Transactions Letters

# Sequential Message-Passing Decoding of LDPC Codes by Partitioning Check Nodes

Sunghwan Kim, Min-Ho Jang, Jong-Seon No, *Member, IEEE,* Song-Nam Hong, and Dong-Joon Shin, *Member, IEEE*

***Abstract*—In this paper, we analyze the sequential message-passing decoding algorithm of low-density parity-check (LDPC) codes by partitioning check nodes. This decoding algorithm shows better bit error rate (BER) performance than the conventional message-passing decoding algorithm, especially for the small number of iterations. Analytical results indicate that as the number of partitioned subsets of check nodes increases, the BER performance is improved. We also derive the recursive equations for mean values of messages at check and variable nodes by using density evolution with a Gaussian approximation. From these equations, the mean values are obtained at each iteration of the sequential decoding algorithm and the corresponding BER values are calculated. They show that the sequential decoding algorithm converges faster than the conventional one. Finally, the analytical results are confirmed by the simulation results.**

***Index Terms*—Density evolution, flooding schedule, low-density parity-check (LDPC) codes, message-passing decoding, sequential decoding.**

## I. Introduction

IN 1996, low-density parity-check (LDPC) codes, originally invented by Gallager [1], were rediscovered by MacKay and Neal [2]. Since then, LDPC codes have been the main research topic in the coding area because they show the capacity-approaching performance with feasible complexity [3], [4]. Compared with turbo codes, they have lower decoding complexity due to the message-passing decoding based on the sum-product algorithm [5], but slower decoding convergence speed.

An LDPC code can be defined by a very sparse parity-check matrix which contains mostly 0's and a few 1's. The sparseness of the parity-check matrix gives low decoding complexity and better decoding performance. LDPC codes can be classified into two classes according to the degrees of variable nodes and check nodes. If the degrees of variable nodes and check nodes of an LDPC code are $d_v$ and $d_c$, respectively, it is called a $(d_v, d_c)$ regular LDPC code. Otherwise, it is called an irregular LDPC code.

Recently, there have been a lot of efforts on implementing LDPC decoder efficiently. In general, hardware implementation of LDPC decoder uses parallel processing. However, if the decoder cannot be implemented in the fully parallel processing mode, sequential decoding approach has to be taken.

An efficient sequential decoding algorithm and its hardware implementation are introduced in [6], where messages between each variable node and its neighbors are sequentially updated. A novel shuffled iterative decoding by partitioning variable nodes is introduced in [7]. This scheme has the same computational complexity as the iterative decoding based on flooding schedule and by simulation it is shown to converge faster. Similarly, a message-passing decoding algorithm which is sequentially performed on each variable node is introduced in [8] and the fast convergence of this algorithm is also verified by simulation. In [9]–[11], turbo product codes/single-parity check (TPC/SPC) codes are investigated, in which the message-passing decoding is performed by partitioning check nodes into two groups. Note that, in this paper, the sequential message-passing decoding algorithm based on various check node partitioning schemes is analyzed.

In [12] and [13], various new schedules have been proposed, in which the messages are exchanged at each iteration according to the parameters of the Tanner graph [14] such as the girths and the closed walks of the nodes. They are categorized as node based versus edge based, unidirectional versus bidirectional, and deterministic versus probabilistic, and the performance/complexity tradeoff is studied through simulation. Reliability-based schedule is also proposed, which performs the message-passing decoding using the reliability of each bit node instead of graph parameters [15]. In [16] and [17], new serial LDPC decoding algorithms which can be considered as the decoding using check node partitioning are introduced. They are especially suitable for hardware implementation. Also, by simulation, they are shown to converge faster than the iterative decoding based on flooding schedule.

In the conventional fully parallel message-passing decoding algorithm, many iterations (50 or more) are required to achieve the desired performance, which results in high decoding complexity. To reduce the number of iterations (or decoding complexity), the decoding should converge faster. The sequential message-passing decoding algorithms based on

Paper approved by A. H. Banihashemi, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received September 24, 2004; revised June 30, 2005, February 26, 2006, and March 8, 2007. This work was supported by the MOE, the MOCIE, and the MOLAB, Korea, through the fostering project of the Laboratory of Excellency. This paper was presented in part at ISITA, Seoul, Korea, October 2006.

