An Adaptive Decoding Algorithm of LDPC Codes over the Binary Erasure Channel

Gou HOSOYA*, Hideki YAGI†, Toshiyasu MATSUSHIMA‡, and Shigeichi HIRASAWA††

* Graduate School of Science and Engineering
Waseda University
Ookubo 3-4-1, Shinjuku-ku, Tokyo 169-8555, Japan
E-mail: hosoya@hirasa.mgmt.waseda.ac.jp

† School of Fundamental Science and Engineering
Waseda University
Ookubo 3-4-1, Shinjuku-ku, Tokyo 169-8555, Japan

‡ Media Network Center
Waseda University
Totsuka-Machi 1–104, Shinjuku-ku, Tokyo 169–8050, Japan

†† School of Creative Science and Engineering
Waseda University
Ookubo 3-4-1, Shinjuku-ku, Tokyo 169-8555, Japan

Abstract

Two decoding algorithms of LDPC codes over the binary erasure channel are presented. These algorithms continue the decoding procedure after the BP decoding algorithm fails. Since the proposed decoding algorithms also use the sparse structure of the parity-check matrix of LDPC codes, the increase of decoding complexity of these algorithms is slightly larger compared to that of the BP decoding algorithm. We show by simulation results that the performance of the proposed decoding algorithms is much superior to that of the BP decoding algorithm.

1. Introduction

The combination of the LDPC codes with the belief propagation (BP) decoding algorithm has high performance with low decoding complexity [1], [2]. It is well known that the BP decoding over the binary erasure channel (BEC) cannot decode whenever a subset of the erased bit positions contains a stopping set [3]. To overcome a decoding failure caused by a stopping set, two approaches have been studied by many researchers. The first approach is adding the redundant rows and columns for a given parity-check matrix of the code to improve the performance of the BP decoding. This approach is taken by K. Kasai et al. [6], S. Sankaranarayanan and B. Vasic [8], and N. Kobayashi et al. [9]. The second one is performing the additional procedure after the BP decoding algorithm fails in decoding. This approach is taken by H. Pishro-Nik and F. Fekri [4], the present authors [7], and B. N. Vellambi and F. Fekri [10]. The decoding algorithms in [4] and [10] guess the erased bits for some value to correct erased bits. The decoding algorithm in [7] needs to substitute the equation which requires only small increment of the decoding complexity.

The main difference of these two approaches is as follows: The first approach needs to perform the additions of redundant rows and columns for the parity-check matrix, so it needs the procedure of constructing the redundant parity-check matrix only one time before transmitting the codewords. The performance of the BP decoding algorithm by using the redundant parity-check matrix is better than by using the original one, but it is only effective for codes with short length. On the other hand, the second approach needs to perform additional decoding procedure, so this procedure is always needed for each received sequence. The performance of these two algorithms is significantly superior to that of the BP decoding algorithm for codes with various length.

In this paper, we propose two decoding algorithms of LDPC codes over the binary erasure channel (BEC) which do not need the guessing procedures. The proposed decoding algorithms are also an iterative one using the sparse structure of the parity-check matrix of LDPC codes. We show by simulation results that the proposed decoding algorithms can attain a smaller bit error rate than the BP decoding algorithm with a little increase of the decoding complexity.

This paper is organized as follows. In Section 2, we describe LDPC codes, decoding for the BEC, and the BP decoding algorithm. In Section 3, we describe the proposed decoding algorithms. We mention the related works of the proposed decoding algorithms in Section 3.1. An overview and procedure of the decoding algorithm A are presented in Section 3.2 and we give some correctable condition of the decoding algorithm A in Section 3.3. The decoding algorithm B is presented in Section 3.4. Finally, some simulation results and discussions are presented in Section 4 and concluding remarks are given in Section 5.
2. Preliminaries

