LETTER Special Section on Information Theory and Its Applications

Performance of a Decoding Algorithm for LDPC Codes Based on the Concave-Convex Procedure

Tomoharu SHIBUYA(*a) and Kohichi SAKANIWA(**b), Regular Members

SUMMARY In this letter, we show the effectiveness of a double-loop algorithm based on the concave-convex procedure (CCCP) in decoding linear codes. For this purpose, we numerically compare the error performance of CCCP-based decoding algorithm with that of a conventional iterative decoding algorithm based on belief propagation (BP). We also investigate computational complexity and its relation to the error performance.

key words: LDPC codes, iterative decoding, belief propagation, concave-convex procedure, Bethe free energy

1. Introduction

Let \( X_i \) (\( i = 1, 2, \ldots, n \)) be finite sets. For a given probability density function (pdf) \( p(x) \) of \( x := (x_1, x_2, \ldots, x_n) \in X_1 \times X_2 \times \cdots \times X_n \), its marginal distribution \( p_i(x_i) \) (\( i = 1, 2, \ldots, n \)) is calculated by

\[
p_i(x_i) = \sum_{x_1 \in X_1} \cdots \sum_{x_{i-1} \in X_{i-1}} \sum_{x_{i+1} \in X_{i+1}} \cdots \sum_{x_n \in X_n} p(x)
\]

and this operation is called marginalization. Taking all summations in Eq. (1), however, can be sometimes computationally intractable for large \( n \).

Belief propagation (BP) introduced by Pearl [1] is one of the most efficient algorithms to calculate \( p_i(x_i) \), and is employed in several applications. For example, bitwise maximum a posteriori probability (MAP) decoding of linear codes is performed by marginalizing a posteriori probability of a transmitted codeword. It is known [2] that an iterative decoding algorithm of low-density parity-check (LDPC) codes [3] based on BP, which approximates bitwise MAP decoding, exhibits extremely good performance. However, it is also known that if there exists a loop in the Bayesian network (BN), a special graph representing dependency relations between random variables of a given pdf, BP does not always converge, and even if it converges, the correctness of resulting \( p_i(x_i) \) is not guaranteed in BP. Therefore some new algorithm has been sought which approximate \( p_i(x_i) \) more precisely, even if BN of a given distribution contains loops.

Recently, Yedidia et al. [4] clarified that there exists one-to-one correspondence between fixed points of BP and stationary points of the Bethe free energy by using a variational technique in statistical physics. Afterwards, Yuille [5] introduced a convergent algorithm based on the concave-convex procedure (CCCP) to find a minimum of the Bethe free energy. Yuille also reported that for a distribution of two and three dimensional spin-glass models, CCCP-based algorithm gives more accurate approximation than that obtained by BP. So the CCCP-based decoding algorithm for LDPC codes is expected to exhibit better error performance than the conventional BP-based iterative decoding algorithm.

The aim of this letter is to examine the effectiveness of CCCP-based decoding of binary linear codes including LDPC codes. In other words, we examine whether acquiring a minimum of Bethe free energy really results in a better error performance. This letter is organized as follows. In Sect. 2, we review CCCP and describe CCCP-based decoding algorithm for linear codes. In Sect. 3, we numerically compare the error performance of CCCP-based and BP-based decoding algorithms. Concluding remarks are given in Sect. 4.

2. Decoding of Linear Codes by CCCP

2.1 Bitwise MAP Decoding

Denote by \( \mathbb{F}_2 = \{0,1\} \) the finite field of two elements and consider a binary linear code \( C \subset \mathbb{F}_2^n \) of length \( n \). Let \( H = (h_{ij}) \) be an \( m \times n \) parity check matrix for \( C \). For \( H \), we define

\[
S := \{(i,j) \mid h_{ij} = 1\}, \quad A_i := \{j \mid h_{ij} = 1\}, \quad B_j := \{i \mid h_{ij} = 1\}.
\]