S. Kim, M.-H. Jang, and J.-S. No are with the Department of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-742, Korea (e-mail: {nodoubt, mhjang}@ccl.snu.ac.kr; jsno@snu.ac.kr).

S. Hong and D.-J. Shin are with the Department of Electronics and Computer Engineering, Hanyang University, Seoul 133-791, Korea (e-mail: sunny795@ccrl.hanyang.ac.kr; djshin@hanyang.ac.kr).

Digital Object Identifier 10.1109/TCOMM.2008.040571.

variable node partitioning as in [7] and [8] or check node partitioning as in [16]–[18] converge faster than the fully parallel message-passing decoding algorithm and have the similar computational complexity.

Through the simulation, we can see that for some cases the decoding using check node partitioning converges faster than the decoding using variable node partitioning. However, in many cases, the difference seems negligible and it is generally accepted that both schemes have the similar convergence speed, which is confirmed through many simulations. In this paper, we will only consider partitioning of check nodes, analyze the sequential message-passing decoding algorithms based on various check node partitioning schemes, and suggest good partitioning schemes.

The sequential message-passing decoding algorithm can be briefly explained as follows. First, the check nodes are partitioned into $p$ subsets appropriately. Then, this LDPC code can be described by the interconnected $p$ subgraphs, each of which consists of the corresponding subset of check nodes and the connected variable nodes. The decoding can be performed by applying the message-passing decoding algorithm to each subgraph sequentially, that makes one iteration. In one iteration, the computational complexity of the sequential decoding algorithm is the same as that of the fully parallel decoding algorithm. If more efficient message update at variable nodes is used for the fully parallel message-passing decoding, the computational complexity of the sequential decoding might be a little bit higher but this additional complexity is not substantial. One drawback is that the decoding delay per iteration becomes larger than that of the fully parallel decoding as $p$ increases. However, it is suitable when the fully parallel decoding is not feasible.

This paper is organized as follows. In Section II, the conventional message-passing decoding algorithm is reviewed and the sequential message-passing decoding algorithm is introduced. In Section III, the means of messages and the corresponding bit error rate (BER) values for uniformly at random partitioning and bimodal partitioning at each iteration are derived by applying density evolution with a Gaussian approximation. In Section IV, two types of known LDPC codes suitable for the sequential message-passing decoding are described. Simulation results and conclusions are given in Sections V and VI, respectively.

## II. A Sequential Message-Passing Decoding Algorithm by Partitioning Check Nodes

In this section, the conventional message-passing decoding algorithm [3] using flooding schedule for $(d_v, d_c)$ regular LDPC codes will be reviewed and a sequential message-passing decoding algorithm by partitioning check nodes will be introduced. Each check node (or each variable node) receives messages from its $d_c$ neighbors (or its $d_v$ neighbors and its channel output), processes the messages, and passes the updated messages back to its neighbors. Each output message of a check node (or a variable node) is a function of all incoming messages to the node except for the incoming message on the edge where the output message will be sent.

Let $v$ and $u$ be messages in the log-likelihood ratio (LLR) form from variable node to check node and vice versa,

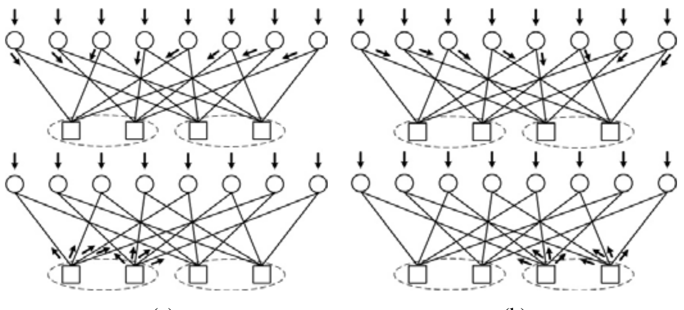

Fig. 1. One iteration in the sequential decoding of a $(2, 4)$ regular LDPC code with length 8 and $p = 2$. (a) Message-passing for the first check node subset. (b) Message-passing for the second check node subset.

respectively. Then $v$ can be updated by summing all incoming messages as

$$v = \sum_{i=0}^{d_v - 1} u_i \tag{1}$$

where $u_0$ is the received value from the channel at the variable node and $u_i, i = 1, \cdots, d_v - 1$, are the incoming messages from the neighbors except for the check node that receives the updated message $v$.