2.1. LDPC Codes

Let $c = (c_1, c_2, \ldots, c_N) \in \{0,1\}^N$ be a codeword of LDPC codes and $H = [H_{mn}]$, $m \in [1,M]$, $n \in [1,N]$, $cH^T = 0$, be a parity-check matrix whose row and column lengths are $M$ and $N$, respectively. In this paper, we consider binary LDPC codes for simplicity of the discussion. Let $\lambda_i$ and $p_i$ denote the fraction of element ones in $H$ which are in columns and rows for weight $i$, respectively, and $\lambda(x) = \sum_{i=2}^\infty \lambda_i x^{i-1}$ and $\rho(x) = \sum_{i=2}^\infty p_i x^{i-1}$ be weight distributions of rows and columns of ones in $H$, respectively. LDPC codes are characterized by $\mathcal{C}(\lambda(x), \rho(x))$. The number of rows $M$ is given by $M = N \frac{\rho(x)}{\lambda(x)}$ and designed rate $R'$ is given by $R' = 1 - \frac{M}{N}$. The rate of the codes $R$ satisfies $R \leq R'$ since $H$ is not guaranteed to be a full rank matrix.

We define a loop of length $2L$, $L \geq 2$ in $H$ as a closed path constituting of the elements $1$ in $H$ at the positions $(m_1, n_1), (m_2, n_2), \ldots, (m_L, n_L)$, and $(m_L, n_1)$ where $m_1 \neq m_2 \neq \ldots \neq m_L$ and $n_1 \neq n_2 \neq \ldots \neq n_1$. For an example, the element ones at the positions $(m_1, n_1), (m_1, n_2), (m_2, n_2)$, and $(m_2, n_1)$ form a loop of length 4.

2.2. Decoding for the BEC

We assume a codeword $c$ is transmitted through the channel $(c_1, c_2, \ldots, c_N) \in \{0,1\}^N$ where $\epsilon$ denotes an erasure, and the decoder receives a sequence $y = c + e$. The addition of a binary bit and the erasure bit are defined as $0 + e = \epsilon$ and $1 + \epsilon = \epsilon$. Therefore, the received bits are either erased or known bits.

Let $N \subseteq [1,N]$ be an index set of the codeword bits or these of the columns in $H$. And let $E \subseteq N$ and $\bar{E} = N \setminus E$ be the index sets of the erased bits and the known bits, respectively. From the definition of a parity-check matrix $H$, we can write

$$cH^T = c_{E}H_{E}^T + c_{\bar{E}}H_{\bar{E}}^T = 0,$$

where $c_{E}$ is a subvector of $c$ whose elements are indexed by $E$, and $H_{E}^T$ is a submatrix whose column positions are indexed by $E$. Since $c_{E}H_{E}^T$ is known to a receiver and from Eq. (1),

$$c_{\bar{E}}H_{\bar{E}}^T = s_{E},$$

where $s_{E} = (s_{E}^{(1)}, s_{E}^{(2)}, \ldots, s_{E}^{(M)}) \in \{0,1\}^M$ is a syndrome sequence calculated by $c_{\bar{E}}H_{\bar{E}}^T$. Therefore, decoding for the BEC is to solve the erased (unknown) sequence $c_{E}$ from the simultaneous equations $c_{E}H_{E}^T = s_{E}$. Since $c$ is a codeword, $c_{E}$ has at least one solution. If $c_{E}$ has multiple solutions, then it cannot be corrected which causes to decoding failure.