When a codeword \( x = (x_1, x_2, \ldots, x_n) \in C \) is transmitted and \( y = (y_1, y_2, \ldots, y_n) \in \mathcal{Y}^n \) is received where \( \mathcal{Y} \) represents an alphabet of received symbols, the bitwise MAP decoder finds \( \hat{x} \in \mathbb{F}_2 \) (\( j = 1, 2, \ldots, n \)) which maximizes the conditional probability \( p(\hat{x}_j | y) \) as an estimate of \( x_j \).
Assume that the channel is memoryless and its transition probabilities are denoted by \( p(y_j|x_j) \) where the transmitted symbol \( x_j \in \mathbb{F}_2 \) and the received symbol \( y_j \in \mathcal{Y} \). We also assume that the transmission of \( x \in C \) is equiprobable, that is,

\[
p(x) = \begin{cases} 1/|C|, & \text{if } x \in C, \\ 0, & \text{if } x \notin C. \end{cases}
\]

Since \( x \in C \) if and only if \( \sum_{j \in A_i} x_j = 0 \) for all \( i = 1, 2, \ldots, m \), \( p(x) \) can be expressed as \( p(x) = \frac{1}{|C|} \prod_{i=1}^m \delta(\sum_{j \in A_i} x_j, 0) \) where

\[
\delta(a, b) := \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{if } a \neq b. \end{cases}
\]

Then by a simple calculation, \( p(x|y) \) is expressed as

\[
p(x|y) = \kappa_1' \prod_{i=1}^m \delta(\sum_{j \in A_i} x_j, 0) \prod_{j=1}^n p(y_j|x_j)
\]  

(2)

where \( \kappa_1 \) is a constant for normalization (\( \sum_{x \in \mathbb{F}_2} p(x|y) = 1 \)).

In order to apply the CCCP algorithm, associating with \( p(x|y) \) we define the following joint pdf \( p(x, \{u_{A_i}\}|y) \) by introducing segmented vector variables \( \{u_{A_i}\} \):

\[
p(x, \{u_{A_i}\}|y) := \kappa_1' \prod_{i=1}^m \delta(\sum_{j \in A_i} u_j', 0) \prod_{j=1}^n p(y_j|x_j)
\]  

(3)

where \( u_{A_i} := (u_j')_{j \in A_i}, (i = 1, 2, \ldots, m) \) and \( \kappa_1' \) is a constant for normalization. Then we have

\[
p(x, \{u_{A_i}\}|y) = \begin{cases} p(x|y), & \text{if } u_{A_i} = x_A, \text{ for } i = 1, 2, \ldots, m, \\ 0, & \text{elsewhere,} \end{cases}
\]

where \( x_A := (x_j)_{j \in A_i}, (i = 1, 2, \ldots, m) \) and therefore

\[
p(x|y) = \sum_{i=1}^m \sum_{u_{A_i} \in \mathbb{F}_2^{|A_i|}} p(x, \{u_{A_i}\}|y).
\]  

(4)

Now we define

\[
\begin{align*}
\psi_i(u_{A_i}) := & \delta(\sum_{j \in A_i} u_j', 0) \\
& \text{for } i = 1, 2, \ldots, m \text{ and } u_{A_i} \in \mathbb{F}_2^{|A_i|}, \\
\psi_j(x_j) := & p(y_j|x_j) \\
& \text{for } j = 1, 2, \ldots, n \text{ and } x_j \in \mathbb{F}_2, \\
\psi_{ij}(u_{A_i}, x_j) := & \delta(u_j', x_j) \psi_i(u_{A_i}) \psi_j(x_j) \\
& \text{for } (i, j) \in S \text{ and } (u_{A_i}, x_j) \in \mathbb{F}_2^{|A_i|} \times \mathbb{F}_2.
\end{align*}
\]

(5)

Then by noting that both \( \prod_{i=1}^m \prod_{j \in A_i} \psi_i(u_{A_i}) \) and \( \prod_{i=1}^m \prod_{j \in B_i} \psi_j(x_j) \) are equivalent to \( \prod_{(i,j) \in S} p(x, \{u_{A_i}\}|y) \) of Eq. (3) is written as

\[
p(x, \{u_{A_i}\}|y) = \kappa_1' \prod_{(i,j) \in S} \psi_{ij}(u_{A_i}, x_j)
\]

\[
\times \prod_{i=1}^m \psi_i(u_{A_i})^{-|A_i|-1} \prod_{j=1}^n \psi_j(x_j)^{-|B_j|-1}.
\]  

(6)

For a pdf of the form of Eq. (6), the CCCP approximates its marginal distribution and thereby approximately realizes bitwise MAP decoding through Eq. (4).