At the check node, the output message $u$ can be updated under 'tanh rule' as

$$\tanh \frac{u}{2} = \prod_{j=1}^{d_c - 1} \tanh \frac{v_j}{2} \tag{2}$$

where $v_j, j = 1, \cdots, d_c - 1$, are the incoming messages from the neighbors except for the variable node that receives the updated message $u$.

Each iteration in the conventional message-passing decoding algorithm consists of two steps. The first step is to calculate messages at all variable nodes and send them to check nodes and the second step is to calculate messages at all check nodes and send them to variable nodes. At each step, all the operations are performed simultaneously.

In the sequential message-passing decoding algorithm, we assume that the check nodes are partitioned into $p$ subsets. The messages from variable nodes to the check nodes in the first subset are updated and then the messages from the check nodes in the first subset to their neighbors are updated and sent. This decoding procedure is sequentially applied to the remaining $p - 1$ subsets of check nodes. One iteration in the sequential decoding algorithm includes all the above sequential message updating and passing for all variable nodes and all subsets of check nodes. Thus, it is clear that the amount of computations for one iteration in the sequential decoding algorithm is the same as that for one iteration in the conventional decoding algorithm if the update rule in $(1)$ is performed for each message at variable node.

Fig. 1 shows the sequential decoding procedure for a $(2, 4)$ regular LDPC code of length 8 when $p = 2$. Circle and square stand for variable node and check node, respectively. The messages received from the channel are represented by the arrows at the top of the circles. In Fig. 1 (b), the messages from variable nodes to the check nodes in the second subset are updated by using the messages from the check nodes in the first subset, which were already updated as in Fig. 1 (a). In the conventional message-passing decoding algorithm, messages of all variable and check nodes at the $l$-th iteration are updated by using the $(l-1)$-st updated messages and passed to their neighbors. Let $S_i$, $1 \leq i \leq p$, be the $i$-th subset of check nodes. During the $l$-th iteration in the sequential decoding algorithm, the $l$-th updated messages from $S_1, S_2, \cdots, S_{i-1}$ and the $(l-1)$-st updated messages from the remaining subsets are used for the $l$-th message updating between the variable nodes and the check nodes in $S_i$. Thus, in the sequential decoding algorithm, the messages are effectively updated up to $p$ times at each iteration without increasing computational complexity. This is the reason why the sequential decoding algorithm gives faster convergence speed.

From now on, we assume that all subsets of check nodes are as equal sized as possible because it has the following advantages. Compared with other partitioning cases, the least powerful processor can be used for the equal-sized partitioning because the processor should be able to process the messages from and to the biggest subset simultaneously. Also, the equal-sized partitioning shows the best performance, which can be verified through simulations.

## III. Analysis by Density Evolution with a Gaussian Approximation

Density evolution with a Gaussian approximation is based on approximating the probability densities of messages as Gaussian or Gaussian mixture [19]. Since this method is easier to analyze and computationally faster than the density evolution, it is useful for investigating the behavior of the message-passing decoding algorithm. In this section, we will only consider $(d_v, d_c)$ regular LDPC codes. For irregular LDPC codes, the similar analysis can be used.

Density evolution with a Gaussian approximation for the conventional message-passing decoding algorithm can be explained as follows. Let $m_u$ and $m_v$ be the means of $u$ and $v$, respectively. By taking the expectation at both sides of (1), we have

$$m_v^{(l)} = m_{u_0} + (d_v - 1)m_u^{(l-1)}. \tag{3}$$

The updated mean $m_u^{(l)}$ at the $l$-th iteration can be calculated by taking the expectation at both sides of (2), i.e.,

$$E\left[\tanh\frac{u^{(l)}}{2}\right] = E\left[\tanh\frac{v^{(l)}}{2}\right]^{d_c-1} \tag{4}$$

where $E\left[\tanh\frac{u}{2}\right] = \frac{1}{\sqrt{4\pi m_u}}\int_R \tanh\frac{u}{2} e^{-\frac{(u-m_u)^2}{4m_u}}du$.