2.3. BP Decoding Algorithm [2]

We define the following sets for all $(m,n)$ such that $H_{mn} = 1$.

$$\mathcal{A}(m) = \{n \mid H_{mn} = 1\}, \quad \mathcal{B}(n) = \{m \mid H_{mn} = 1\}.$$

Let $\mathcal{A}_E(m) = \{\mathcal{A}(m) \cap E\}, \quad m \in [1,M]$ be an index set of the erased bit positions at row $m$ in $H_E$. Therefore, we can rewrite $c_{E}H_{E}^T = s_{E}$ as

$$\sum_{i \in \mathcal{A}_E(m)} c_i = s_{E}^{m}, \quad \text{in} \quad [1,M].$$

Note that $c_i$, $i \in \mathcal{A}_E(m)$, are not known to the receiver and from Eq. (2), $s_{E}^{m}$ is obtained by calculating

$$s_{E}^{m} = \sum_{i \in \mathcal{A}(m) \setminus \mathcal{A}_E(m)} c_i.$$

From Eq. (4), BP decoding algorithm can correct the erased bit $c_i$, $i \in \mathcal{A}_E(m)$ if $|\mathcal{A}_E(m)| = 1$. The algorithm can continue the above procedure until all erased bits are corrected or there is no $m$ satisfying $|\mathcal{A}_E(m)| = 1$.

The BP decoding algorithm over the BEC is constituted by the following procedures:

**B1** For $m \in [1,M]$, set $\Psi_B(m) := \mathcal{A}_E(m)$ and $s_B(m) = s_{E}^{m} \quad \text{in} \quad \mathcal{B}(m) \setminus \mathcal{A}_E(m)$.

**B2** If there exists $m \in [1,M]$ satisfying $|\Psi_B(m)| = 1$, then go to **B3**. Otherwise the algorithm fails to stop.

**B3** For $m \in [1,M]$ such that $|\Psi_B(m)| = 1$, perform the followings.

**B3-1** Set $c_i := s_B(m) \quad \text{in} \quad \mathcal{B}(i) \setminus \mathcal{A}_E(m)$.

**B3-2** For $m' \in \mathcal{B}(i) \setminus \mathcal{A}_E(m)$, set $s_B(m') := s_B(m') + s_B(m)$.

**B4** If $\mathcal{E} \neq \emptyset$, then go to **B2**. Otherwise the algorithm successfully finishes to decode.

The BP decoding algorithm on the BEC fails when a subset of the erased bit positions contains a stopping set.

**Definition 1.** [Stopping set [3]] Choose some columns of $H$ to make a submatrix. A stopping set $\mathcal{S} \subseteq E$ is a subset of the erased bit positions such that the weights of all rows in the submatrix $H_{\mathcal{S}}$ of $H$, whose column positions are indexed by $\mathcal{S}$, are at least two.\footnote{Notice that we do not consider the rows of weight zero.}
In the procedure B2) at the BP decoding algorithm, the algorithm stops when there does not exist \( m \in [1, M] \) satisfying \( \Psi_B(m) = 1 \). At this time, all rows of the submatrix of \( H \) whose column positions are indexed by \( \mathcal{E}_B \), are at least two. Therefore \( \mathcal{E}_B \) contains a stopping set\(^3\) \( S \).

3. Proposed Decoding Algorithms

In this section, we propose two decoding algorithms of LDPC codes over the BEC which utilize the decoding procedure after the BP decoding algorithm fails.

3.1. Relation with Other Methods

To overcome a decoding failure caused by a stopping set, two approaches have been studied.

The first approach is adding the redundant rows and columns for a given parity-check matrix of the code to improve the performance of the BP decoding [6], [8], [9]. The second one is performing the additional procedure after the BP decoding algorithm fails [4], [7], [10].

The first approach needs to add the redundant rows or columns for the parity-check matrix before transmission, therefore it needs only once. The key idea is to make the small size of the stopping set be a large value by adding rows or columns for \( H \). When adding rows or columns, this method needs to investigate the loops in \( H \) as possible. And its computational complexity is large when we look for large length of loops. Therefore it usually take into account only the loops with short length and the performance by this method can be improved only for the codes only with short length.