Now we introduce another joint distribution of \( (x, \{u_{A_i}\}) \), denoted by \( q(x, \{u_{A_i}\}) \), and consider Kulback-Leibler divergence for \( p = p(x, \{u_{A_i}\}|y) \) and \( q = q(x, \{u_{A_i}\}) \) defined by

\[
D(q||p) := \sum_{x, \{u_{A_i}\}} q(x, \{u_{A_i}\}) \ln \frac{q(x, \{u_{A_i}\})}{p(x, \{u_{A_i}\}|y)}.
\]

It is well known that \( D(q||p) \geq 0 \) and \( D(q||p) = 0 \) if and only if \( q = p \). Moreover, it is also known that if BN of \( p(x, \{u_{A_i}\}|y) \) is tree, by restricting \( q(x, \{u_{A_i}\}) \) to the form of

\[
q(x, \{u_{A_i}\}) = \prod_{(i,j) \in S} q_{ij}(u_{A_i}, x_j)
\]

\[
\times \prod_{i=1}^m q_i(u_{A_i})^{-|A_i|-1} \prod_{j=1}^n q_j(x_j)^{-|B_j|-1}
\]  

(7)

where \( q_{ij}(u_{A_i}, x_j) \) \( ((i, j) \in S) \) denotes a joint distribution of \( (u_{A_i}, x_j) \in \mathbb{F}_2^{|A_i|} \times \mathbb{F}_2 \), and \( q_i(u_{A_i}) \) \( (i = 1, 2, \ldots, m) \) and \( q_j(x_j) \) \( (j = 1, 2, \ldots, n) \) denote a pdf of \( u_{A_i} \in \mathbb{F}_2^{|A_i|} \) and \( x_j \in \mathbb{F}_2 \), respectively, \( D(q||p) \) is minimized when \( q_{ij}(u_{A_i}, x_j) = p(u_{A_i}, x_j|y) \), \( q_i(u_{A_i}) = p_i(u_{A_i}|y) \) and \( q_j(x_j) = p_j(x_j|y) \). Hence by substituting \( q \) given as Eq. (7) for \( D(q||p) \) and obtaining \( q_{ij}(u_{A_i}, x_j) \), \( q_i(u_{A_i}) \) and \( q_j(x_j) \) which minimize \( D(q||p) \) subject to conditions expressed as

\[
\begin{align*}
\sum_{u_{A_i} \in \mathbb{F}_2^{|A_i|}} q_{ij}(u_{A_i}, x_j) &= q_j(x_j), \\
\sum_{x_j \in \mathbb{F}_2} q_i(u_{A_i}) &= q_i(u_{A_i}), \\
\sum_{u_{A_i} \in \mathbb{F}_2^{|A_i|}} q_{ij}(u_{A_i}, x_j) &= q_i(u_{A_i}), \\
\sum_{x_j \in \mathbb{F}_2} q_j(x_j) &= 1,
\end{align*}
\]

(8)

it is expected that even if BN of \( p(x, \{u_{A_i}\}|y) \) contains loops, \( p_j(x_j|y) \) is approximated by \( q_j(x_j) \). This approximately realizes bitwise MAP decoding for binary linear codes.

For \( q \) given in Eq. (7), let \( F(q) := D(q||p) + \ln \kappa_1' \). By simple calculations, \( F(q) \) is expressed as

\[
F(q) = \sum_{(i,j) \in S} \sum_{u_{A_i}} q_{ij}(u_{A_i}, x_j) \ln \frac{q_{ij}(u_{A_i}, x_j)}{\psi_{ij}(u_{A_i}, x_j)}
\]

\[
- \sum_{i=1}^m (|A_i| - 1) \sum_{u_{A_i}} q_i(u_{A_i}) \ln \frac{q_i(u_{A_i})}{\psi_i(u_{A_i})}.
\]
\[-\frac{n}{\sum_{j=1}^{n}(|B_j| - 1) \sum_{x_j} q_j(x_j) \ln \frac{q_j(x_j)}{\psi_j(x_j)}} \quad (9)\]

\(F(q)\) is known as the Bethe free energy in statistical physics. For given \(p(x, \{u_{A_i}\}|y)\), the CCCP first decomposes \(F(q)\) into the sum of concave parts with respect to \(q_j(u_{A_i}, x_j)\) and convex parts with respect to \(q_j(u_{A_i})\) and \(q_j(x_j)\), as given in Eq. (9). Then, it updates \(q_j(u_{A_i}, x_j)\), \(q_j(u_{A_i})\) and \(q_j(x_j)\) in this decomposition so as to reduce \(F(q)\). The CCCP consists of an outer and an inner loop as shown below.

