The 27th Symposium on Information Theory
and Its Applications (SITA2004)
Gero, Gifu, Japan, Dec. 14–17, 2004

A Decoding Method of Low-Density Parity-Check Codes Over the Binary Erasure Channel

Gou HOSOYA * Toshiyasu MATSUSHIMA * and Shigeichi HIRASAWA *

Abstract— We propose a new decoding algorithm of LDPC codes over the binary erasure channel (BEC). The proposed decoding algorithm is an iterative one which uses the sparseness structure of the parity-check matrix of LDPC codes. We show by simulation results that the proposed decoding algorithm and the conventional decoding algorithm have a trade-off between remaining erasure rate and complexity.

Keywords— low-density parity-check (LDPC) code, binary erasure channel (BEC), iterative decoding algorithm

1 Introduction

In recent years, *Low-density parity-check* (LDPC) codes [3, 4, 8] have been widely studied. LDPC codes with the *iterative message-passing algorithm* based on belief propagation (BP) can be decoded in high performance with low complexity [3, 3, 6, 7, 8]. The decoding performance of LDPC codes by the iterative decoding algorithm based on BP for the binary symmetric channel (BSC), and the binary input additive white Gaussian noise channel (BIAWGNC) have been studied so much assuming the code length is infinite using the density-evolution analysis [1, 6, 7].

As for the case of the iterative decoding algorithm of LDPC codes with finite code has not been studied so much, because the existence of loops in codes which degrades the decoding performance and makes difficult to analyze. However, when a channel is the binary erasure channel (BEC), analyzing the performance of iterative decoding algorithm of LDPC codes with finite code length has been studied using the *stopping sets weight distribution* [5].

Maximum likelihood decoding (MLD) over the BEC is utilized by the Gaussian elimination (GE) for arbitrary linear block codes of length N including LDPC codes, but the decoding complexity is $O(N^3)$ and is not practical. M. Luby, et al. have proposed the iterative decoding algorithm of LDPC codes over the BEC [1]. The symbol erasure rate (SER) of this decoding algorithm can be evaluated asymptotically by the recursive calculation method [1] that is similar way to the Gallager's hard decision iterative decoding algorithm over the BSC [3, 4]. The SER of this decoding algorithm for LDPC codes over the BEC are very low, but higher than MLD, and the decoding complexity is $O(N)$, which is efficient. However, this algorithm has often stopped when erasures have occurred at *stopping sets* [5].

In this paper, we propose a new decoding algorithm of LDPC codes over the binary erasure channel (BEC). The proposed decoding algorithm is an iterative one which uses the sparseness structure of the parity-check matrix of LDPC codes. We show by simulation results that the proposed decoding algorithm can attain smaller SER than the conventional decoding algorithm, although the former requires slightly larger complexity than the latter for erasure probability larger than 0.3, have a favorable trade-off between remaining erasure rate and complexity.

2 Preliminaries

2.1 LDPC Codes

Let $H = [H_{mn}]$, $m \in [1, M]$, $n \in [1, N]$, be a parity-check matrix whose row and column lengths are M and N, respectively, and $\boldsymbol{c} = (c_1, c_2, \ldots, c_N) \in \{0,1\}^N$ be a codeword of the LDPC code such that $\boldsymbol{c}H^{\mathrm{T}} = \boldsymbol{0}$. Let λ_i and ρ_i denote the fraction of ones of H which are in columns and rows for weight i, respectively, and $\lambda(x) \triangleq \sum_{i=2}^{\infty} \lambda_i x^{i-1}$ and $\rho(x) \triangleq \sum_{i=2}^{\infty} \rho_i x^{i-1}$ be a *degree distribution* of row and column of ones in H, respectively. LDPC codes are characterized by the code length N and the *degree distribution pair* (DDP) $\big(\lambda(x), \rho(x)\big)$ which are denoted by $\mathcal{C}\big(N, \lambda(x), \rho(x)\big)$, and the row length M is given by $M = N\frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$.

A parity-check matrix is represented by the *bipartite graph*, which consists of two types of nodes called *check nodes* indexed by position of rows, and *symbol nodes*

*Department of Industrial and Management Systems Engineering, School of Science and Engineering, Waseda University, Okubo 3–4–1, Shinjuku-ku, Tokyo, 169–8555 Japan. E-mail: `hosoya@hirasa.mgmt.waseda.ac.jp`

indexed by position of columns. A check node m and a symbol node n are connected with an edge if and only if $H_{mn} = 1$. A loop in the bipartite graph is a closed path that starts from a symbol node and returns to the same symbol node through edges without passing the same edges more than once. A length of a loop is a number of edges of the closed path.