On the other hand, the second approach needs to perform additional decoding procedure, therefore this procedures are always needed for each received sequence. The decoding performance of these improved BP decoding algorithms are significantly better than that of the ordinary BP decoding algorithm for codes with various length. The decoding algorithms by Pishro-Nik and Fekri [4] and Vellambi and Fekri [10] guess the values of erased bits (0 or 1) which are not corrected by the BP decoding algorithm. Clearly the performance of these algorithms depends on the number of guessed bits and the way of choosing these bits.

In this section, we propose two decoding algorithms which do not need the guessing procedures.

3.2. Decoding Algorithm A

The decoding algorithm A continues the decoding procedure after the BP decoding algorithm fails.

\(^3\)Note that at this time, \( \mathcal{E}_B \) equals to the set of erased bit positions which cannot be corrected by the BP decoding algorithm.

Let \( c_{\xi_a} \) is a subvector of \( c \) whose elements are indexed by \( \mathcal{E}_B \), and \( H_{\xi_a} \) is a submatrix whose column positions are indexed by \( \mathcal{E}_B \). Let \( s^p = c_{\xi_a}H_{\xi_a}^T \) where \( s^p = (s^p_1, s^p_2, \ldots, s^p_M) \in \{0, 1\}^M \). The decoding algorithm A tries to solve a simultaneous equation \( c_{\xi_a}H_{\xi_a}^T = s^p \). From the Definition 1, weights of all rows in \( H_{\xi_a} \) are at least two since \( \mathcal{E}_B \) contains a stopping set \( S \). Let \( \mathcal{A}_P(m) = \{A(m) \cap \mathcal{E}_B\}, m \in [1, M]\), be an index set of erased bit positions at row \( m \) in \( H_{\xi_a} \). The inequality \( |\mathcal{A}_P(m)| \geq 2 \) always holds from the Definition 1.

At first, we choose row \( m \) satisfying \( |\mathcal{A}_P(m)| = 2 \). We here assume that \( \mathcal{A}_P(m) = \{i_1, i_2\} \) where \( B(i_1) \geq B(i_2) \). Notice that choosing either \( i_1 \) or \( i_2 \) does not influence on the decoding result. In a view-point of the simultaneous equation \( c_{\xi_a}H_{\xi_a}^T = s^p \), the equaiton can be written as follows:

\[
c_{i_2} = c_{i_1} + s^p_m.
\]

Next, we substitute the above equation to the other equations that have element at \( i_2 \). The equation that we used to substitute will never be used in a substitution procedure. This substitution procedure sometimes makes an erased bit to a known bit. The procedure continues until all erased bits are corrected or the number of elements in all the equations that are not used in substitution procedures, are at least three.

The decoding algorithm A is constituted by the following procedures after the BP decoding algorithm fails:

[Decoding Algorithm A]

**P1** For any \( m \in [1, M] \), set \( \Psi_P(m) := \mathcal{A}_P(m) \) and \( s_P(m) := s^p_m \). For \( n \in \mathcal{E}_B \), set \( \Delta_P(n) := B(n) \). Set \( \mathcal{E}_P := \mathcal{E}_B, \mathcal{M}_P := \{m \mid m = [1, M], |\Psi_P(m)| \geq 2\} \).

**P2** If there does not exist \( m \in \mathcal{M}_P \) satisfying \( |\Psi_P(m)| = 2 \) or \( |\Psi_P(m)| = 1 \), then the algorithm fails. If there exists \( m \in \mathcal{M}_P \) satisfying \( |\Psi_P(m)| = 1 \), then go to **P4**. Otherwise (If there exists \( m \in \mathcal{M}_P \) satisfying \( |\Psi_P(m)| = 2 \), go to **P3**.