2.2 Outer Loop of CCCP

Let \(t\) denote the iteration index. It is shown [5, Theorem 4] that the update rules so called outer loop of CCCP given by

\[
\begin{align*}
\gamma_{ij}^{t+1}(u_{A_i}, x_j) & = \psi_j(u_{A_i}, x_j)e^{-1-\lambda_j(x_j)-\lambda_j(u_{A_i})} - \gamma_{ij}^t, \\
\delta_{ij}^{t+1}(u_{A_i}) & = e^{-1+|A_i|+\sum_{j \in A_i} \lambda_j(u_{A_i})} - \gamma_{ij}^t, \\
\delta_{ij}^{t+1}(x_j) & = \psi_j(x_j)e^{-1+|B_j|+\sum_{j \in A_i} \lambda_j(x_j)} \\
& \quad \times (\frac{q_j(x_j)}{\psi_j(x_j)})^{B_j},
\end{align*}
\]

make \(F(q')\) decrease monotonically, i.e., \(F(q') \geq F(q^{t+1})\), provided that

\[
\{\gamma_{ij} | (i, j) \in S\},
\{\lambda_j(x_j) | (i, j) \in S, x_j \in \mathbb{F}_2 \text{ with } \psi_j(x_j) \neq 0\},
\{\lambda_j(u_{A_i}) | (i, j) \in S, u_{A_i} \in \mathbb{F}_2^{|A_i|} \text{ with } \psi_j(u_{A_i}) \neq 0\}
\]

are so chosen that \(\gamma_{ij}^{t+1}(u_{A_i}, x_j)\), \(\delta_{ij}^{t+1}(u_{A_i})\) and \(\delta_{ij}^{t+1}(x_j)\) in Eq. (10) satisfy the constraints given in Eq. (8).

In the following argument, we roughly investigate the computational complexity of each iteration of outer loop through the numbers of \(q_j^{t+1}(u_{A_i})\) for each \(i = 1, 2, \ldots, m\) and \(\delta_{ij}^{t+1}(x_j)\) for each \(j = 1, 2, \ldots, n\) to be updated in each iteration of outer loop.

Update rules for \(\gamma_{ij}^{t+1}(u_{A_i}, x_j)\) and \(\delta_{ij}^{t+1}(x_j)\) are considered for \(u_{A_i}\) with \(\psi_j(u_{A_i}) \neq 0\) and \(x_j\) with \(\psi_j(x_j) \neq 0\), respectively. By definition of \(\psi_j(u_{A_i})\) given in Eq. (5), \(\psi_j(u_{A_i}) \neq 0\) is equivalent to \(\sum_{j \in A_i} u_j = 0\). Therefore there exist \(2^{|A_i|-1}\) such \(u_{A_i}'s\) in \(\mathbb{F}_2^{|A_i|}\). On the other hand, we see from the definition of \(\psi_j(x_j)\) given in Eq. (5) that \(\psi_j(x_j) \neq 0\) for at most two \(x_j\)'s. Hence the numbers of \(q_j^{t+1}(u_{A_i})\) for each \(i = 1, 2, \ldots, m\) and \(\delta_{ij}^{t+1}(x_j)\) for each \(j = 1, 2, \ldots, n\) to be updated in each iteration of outer loop are \(2^{|A_i|-1}\) and 2, respectively, which are summarized in Table 1. This implies that if the number of nonzero elements in each row of \(H\) is large, the computational complexity of each iteration of outer loop is not so small.

Finally, we note that in order to calculate marginal distributions \(q_j(u_{A_i})\) and \(q_j(x_j)\), it is not necessary to calculate \(q_j^{t+1}(u_{A_i}, x_j)\).

