^+(p) \\ 1 & f^-(q - e^i) + \gamma < f^+(p) \end{cases} \tag{8}$$

- *For box* $[p', q]$*, and* $I' = \{i \in I : p'_i = 0, q_i = 1\}$,

$$q'_i = \begin{cases} p'_i = q_i & i \notin I' \\ 1 & f^+(p' + e^i) \leq f^-(q) + \gamma \\ 0 & f^+(p' + e^i) > f^-(q) + \gamma \end{cases} \quad (9)$$

Proof. Firstly, we show that

$$\{u \in [p, q] : \ f^+(u) - f^-(u) \leq \gamma\} \subset [p', q]. \quad (10)$$

Indeed, if $u \in [p, q]$ and $u \not\geq p'$ then there is $i \in I$ such that $u_i = 0$ and $p'_i = 1$ and also according to (8), $f^-(q - e^i) < f^+(p) - \gamma$. Then $f^-(u) \leq f^-(q - e^i) < f^+(p) - \gamma \leq f^+(u) - \gamma$, i.e. $f^+(u) - f^-(u) > \gamma$, showing (10).

Secondly, we show that

$$\{u \in [p', q] : \ f^+(u) - f^-(u) \leq \gamma\} \subset [p', q']. \quad (11)$$

Now, if $u \in [p', q]$ but $u \not\leq q'$ then there is $i \in I'$ such that $u_i = 1$ and $q'_i = 0$ and according to (9), $f^-(q) + \gamma < f^+(p' + e^i)$ and $p'_i = 0$. The latter particularly implies $p' + e^i \leq u$ as well. Then $f^-(u) \leq f^-(q) < f^+(p' + e^i) - \gamma \leq f^+(u) - \gamma$, i.e. $f^+(u) - f^-(u) > \gamma$, showing (11).

The following corollary summarizes our reduction strategy.

Corollary 1. *If* $f^+(p) > f^-(q) + \gamma$ *then* M *can be fathomed. Otherwise,* M *can be replaced by its valid reduction* $[p', q']$ *without losing any better feasible solution than the current best.*

3.3. Bounding

For $M = [p, q]^{I,J}$, we compute a number $\mu(M)$ such that

$$\mu(M) \leq \rho(M) = \min_{u \in M \cap S^*} \{f^+(u) - f^-(u)\}.$$

To ensure convergence, this lower bound must be consistent in the sense that for any infinite nested sequence of boxes M_{k_v} shrinking to a single point u^*,

$$\lim_{v \to \infty} \mu(M_{k_v}) = f(u^*).$$

Of course, the efficiency of an algorithm depends on the used bounding techniques. With d.m representation (2) intact, the less computationally consuming but efficient enough lower bound is

$$\mu(M) = f^+(p) - f^-(q). \quad (12)$$

Note that if M is the partition set with smallest $\rho(M)$ among all partition sets still under consideration in the current iteration of the BRB algorithm, then $\rho(M)$ is the optimal value of the problem, i.e. u such that $\rho(M) = f^+(u) - f^-(u)$ is the globally optimal solution.

3.4. Branch-Reduce-and-Bound Algorithm

Now, having the branching, reduction and bounding developed in the previous subsections, the BRB algorithm [8] adapted to the solution of (4), (5) is as follows.

Initialization. **Let** $\mathcal{P}_1 = \{M_1\}$, $M_1 = [a, b]$, $\mathcal{R}_1 = \emptyset$. **If some feasible solutions** u **are available let** $CBV = f(u)$ **at the best of them (current best value). Otherwise, set** $CBV = \infty$. **Set** k = *1*.

Step 1. **Apply the reduction rule for each box** $[p, q] \in \mathcal{P}_k$. **In particular, delete any box** $[p, q]$ **such that** $f^+(p) > f^-(q) + \gamma$ **for** $\gamma = CBV$. **Let** $\mathcal{P}'_k = \{[p', q'] = red_\gamma[p, q] : [p, q] \in \mathcal{P}_k\}$. **Then, takes the smaller of** $f(p')$ **or** $f(q')$ **if possible for updating** *CBV*.

Step 2. **For each box** $M = [p', q']^{I',J'} \in \mathcal{P}'_k$, **compute a bound** $\mu(M)$ **by (12). If there is** u **such that** $\mu(M) = \rho(M) = f^+(u) - f^-(u)$, **then** u **and** $\mu(M)$ **are the optimal solution and optimal value respectively of the problem in the box** M **under consideration. Then, we can use** $\mu(M)$ **as an update for the** CBV.

Step 3. **Let** $\mathcal{S}_k = \mathcal{R}_k \cup \mathcal{P}'_k$. **Delete every** $M \in \mathcal{S}_k$ **such that** $\mu(M) > CBV$ **and let** $\mathcal{R}_{k+1}$ **be the collection of remaining boxes.**

Step 4. **If** $\mathcal{R}_{k+1} = \emptyset$ **then terminate.** CBV **is the optimal value and the feasible solution** $\tilde{u}$ **with** $f(\tilde{u}) = CBV$ **is the optimal solution.**

Step 5. **If** $\mathcal{R}_{k+1} \neq \emptyset$, **let** $M_k \in$ **arg** $\min \{\mu(M) | \ M \in \mathcal{R}_{k+1}\}$. **Divide** M_k **into two boxes according to the branching rule. Let** $\mathcal{P}_{k+1}$ **be the collection of these two sub-boxes of** M_k.