**P3** For \( m \in \mathcal{M}_P \) satisfying \( |\Psi_P(m)| = 2 \), perform the followings:

**P3-1** For \( m' \in \Delta_P(j) \) where \( \Psi_P(m) = \{i, j\} \), \( |\Delta_P(i)| \geq |\Delta_P(j)| \), set

\[
\Psi_P(m') := \{\Psi_P(m') \cup i\},
\]

\[
s_P(m') := s_P(m') + s_P(m).
\]

**P3-2** Set \( \Delta_P(i) := \{\Delta_P(i) \backslash m\} \cup \{\Delta_P(j) \backslash m\} \) and set \( \mathcal{M}_P := \mathcal{M}_P \backslash m \). If \( \Delta_P(i) \) has the same elements, then remove all of them from \( \Delta_P(i) \).

**P4** For \( m \in [1, M] \) such that \( |\Psi_P(m)| = 1 \), perform the followings.
Theorem 1. Assume that a loop of length $2L$, $L \geq 2$, is contained in $H_S$. Let the positions of this loop be $(m_1, n_1), (m_1, n_2), (m_2, n_2), \ldots, (m_L, n_L)$, and $(m_L, n_1)$. We set $M_L = \{m_1, m_2, \ldots, m_L\}$ and $N_L = \{n_1, n_2, \ldots, n_L\}$. The decoding algorithm $A$ can correct an erased bit $n^*$ if there exist only one $m' \in M_L$ such that $|\Psi_P(m')| = 3$, $|\Psi_P(m') \setminus n^*| \subseteq N_L$, and $n^* \in \Psi_P(m')$ and other $m \in M_L \setminus m'$ satisfy $|\Psi_P(m)| = 2$, $m \in M_L$, and $\Psi_P(m) \subseteq N_L$.

Proof. We assume that $m' = m_L$. Therefore $|\Psi_P(m_L)| = 3$, $|\Psi_P(m_1)| = 2$, $m \in M_L \setminus m_L$, and $\Psi_P(m_L) \subseteq \{n_1, n_L, n^*\}$ hold. An example of this is valid even if $m'$ is other value chosen from $M_L$.

We substitute equation $m_1$ to the other equations having element at position $j = n_2$ and $\Psi_P(m_2)$ is rewritten from $\{n_2, n_3\}$ to $\{n_1, n_3\}$. We can see this result in Fig. 1 (b). We substitute equations $m_2, m_3, \ldots, m_{L-1}$ in order and obtain $\Psi_P(m_L') = n^*$. Therefore we can calculate an erased bit at position $n^*$ in $m$. We can see the result before substituting $m_L$ in equation $m_L$. An example of this situation can be seen in Fig. 1 (c) where this is the case of $L = 2$. The cases of $L \geq 3$ are obviously proven.

3.4. Decoding Algorithm B

The decoding algorithm B continues the decoding procedure after the algorithm A fails. At this time, we set $\Psi_P(m) := \Psi_Q(m)$ and $M_L := M_P$. The matrix $H_Q$ whose row index sets are $\Psi_P(m)$, $m \in M_P$, contains many short loops in common. The size of the matrix $H_Q$ is $|M_P| \times \sum_{m \in M_P} |\Psi_P(m)|$.

First the algorithm investigates the loops of short length in the matrix $H_Q$, then the row index sets for each loop are obtained. Next the set of row vectors $H_Q$, are produced by linearly combing the row vectors of $H_Q$ indexed by the above row index sets. These rows are concatenated to $H_Q$ and we obtain a new matrix $H^1_Q$, such that

$$H^1_Q = \left[ \frac{H_Q}{H_{Q1}} \right].$$

If $H^1_Q$ has a row of weight two, then we go back to the decoding algorithm A for $H^1_Q$. If the decoding algorithm A stops in failure, we again proceed the same procedure. These procedures continue until all the erased bits are corrected. If the algorithm cannot produce a new row vector or the number of iterations of the
above procedures reaches pre-determined value, then the algorithm fails.