<table>
<thead>
<tr>
<th>Table 1</th>
<th>The numbers of updates to be executed in each iteration.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outer loop</td>
<td>Inner loop</td>
</tr>
<tr>
<td>(q_j^{t+1}(u_{A_i}))</td>
<td>(\gamma_{ij}^{t+1}(x_j))</td>
</tr>
<tr>
<td>(2^{</td>
<td>A_i</td>
</tr>
</tbody>
</table>

2.3 Inner Loop of CCCP

In order to execute outer loop of CCCP, we need to set up constants \(\{\gamma_{ij}\}, \{\lambda_{ij}(x_j)\}\) and \(\{\lambda_{ij}(x_j)\}\) at every iteration. It has been shown [5, Theorem 5] that these constants are obtained by the so called inner loop of CCCP given by

\[
\begin{align*}
\gamma_{ij}^{t+1} = -1 + \ln \left(\sum_{x_j \in \mathbb{F}_2} \psi_j(x_j)e^{-\lambda_j(x_j)} \times \sum_{u_{A_i} \in \mathcal{U}_i(x_j)} e^{-\lambda_j(u_{A_i})}\right),
\lambda_{ij}^{t+1}(u_{A_i}) = \frac{1}{2} \left(\ln \psi_j(x_j(u_{A_i})) - |A_i| + \ln \gamma_{ij}^{t+1}(u_{A_i})\right) - \frac{1}{2} |A_i|, \\
\lambda_{ij}^{t+1}(x_j) = \frac{1}{2} \left(- |B_j| \left(1+\ln \gamma_{ij}^{t+1}(x_j)\right) + \ln \left(\sum_{u_{A_i} \in \mathcal{U}_i(x_j)} e^{-\lambda_j(u_{A_i})}\right) - \gamma_{ij}^{t+1}\right),
\end{align*}
\]

where \(\tau\) denotes the iteration index of the inner loop. It is guaranteed that the inner loop of CCCP converges monotonically to a unique solution. In the update rules for \(\gamma_{ij}^{t+1}\) and \(\lambda_{ij}^{t+1}(x_j)\) given in Eq. (11), \(\mathcal{U}_i(x_j)\) denotes \(\{u_i \in \mathbb{F}_2^{|A_i|} | \psi_j(u_{A_i}, x_j) \neq 0\}\) for each \(x_j \in \mathbb{F}_2\). On the other hand, it can be verified from the definition of \(\psi_j(u_{A_i}, x_j)\) given in Eq. (5) that the cardinality of \(\{x_j \in \mathbb{F}_2 | \psi_j(u_{A_i}, x_j) \neq 0\}\) for each \(u_{A_i} \in \mathbb{F}_2^{|A_i|}\) with \(\psi_j(u_{A_i}) \neq 0\) is one. Hence in the update rule for \(\lambda_{ij}^{t+1}(u_{A_i})\) given in Eq. (11), we denote the unique element by \(x_j(u_{A_i})\).

In the remaining part of this subsection, we roughly investigate the computational complexity of each iteration of inner loop.

The numbers of \(\lambda_{ij}^{t+1}(u_{A_i})\) and \(\lambda_{ij}^{t+1}(x_j)\) for each \((i, j) \in S\) to be updated in each iteration of inner loop are obtained from a similar argument to the numbers of \(q_j^{t+1}(u_{A_i})\) and \(q_j^{t+1}(x_j)\), respectively, and are summarized in Table 1 together with the number of \(\gamma_{ij}^{t+1}\). Moreover, in the update rules of \(\gamma_{ij}^{t+1}\) and \(\lambda_{ij}^{t+1}(x_j)\) given in Eq. (11), summation for \(u_{A_i}\) involves \(|\mathcal{U}_i(x_j)| = 2^{|A_i|-2}\) terms. Hence if the number of nonzero elements in each row of \(H\) is large, the computational complexity of each iteration of inner loop also becomes large.
2.4 CCCP-Based Decoding Algorithm

On the basis of the preceding argument, whole algorithm of original CCCP is summarized as follows [5].

[CCCP]

1. Set initial values of \( q_i(u_{A_i}) \) and \( q_j(x_j) \) appropriately. Set \( t = 0 \).
2. Set initial values of \( \{ \gamma_{ij} \}, \{ \lambda_{ji}(u_{A_i}) \} \) and \( \{ \lambda_{ij}(x_j) \} \) appropriately. Iterate inner loop until it converges.
3. Execute the update rules of outer loop once. If \( F(q^t) \) converges, stop the algorithm. Otherwise let \( t = t + 1 \) and go back to the second step.