Let $\phi(x)$ be the function defined as

$$\phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}}\int_R \tanh\frac{u}{2}e^{-\frac{(u-x)^2}{4x}}du, & \text{if } x > 0 \\ 1, & \text{if } x = 0 \\ 0, & \text{if } x = \infty. \end{cases} \tag{5}$$

By combining (3), (4), and (5), the recursive equation for $m_u$ can be derived as

$$m_u^{(l)} = \phi^{-1}\left(1 - \left[1 - \phi(m_{u_0} + (d_v-1)m_u^{(l-1)})\right]^{d_c-1}\right). \tag{6}$$

In the next subsections, density evolution analysis will be performed for two partitioning schemes, uniformly at random partitioning and bimodal partitioning. The former one assumes distributing check nodes equally probably among subsets but the latter one assumes some constraints for distributing check nodes among subsets, which may not be feasible for some cases. By considering these two rather extreme schemes, we can have an idea about how to partition the check nodes.

### *A. Density Evolution Analysis with Gaussian Approximation for Uniformly at Random Partitioning*

Suppose that all check nodes are partitioned into $p$ subsets. We assume that each check node is placed among $p$ subsets with probability $1/p$, which is called *uniformly at random partitioning*. Then, the sequential message-passing decoding algorithm can be analyzed as follows. Let $S_j$, $1 \leq j \leq p$, be the $j$-th subset of check nodes and $u_{S_j}$ be a message from a check node in $S_j$ to a variable node. Let $(a_1, a_2, \cdots, a_p)$ denote the distribution of edges from a variable node to $p$ subsets, where $a_j$ is the number of edges connected from a variable node to the check nodes in $S_j$ and $\sum_{j=1}^{p} a_j = d_v$. Then, the number of distinct edge distributions $(a_1, a_2, \cdots, a_p)$ for connections from a variable node to $p$ subsets is the repeated combination ${}_p\mathrm{H}_{d_v} = (d_v + p - 1)!/d_v!(p-1)!$, where $x!$ denotes $x \times (x-1) \times \cdots \times 2 \times 1$. When $e$ edges from a variable node are connected to $S_j$, the number of distinct edge distributions for connections from this variable node to $p$ subsets is ${}_{p-1}\mathrm{H}_{d_v-e}$.

For a given distribution $(a_1, a_2, \cdots, a_p)$ with $a_i \neq 0$, the mean of message from a variable node to the subset $S_i$ in the conventional message-passing decoding algorithm can be expressed as

$$m_{u_0} + (a_i - 1)m_{u_{S_i}}^{(l-1)} + \sum_{\substack{j=1 \\ j\neq i}}^{p} a_j m_{u_{S_j}}^{(l-1)}.$$

Since the message-passing decoding is sequentially performed from $S_1$ to $S_p$, the above equation should be modified as

$$m_{u_0} + (a_i - 1)m_{u_{S_i}}^{(l-1)} + \sum_{j=1}^{i-1} a_j m_{u_{S_j}}^{(l)} + \sum_{j=i+1}^{p} a_j m_{u_{S_j}}^{(l-1)}. \tag{7}$$

For the messages from a variable node to the subset $S_i$, it is assumed that $a_i$ can vary from 1 to $d_v$. Since the uniformly at random partitioning is assumed, the probability that the message with mean value in (7) is passed to the subset $S_i$ can be derived as

$$\frac{(d_v-1)!}{a_1!a_2!\cdots a_{i-1}!(a_i-1)!a_{i+1}!\cdots a_p!} \times \frac{1}{p^{d_v-1}}. \tag{8}$$

Using (6), (7), and (8), the recursive equation for the mean of message from the check node in $S_i$ to the variable node can be expressed as

$$m_{u_{S_i}}^{(l)} = \phi^{-1}\left(1 - \left[1 - \sum_{\substack{(a_1,\cdots,a_p)\\ a_i\neq 0}} \frac{(d_v-1)!}{a_1!a_2!\cdots(a_i-1)!\cdots a_p!p^{d_v-1}} \times\phi\left(m_{u_0} + (a_i-1)m_{u_{S_i}}^{(l-1)} + \sum_{j=1}^{i-1} a_j m_{u_{S_j}}^{(l)} + \sum_{j=i+1}^{p} a_j m_{u_{S_j}}^{(l-1)}\right)\right]^{d_c-1}\right). \quad (9)$$