4. Simulation Results

4.1. Conditions for Simulation

In order to show the performance of the proposed decoding algorithms, we show the simulation results. We construct codes $C_1$ and $C_2$ which are denoted by $C_1(N_1, \lambda_1(x), \rho_1(x))$ and $C_2(N_2, \lambda_2(x), \rho_2(x))$ such that

$$N_1 = 1000, \quad \lambda_1(x) = x^2, \quad \rho_1(x) = x^5, \quad (6)$$

$$N_2 = 1000, \quad \lambda_2(x) = 0.0769x + 0.6923x^2 + 0.2308x^5, \quad \rho_2(x) = 0.46135x^5 + 0.53865x^6. \quad (7)$$

The designed rate of these codes are one half.

We compare the BP decoding algorithm [2] (denoted by “BP”), the decoding algorithm A (denoted by “PropA”), and the decoding algorithm B (denoted by “PropB”). For each decoding algorithm, we transmit at least $10^5$ codewords over the BEC with channel erasure probability $p$ until 50 codewords are failed in decoding. In “PropB” we repeat the procedure of investigating the loops of length 4 twice for each received sequence.

We evaluate them by (i) decoding performance as bit erasure rate (BER) and (ii) decoding complexity as the number of Exclusive OR operations needed for decoding.

4.2. Decoding Results and Discussions

4.2.1. Decoding Performance

Figs. 2 and 3 show the decoding performance for the Code $C_1$ and the $C_2$, respectively. The horizontal axis and the vertical axis represent the erasure probability of the BEC and BER, respectively.

From Figures. 2 and 3, the performance of both proposed decoding algorithms are superior to that of the “BP”. In Fig. 3 at $p = 0.36$, the BER of the “PropA” is 100 times smaller than that of the “BP” and that of the “PropB” is 1000 times smaller than that of the “BP”, respectively.

We confirm that there is no significant difference of behavior between BER and the word erasure rate.

4.2.2. Decoding Complexity

The “PropB” can be divided into the following parts of the procedures.

(A) The procedure of computing $s_E$ by $c_E H_F^T$

(B) The procedure of the “BP”

(C) The procedure of the “PropA” after the “BP” fails

(D) The procedure of the “PropB” after the “PropA” fails except finding the loops of length 4

(E) The procedure of finding the loops in the “PropB”

Clearly the combination of the procedures (A) and (B) equals to the “BP” and the combination of the procedures (A) $\sim$ (C) equals to the “PropA”.

Tables 1 and 3 show the average number of decoding operations of decoding procedures (A) $\sim$ (E) for the Code $C_1$ and the Code $C_2$, respectively. Tables 2 and 4 show the average number of Exclusive OR operations needed for decoding algorithms of the Code $C_1$ and the Code $C_2$, respectively. Notice that we here average the number of computational operations of decoding algorithms over all the transmitted sequences.

![Figure 2: Decoding result of the Code $C_1$](image2.png)

![Figure 3: Decoding result of the Code $B$](image3.png)
Table 1: The number of Exclusive OR operations of each procedure for the Code $C_1$

<table>
<thead>
<tr>
<th>$p$</th>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(D)</th>
<th>(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.35</td>
<td>700</td>
<td>700</td>
<td>948.7</td>
<td>1312.3</td>
<td>3675.7</td>
</tr>
<tr>
<td>0.36</td>
<td>719.9</td>
<td>719.9</td>
<td>975</td>
<td>1281.5</td>
<td>3616.8</td>
</tr>
<tr>
<td>0.38</td>
<td>757</td>
<td>757</td>
<td>1010.8</td>
<td>1187.2</td>
<td>3661.4</td>
</tr>
<tr>
<td>0.4</td>
<td>755.5</td>
<td>755.2</td>
<td>1003.9</td>
<td>1006.1</td>
<td>3725.9</td>
</tr>
<tr>
<td>0.42</td>
<td>643.4</td>
<td>641</td>
<td>980.3</td>
<td>799.4</td>
<td>3698.1</td>
</tr>
<tr>
<td>0.44</td>
<td>420.9</td>
<td>412.1</td>
<td>828.5</td>
<td>540.8</td>
<td>3652.2</td>
</tr>
</tbody>
</table>