It is easily recognized that CCCP contains nested infinite loops, which are inconvenient for actual decoding. Hence we fix the number of iterations of inner loop and restrict the maximum number of iterations of outer loop. Moreover, we introduce another stopping rule for outer loop. Then we propose CCCP-based double-loop decoding algorithm, which approximates bitwise MAP decoding, as follows.

[CCCP-based decoding algorithm]

1. Set initial values: \[ q_i(u_{A_i}) = 1/2^{\vert A_i \vert - 1} \] and \[ q_j(x_j) = 1/2 \]. Set also the number of iterations of inner loop \( I_{in} \) and the maximum number of iterations of outer loop \( I_{out} \). Set \( t = 0 \).
2. Set initial values of \( \{ \gamma_{ij} \}, \{ \lambda_{ji}(u_{A_i}) \} \) and \( \{ \lambda_{ij}(x_j) \} \) to be 0\(^t\). Iterate inner loop \( I_{in} \) times.
3. Execute the update rules of outer loop once, and determine \( \hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \) by

\[
\hat{x}_j = \begin{cases} 
0, & \text{if } q_j(x_j = 0) \geq q_j(x_j = 1), \\
1, & \text{otherwise}.
\end{cases}
\]

If \( H\hat{x} = 0 \) or \( t+1 = I_{out} \) then let \( \hat{x} \) be the decoding result and stop. Otherwise let \( t = t+1 \) and go back to the second step.

Since the number of iterations of inner loop is fixed to \( I_{in} \), the algorithm may not reach a minimum of \( D(q\Vert p) \). Thus the error performance may degrade. The effect of this limitation on the error performance and the computational complexity will be investigated in Sect.3.3.

It is immediately obtained from Table 1 that the total number of update rules to be executed for \( q_i^{t+1}(u_{A_i}) \) \((i = 1, 2, \ldots, m)\) and \( q_j^{t+1}(x_j) \) \((j = 1, 2, \ldots, n)\) at each iteration of outer loop is \( \sum_{i=1}^{m} 2^{\vert A_i \vert - 1} \) and \( 2n \), respectively. On the other hand, since the inner loop is executed \( I_{in} \) times at each one iteration of outer loop, the number of update rules to be executed for \( \gamma_{ij}^{t+1}, \lambda_{ji}^{t+1}(u_{A_i}) \) and \( \lambda_{ij}^{t+1}(x_j) \) \((i, j) \in S\) in inner loop at each one iteration of outer loop is \( |S|I_{in} \), \( (\sum_{i=1}^{m} |A_i|2^{\vert A_i \vert - 1})I_{in} \), and \( 2|S|I_{in} \), respectively. These are summarized in Table 2.

Table 2: The total numbers of update rules to be executed in the outer and the inner loops of each iteration of one outer loop.

<table>
<thead>
<tr>
<th>Outer loop</th>
<th>Inner loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_i^{t+1}(u_{A_i}) )</td>
<td>( \gamma_{ij}^{t+1} )</td>
</tr>
<tr>
<td>( q_j^{t+1}(x_j) )</td>
<td>( \lambda_{ji}^{t+1}(u_{A_i}) )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{m} 2^{\vert A_i \vert - 1} )</td>
<td>( (\sum_{i=1}^{m}</td>
</tr>
<tr>
<td>( 2n )</td>
<td>( 2</td>
</tr>
</tbody>
</table>

For a fixed coding rate, \( m \) is proportional to code length \( n \). So when we consider LDPC codes whose each \([A_i] i = 1, 2, \ldots, m\) is constant with respect to \( n \), \(|S| = \sum_{i=1}^{m} |A_i|\) is proportional to \( n \). Hence, as shown in Table 2, the computational complexity of each iteration of outer loop of CCCP-based decoding algorithm is proportional to \( n \). That of the one iteration of the conventional BP-based iterative decoding algorithm [6] is also proportional to \( n \).