2.2 MLD Algorithm Over the BEC

We assume a codeword $\boldsymbol{c}$ is transmitted through the BEC and receives a sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_N) \in \{0, 1, \xi\}^N$, where ξ denotes an erased symbol. Let ϵ be an erasure probability of the BEC such that $\Pr(y_n = \xi \mid c_n = a) = \epsilon$, $\Pr(y_n = a \mid c_n = a) = 1 - \epsilon$, $a \in \{0, 1\}$.

Let $\mathcal{N} \in \{1, N\}$ be the index set of symbols. And let $\mathcal{E} \in \mathcal{N}$ and $\bar{\mathcal{E}} \in \mathcal{N} \setminus \mathcal{E}$ be the index set of erased symbols and known symbols, respectively. From the definition of parity-check matrix, we can write

$$\boldsymbol{c}H^{\mathrm{T}} = \boldsymbol{c}_{\mathcal{E}}H_{\mathcal{E}}^{\mathrm{T}} + \boldsymbol{c}_{\bar{\mathcal{E}}}H_{\bar{\mathcal{E}}}^{\mathrm{T}} = \mathbf{0} \tag{1}$$

where $\boldsymbol{c}_{\mathcal{E}}$ and $H_{\mathcal{E}}$ are the subvector and submatrix of $\boldsymbol{c}$ and H which consist of those columns indexed by $\mathcal{E}$, respectively. Since $\boldsymbol{c}_{\bar{\mathcal{E}}}H_{\bar{\mathcal{E}}}^{\mathrm{T}}$ is known to a receiver and from eq.(1),

$$\begin{aligned} \boldsymbol{c}_{\mathcal{E}}H_{\mathcal{E}}^{\mathrm{T}} &= \boldsymbol{c}_{\bar{\mathcal{E}}}H_{\bar{\mathcal{E}}}^{\mathrm{T}} \\ &= \boldsymbol{s}', \end{aligned} \tag{2}$$

where $\boldsymbol{s}' = (s'_1, s'_2, \ldots, s'_M) \in \{0, 1\}^M$ is calculated by $\boldsymbol{c}_{\bar{\mathcal{E}}}H_{\bar{\mathcal{E}}}^{\mathrm{T}}$. Therefore, MLD algorithm is solved the erased (unknown) sequence $\boldsymbol{c}_{\mathcal{E}}$ from the simultaneous equations $\boldsymbol{c}_{\bar{\mathcal{E}}}H_{\bar{\mathcal{E}}}^{\mathrm{T}} = \boldsymbol{s}'$ by using the GE. Since $\boldsymbol{c}$ is a codeword, $\boldsymbol{c}_{\mathcal{E}}$ has at least one solution and MLD can correct a received sequence iff $\boldsymbol{c}_{\mathcal{E}}$ has a unique solution. If $\boldsymbol{c}_{\mathcal{E}}$ has multiple solutions, then it cannot be corrected[1], which causes decoding failure.

3 Conventional Decoding Algorithm [1]

The conventional iterative decoding algorithm can correct an erased symbol whenever its neighboring check node has only one erased symbol. We assume that the check node is labeled as 'satisfied' when its all neighboring symbol nodes are known, and otherwise it is 'unsatisfied'. The algorithm is following procedure:

[Conventional iterative decoding algorithm]

For all unsatisfied check nodes, do the following:

[1]This is a detected error. The receiver can know that it cannot be corrected.

C1) If the values of all but one of the symbol nodes connected to the check nodes are known, set the erased symbol to the XOR operation of the other symbol nodes and label that check node as 'finished'. If all the symbol nodes connected to the check node are known, label the check node as 'finished'. This procedure is done sequentially.

C2) Continue C1) until all check nodes are labeled as finished or decoding cannot continue further. □

4 Proposed Decoding Algorithm

4.1 Problems of the Conventional Decoding Algorithms

The conventional iterative decoding algorithm cannot decode whenever there are no unlabeled check nodes that have one erased symbol in its neighboring symbol nodes.

On the other hand, MLD which is utilized by the GE is the optimal, for arbitrary linear block codes of length N including LDPC codes, but the decoding complexity is $O(N^3)$ and is not practical. Moreover, when GE is applied to LDPC codes, sparseness of its parity-check matrix which can be decoded with low complexity by the iterative decoding algorithm, is lost by the row operation of GE. Generally, the resulting check node which is obtained by adding two check nodes (equations), has more symbols than the previous check node only when they have the same symbols.