Table 2: The average number of Exclusive OR operations needed by both decoding algorithms for the Code $C_1$

<table>
<thead>
<tr>
<th>$p$</th>
<th>BP</th>
<th>PropA</th>
<th>PropB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.35</td>
<td>1399.9</td>
<td>1400</td>
<td>1400</td>
</tr>
<tr>
<td>0.36</td>
<td>1439.8</td>
<td>1440</td>
<td>1440</td>
</tr>
<tr>
<td>0.38</td>
<td>1514</td>
<td>1520.8</td>
<td>1525.4</td>
</tr>
<tr>
<td>0.4</td>
<td>1510.7</td>
<td>1604.4</td>
<td>1737.2</td>
</tr>
<tr>
<td>0.42</td>
<td>1284.4</td>
<td>1647.7</td>
<td>2535.2</td>
</tr>
<tr>
<td>0.44</td>
<td>833</td>
<td>1477.4</td>
<td>4145.7</td>
</tr>
</tbody>
</table>

From these tables, “PropA” and “PropB” need slightly more operations than “BP”. Both of these algorithms need much more operations than “BP” as $p$ takes large value. From Tables 1 and 3, the procedure (E) dominates much times in the “PropB”. In procedure (E), we only investigate the loops of length 4. The computational complexity of the procedure (E) is $O(d_{max}c_{max})$ where $d_{max}$ and $c_{max}$ represent the maximum weights of rows and columns, respectively if we look for the loops of length 4. Therefore the computational complexity of the “PropB” grows large if we look for loops with long length.

5. Concluding Remarks

We have proposed new iterative decoding algorithms of LDPC codes over the BEC. From simulation results, BER of the proposed decoding algorithms are much lower than that of the BP decoding algorithm. They have a favorable trade-off between BER and complexity when the channel erasure probability is a small value.

References


Table 3: The number of Exclusive OR operations of each procedure for the Code $C_2$

<table>
<thead>
<tr>
<th>$p$</th>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(D)</th>
<th>(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.36</td>
<td>810</td>
<td>810</td>
<td>1014.5</td>
<td>2139.4</td>
<td>7109.1</td>
</tr>
<tr>
<td>0.38</td>
<td>854.1</td>
<td>854.1</td>
<td>1071.5</td>
<td>1978.4</td>
<td>6630.1</td>
</tr>
<tr>
<td>0.4</td>
<td>881.4</td>
<td>881.4</td>
<td>1095.4</td>
<td>1725.8</td>
<td>6224.7</td>
</tr>
<tr>
<td>0.42</td>
<td>809.1</td>
<td>808.3</td>
<td>1059.4</td>
<td>1443.3</td>
<td>6363.9</td>
</tr>
<tr>
<td>0.44</td>
<td>610.7</td>
<td>606</td>
<td>981.5</td>
<td>1268.7</td>
<td>6901.1</td>
</tr>
</tbody>
</table>

Table 4: The average number of Exclusive OR operations needed by both decoding algorithms for the Code $C_2$

<table>
<thead>
<tr>
<th>$p$</th>
<th>BP</th>
<th>PropA</th>
<th>PropB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.36</td>
<td>1619.9</td>
<td>1620</td>
<td>1620</td>
</tr>
<tr>
<td>0.38</td>
<td>1708.1</td>
<td>1710</td>
<td>1711.1</td>
</tr>
<tr>
<td>0.4</td>
<td>1762.8</td>
<td>1799.8</td>
<td>1845.1</td>
</tr>
<tr>
<td>0.42</td>
<td>1617.5</td>
<td>1855.5</td>
<td>2526.5</td>
</tr>
<tr>
<td>0.44</td>
<td>1216.7</td>
<td>1782</td>
<td>4590.3</td>
</tr>
</tbody>
</table>