3. Experimental Results

In this section, we numerically compare the error performance of CCCP-based decoding algorithm with that of BP-based one over an AWGN channel. It is well known [2] that BP-based decoding algorithm exhibits extremely good performance for most of long LDPC codes because their BN’s contain few short loops. On the other hand, in spite of practical importance, BP-based decoding algorithm does not exhibit good performance for short LDPC codes even if they have good code parameters, because their BN’s usually contain many short loops. Therefore, it is worthwhile examining if the CCCP-based decoding algorithm exhibits better performance than BP-based one for relatively short codes. In the following simulations, we employ an algorithm arranged by Wadayama [6] as the BP-based iterative decoding algorithm.

3.1 Random LDPC Code

In Fig.1, we show the word and bit error probability of two decoding algorithms for a randomly generated (3,6)-regular LDPC code with length 486 and rate 0.504. The parity check matrix for the code is constructed by Gallager’s method [3]. The associated BN has loops of length four, which is the shortest possible loop. For CCP-based algorithm, we set \( I_{in} = 3 \) and \( I_{out} = 500 \) respectively (see Sect.3.3 for the selection of \( I_{in} = 3 \)). As for BP-based algorithm, the maximum number of iterations \( I_{BP} \) is set to be 100, since the error performances are not improved even if \( I_{out} \) and \( I_{BP} \) are increased.

We see from Fig.1 that the performance of CCCP-based algorithm is superior to that of BP-based one by

\[ \text{Since the convergent point is unique, it does not depend on the initial values.} \]
Fig. 1 Comparison of BP and CCCP based algorithms for a (3,6)-regular LDPC code of length 486.

Fig. 2 Comparison of CCCP and BP based algorithms for a cyclic code of length 127.

0.35 dB at word error probability $10^{-4}$ and by 0.4 dB at bit error probability $10^{-5}$.

3.2 Cyclic Code

We next consider a [127, 63, 8] cyclic code whose parity check matrix $H$ is constructed by the method shown in [7]. Then the associated BN has no loops of length four. Therefore BP-based algorithm for this $H$ exhibits much better performance than that for the parity check matrix determined by the parity check polynomial. We also set $I_{in} = 3$, $I_{out} = 500$ and $I_{BP} = 100$.

Figure 2 shows the word and bit error probability of two decoding algorithms for the cyclic code. We see from Fig. 2 that the performance of CCCP-based algorithm is superior to that of BP-based one by 0.25 dB at word error probability $10^{-4}$ and by 0.4 dB at bit error probability $10^{-5}$.

3.3 Effect of $I_{in}$

In this subsection, we numerically investigate how $I_{in}$ affects the error performance of the CCCP-based decoding algorithm and the total computational complexity.

For the cyclic code treated in Sect. 3.2 at $E_b/N_0 = 4.0$ dB, we show in Fig. 3 the bit error probability ($P_e$) for several $I_{in}$’s, together with the average number of iterations of outer loop until the algorithm terminates ($T$), and the total average number of executions of inner loop ($I_{in}T$). We see from Fig. 3 that the error performance is not improved for $I_{in} \geq 3$ while the total average number of executions of inner loop monotonically increases as $I_{in}$ increases.

By comparing Eqs. (10) and (11), we can roughly say that the computational complexity of each update of inner loop is larger than that of outer loop. Hence if we set $I_{in}$ large, computational complexity of inner loop dominates that of the whole algorithm. Therefore by adjusting $I_{in}$ appropriately, we can reduce the total number of calculation without degrading the error performance.

4. Conclusion

In this letter, we have compared the error performance of CCCP-based decoding algorithm with that of BP-based one for a randomly generated (3,6)-regular LDPC code and a cyclic code with relatively short code length. Numerical experiments have shown that CCCP-based decoding algorithm exhibits better performance than that of BP-based one.

Though the computational complexity of CCCP-based algorithm is proportional to the code length $n$, the same as that of BP-based algorithm [6], some update rules require computations proportional to $2^{|A_i|}$. Hence in order for CCCP-based decoding algorithm to become true alternative to BP-based one, further reduction of computational complexity is required, which should be a further study.

Acknowledgement

The authors are grateful to Prof. T. Sato of Tokyo Institute of Technology for his introducing CCCP algorithm to them. The authors also express their gratitude...
to Prof. K. Araki and Prof. O. Watanabe both of Tokyo Institute of Technology and Dr. T. Wadayama of Okayama Prefectural University for their useful comments and discussions.

References