4.2 Proposed Decoding Method

Consider the cases when the conventional decoding algorithm stops, there are two cases below: i) all check nodes are labeled as finished, ii) all unsatisfied check nodes have equal or greater than two erased symbols. The proposed decoding algorithm continues the decoding procedure after the case ii). At first, we choose unsatisfied check node that has two erased symbols whose positions are denoted by $\epsilon_1, \epsilon_2 \in \mathcal{E}$. Next, we substitute these check node to the other unsatisfied check nodes that have erased symbol in position ϵ_2 (or ϵ_1). We assume that the check node, that have two erased symbols and is substituted, is labeled as 'substituted'. This substituted procedure does not make erased symbols of resulting check node increased. Moreover, sometimes it can reduce erased symbols of the resulting check node. This procedure is continued until all erased symbols are corrected or there are no unsatisfied check nodes that have erased symbols greater than two and all unsatisfied check nodes that have two erased symbols are labeled as 'substituted'. The proposed decoding algorithm is following procedures:

[**Proposed decoding algorithm**]

For all unsatisfied and substituted check nodes, do the following:

P1) Perform C1) and C2).

P2) If P1) cannot continue further, then label the unsatisfied check node that have two erased symbols at positions ϵ_1, ϵ_2 as 'substituted', and substitute that check node to other check nodes that have erased symbol at position ϵ_2 (or ϵ_1). This procedure is done sequentially.

P3) If there is an unsatisfied check node that has one erased symbol, label that check node as 'finished', and this erased symbol is corrected. Next, substitute this corrected symbol to the other 'substituted' and 'unsatisfied' check nodes. This procedure is done sequentially.

P4) Continue P2) and P3) until all check nodes are labeled as 'finished' or all unsatisfied check nodes that have erased symbols greater than two and all unsatisfied check nodes that have two erased symbols are labeled as 'substituted'. □

5 Simulation Results

5.1 Conditions for Simulation

We construct codes $\mathcal{C}_1$ and $\mathcal{C}_2$ which are denoted by $\mathcal{C}_1(N_1, \lambda_1(x), \rho_1(x))$, $\mathcal{C}_2(N_2, \lambda_2(x), \rho_2(x))$ such that

$$N_1 = 500,\ \lambda_1(x) = x^2,\ \rho_1(x) = x^5, \quad (3)$$

$$N_2 = 1000,\ \lambda_2(x) = x^2,\ \rho_2(x) = x^5. \quad (4)$$

We compare the conventional iterative decoding algorithm [1] (denoted by "Conv.") with the proposed decoding algorithm (denoted by "Prop."). For each decoding algorithm, at least 3×10^{10} symbols are transmitted until 300 decoding erasures occur.

We evaluate them by (i) decoding performance (SER) and (ii) decoding complexity (the number of binary operations).

5.2 Simulation Results and Discussions

5.2.1 Decoding Results

Figs. 1 and 2 show SER of both decoding algorithms for the code. From these figures, SER of "Prop."