Step 6. **Increment** k **and return to Step 1.**

Theorem 1. *For our problem (4), BRB algorithm terminates after finitely many iterations, yielding an* **globally** *optimal solution of the problem.*

Remarks. A linear systematic (k, n) LDPC code can be represented as a generator matrix $G = [I_k \ P_{k \times (n-k)}]^T$ where I_k is the identity matrix of size k. Instead of starting the algorithm with the largest box $[0, 1]^k$ as described above, we can possibly detect some bit variables after examining the observed output $\tilde{y}$ by the following steps:

- For $i = 1 \ldots k$, let $\tilde{u}_i$ be such that

$$\tilde{u}_i = \begin{cases} 1 & \tilde{y}_i \geq 0 \\ 0 & \tilde{y}_i < 0. \end{cases}$$

- Let S be the set of indices such that $S = S_1 \bigcup S_2$ where $S_1 = \{i \mid \tilde{y}_i < 0, \ \cos(\pi g_i \tilde{u}) \neq 1, \ i = (k+1), \ldots, n\}$ and $S_2 = \{i \mid \tilde{y}_i \geq 0, \ \cos(\pi g_i \tilde{u}\} \neq -1, \ i = (k+1), \ldots, n\}$.
- Set $\tilde{G} = \{g_i\}$ whose rows are rows $\{g_i\}, i \in I$ of G.
- Set $\tilde{a}_i = \tilde{b}_i = \tilde{u}_i$ if column i^{th} of $\tilde{G}$ contains no 1's.

For LDPC codes, due to small fraction of 1's in G, more input bits are likely to be resolved in the final step. Therefore, the algorithm can initially start with the tigher box $[\tilde{a}, \tilde{b}]$.

4. EXPERIMENTAL RESULTS

This section provides numerical results obtained from running the BRB algorithm for different coding schemes, where $\mathbf{k} - \mathbf{n} \doteq (k, n)$ linear codes. The proposed algorithm is simulated in the C programming environment on a PC with CPU 3.0 Ghz. The generator matrix G is full rank and randomly generated with a small fraction of 1's. The two high rate codes are systematic. The Bit-Error-Rate curves for different codes are plotted against the E_b/N_o where E_b, N_o are bit energy and noise power spectral density, respectively. The average running time and the average number of iterative loops for each trial are also tabled. The BERs of random rate $1/4$ codes are similar to those presented in [1]. For codes with special properties, the average processing time and running loops do not increase noticeably when k increases due to the two reasons: (i) the convergence rate to the optimal point is faster when E_b/N_o is larger as the result of tigher bounding calculations. This leads to more useless boxes being

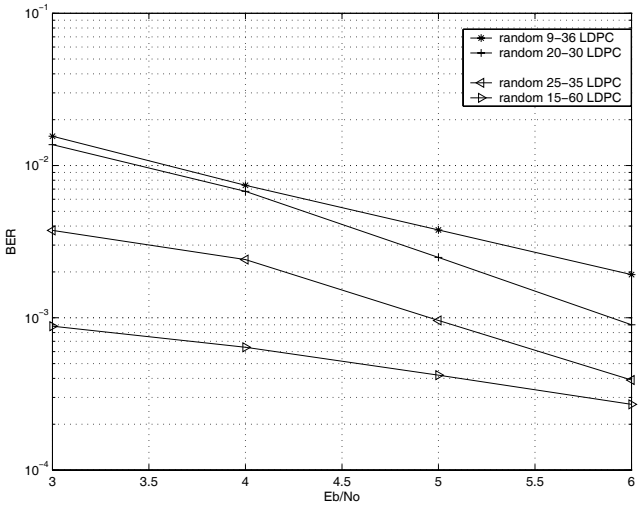

Fig. 1. Bit-Error-Rate performance of different codes

Codes \ Eb/No	3	4	5	6
9-36	20.3247	19.1866	18.2872	17.5104
15-60	434.6903	428.6563	399.597	397.1853
20-30	109.35729	70.97733	38.83573	20.95469
25-35	157.344	130.200	95.408	89.02946

Table 1. Average iterations for each trial

deleted in step 3 of the algorithm; (ii) by the structure of systematic codes, most bits can be determined as mentioned in **Remarks**. Therefore, it takes a significantly small number of loops for each trial.

5. CONCLUSION

We have described the method ML decoding for Gaussian channels using $0-1$ monotonic optimization. The technique can also be applied for other common channels i.e. binary symmetric channel, binary erasure channel etc but with different objective functions. The proposed algorithm complexity depends on the input sequence length k which is much less than the codeword length n, especially for a coding scheme with a small code rate.

Codes \ Eb/No	3	4	5	6
9-36	0.01157	0.01100	0.01064	0.01015
15-60	0.54700	0.53589	0.52536	0.51854
20-30	0.12514	0.08278	0.04582	0.02515
25-35	0.57943	0.41833	0.26591	0.15192

Table 2. Average running time (in second) for each trial

6. REFERENCES

[1] J. Feldman, M.J. Wainwright and D.R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol.51, pp. 954-972, Mar. 2005.

[2] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Log. Quart.* 3 (1956), pp. 95-110.

[3] R. Gallager , *Low Density Parity Check Codes* . MIT Press 1962.

[4] D.J.C. Mackay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory* , vol. 45, pp. 399-431, Mar. 1999.

[5] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 140-152, 1998.

[6] H. Tuy, *Convex Analysis and Global Optimization*. Kluwer Akademic, 1999.

[7] H. Tuy,"Monotonic optimization: problems and solution approaches," *SIAM Journal on Optimization*, vol. 11, no. 2, pp. 464-494, 2000.

[8] H. Tuy, M. Minoux and N. T. H. Phuong,"Discrete monotonic optimization with application to a discrete location problem," to appear in *SIAM Journal on Optimization*.

[9] A. L. Yuille, "CCCP algorithms to minimize the Bethe and Kikuchi energies: convergent alternatives to belief propagation,", *Neural Computation*, vol. 14, no. 7, pp. 1691-1722, 2002.