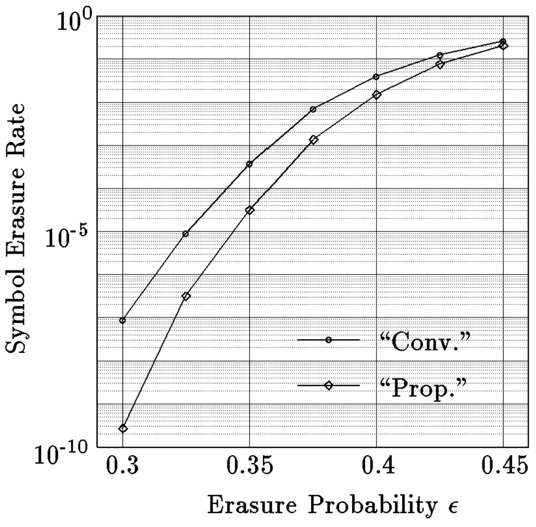

Figure 1: Simulation result of $\mathcal{C}_1$

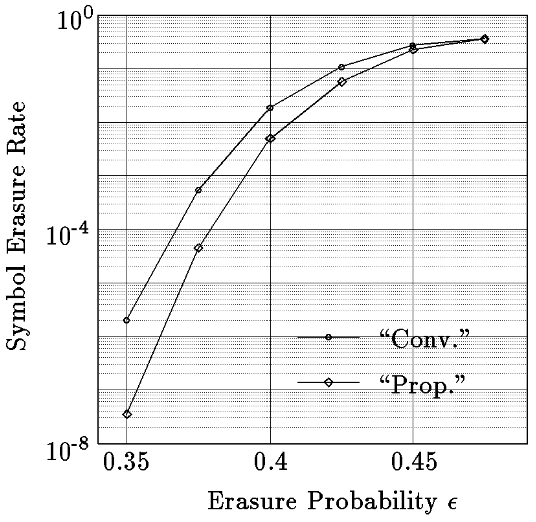

Figure 2: Simulation result of $\mathcal{C}_2$

is lower than that of "Conv.". This is because "Prop." continues decoding after "Conv." fails to decode and can decode much more erased symbols without decoding to a wrong codeword. "Prop." fails to decode when all unsatisfied check nodes that have erased symbols more than two. In the case of $\mathcal{C}_1$, SER of "Conv." is about 10^2 times larger than that of "Prop." at $\epsilon = 0.3$. And this gap tends to smaller for ϵ larger than 0.3.

5.2.2 Decoding Complexity

Tables 1 and 2 show the average number (per iteration) of binary operations required for both decoding algorithms. From these tables, "Prop." needs slightly more operations than "Conv." at smaller value of ϵ. When ϵ takes larger value, "Prop." needs much more operations than "Conv.". This is because "Conv." fails to decode after a few operations are done. But "Prop."

Table 1: The average number of binary operations required for both decoding algorithms of code $\mathcal{C}_1$

ϵ	"Conv."(a)	"Prop."(b)	(b)/(a)
0.3	940.56079	940.5612337	1.000000472
0.325	982.9811643	983.0354183	1.000055193
0.35	1022.858608	1024.936272	1.002031233
0.375	1043.50082	1080.029256	1.035005661
0.4	989.9892207	1197.759013	1.209870762
0.425	796.321766	1327.219206	1.666687088
0.45	473.1815157	1253.13128	2.648309874

Table 2: The average number of binary operations required for both decoding algorithms of code $\mathcal{C}_2$

ϵ	"Conv."(a)	"Prop."(b)	(b)/(a)
0.35	2049.91816	2049.939081	1.000010206
0.375	2125.949063	2131.895305	1.002796982
0.4	2101.692417	2314.109714	1.10106964
0.425	1677.32257	2757.201832	1.643811322
0.45	908.989028	2691.852697	2.961369845
0.475	549.0381187	1824.04476	3.322255227

continues decoding after "Conv." fails to decode, and after at a cost of much operations, "Prop." often fails to decode. For $\epsilon < 0.3$, "Prop." attains smaller SER than "Conv." with almost the same complexity. This is a disadvantage of the proposed decoding algorithm.

6 Concluding Remarks

We have proposed new iterative decoding algorithm of LDPC codes over the BEC. The proposed decoding algorithm substitutes the check nodes that have two erased symbols to other check nodes. This procedure does not significantly increase the average number of operations required for decoding, since the parity-check matrix of LDPC codes is sparse. From simulation results, SER of the proposed decoding algorithm is much lower than that of the conventional decoding algorithm, and they have a favorable trade-off between remaining erasure rate and complexity.

For the further works, it should be needed to compare the performance of the proposed decoding algorithm with that proposed by H. P. Nik, et al. at first [2]. Nik's decoding algorithm often can decode much more erased symbols than that the conventional iterative decoding algorithm can, and its decoding complexity is a little greater when the maximum number of guesses is relatively small. But this algorithm has a possibility of decoding to a wrong codeword due to wrong guesses when $H_{\mathcal{E}}^{\mathrm{T}}$ is singular. Next, the performance analysis of the proposed decoding algorithm using stopping sets like a conventional iterative decoding algorithm is needed. From simulation results, the proposed decoding algorithm has a disadvantage that its decoding complexity becomes larger than that of the conventional decoding algorithm as ϵ increases. But it would be overcome by using the early stopping criterion to reduce the decoding complexity.

Acknowledgments

One of the authors, G. Hosoya, would like to thank Mr. T. Niinomi at Kanagawa Institute of Technology and Mr. H. Yagi at Waseda University for their valuable comments.

Bibliography

[1] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no.2, pp.569–584, Feb. 2001.

[2] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary-erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no.3, pp.439–454, March 2004.

[3] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol.8, pp.21–28, Jan. 1962.

[4] R. G. Gallager, *Low density parity check codes*, MIT Press, 1963.

[5] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.

[6] M. G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, "Improved low-denstiy parity-check codes using irregular graphs", *IEEE Trans. Inform. Theory*, vol.47, pp.585–598, Feb. 2001.

[7] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol.47, pp.599–618, Feb. 2001.

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