By using the updated message in (9), we can derive the BER of a variable node. Also, the probability density function (pdf) $f(x)$ of the LLR message at a variable node, which is the combined value of intrinsic and extrinsic information, can be expressed as the following Gaussian mixture

$$f_v^{(l)}(x) = \sum_{(a_1,\cdots,a_p)} p_{(a_1,\cdots,a_p)} f_{(a_1,\cdots,a_p)}^{(l)}(x)$$

where $p_{(a_1,\cdots,a_p)} = d_v!/(a_1!a_2!\cdots a_p!p^{d_v})$ and $f_{(a_1,\cdots,a_p)}^{(l)}(x)$ is the Gaussian pdf with mean $m_{(a_1,\cdots,a_p)}^{(l)} = m_{u_0} + \sum_{j=1}^{p} a_j m_{u_{S_j}}^{(l)}$ and variance $\left(\sigma_{(a_1,\cdots,a_p)}^{(l)}\right)^2 = 2m_{(a_1,\cdots,a_p)}^{(l)}$ from the symmetric condition [3].

Since we assume that all-zero codeword is transmitted, the BER of codeword bits can be expressed as

$$P_e = \Pr(x<0) = \sum_{(a_1,\cdots,a_p)} p_{(a_1,\cdots,a_p)} Q\left(\sqrt{\frac{E_{(a_1,\cdots,a_p)}^{(l)}}{(\sigma_{(a_1,\cdots,a_p)}^{(l)})^2}}\right) \quad (10)$$

where $E_{(a_1,\cdots,a_p)}^{(l)}$ is the squared mean of the message at a variable node with the edge distribution $(a_1,a_2,\cdots,a_p)$ for the $l$-th iteration, $\sigma_{(a_1,\cdots,a_p)}^{(l)}$ is the standard deviation of the message at the $l$-th iteration, and $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\left(-\frac{u^2}{2}\right)du$. Since $E_{(a_1,\cdots,a_p)}^{(l)} = \left(m_{(a_1,\cdots,a_p)}^{(l)}\right)^2$ and $\left(\sigma_{(a_1,\cdots,a_p)}^{(l)}\right)^2 = 2m_{(a_1,\cdots,a_p)}^{(l)}$, we have $Q\left(\sqrt{E_{(a_1,\cdots,a_p)}^{(l)}/(\sigma_{(a_1,\cdots,a_p)}^{(l)})^2}\right) = Q\left(\sqrt{m_{(a_1,\cdots,a_p)}^{(l)}/2}\right)$. Thus, the BER in (10) can be rewritten as

$$P_e = \sum_{(a_1,\cdots,a_p)} p_{(a_1,\cdots,a_p)} Q\left(\sqrt{\frac{m_{(a_1,\cdots,a_p)}^{(l)}}{2}}\right).$$

### *B. Density Evolution Analysis with Gaussian Approximation for Bimodal Partitioning*

We consider the case that check nodes connected to a variable node are distributed among all subsets, $S_1, S_2, \cdots, S_p$, as evenly as possible. Then, among the check nodes connected to a variable node, the number of check nodes contained in each subset should be one of two consecutive numbers. Therefore, this scheme is called the *bimodal partitioning*. Note that if the number of check nodes connected to a variable node is divisible by the number of subsets, then each subset contains the same number of check nodes. The bimodal partitioning will be analyzed by considering the following two cases, $p < d_v$ and $p \geq d_v$.

*1) $p < d_v$:* Suppose $d_v = bp + r$, where $b$ is a positive integer and $0 \leq r \leq p-1$. Then $b$ edges from a variable node are connected to each subset and the edge distributions for connections from a variable node to $p$ subsets are determined by the connections of the remaining $r$ edges. Hence, the number of distinct edge distributions $(a_1, a_2, \cdots, a_p)$ for connections from a variable node to $p$ subsets is ${}_p\mathrm{C}_r = p!/r!(p-r)!$ and $b$ or $b+1$ edges from a variable node are connected to each subset. When $b$ edges from a variable node are connected to $S_i$, the number of distinct edge distributions for connections between this variable node and $p$ subsets is ${}_{p-1}\mathrm{C}_r$. When $b+1$ edges from a variable node are connected to $S_j$, the number of distinct edge distributions for connections between this variable node and $p$ subsets is ${}_{p-1}\mathrm{C}_{r-1}$. Then, for a given $(a_1, a_2, \cdots, a_p)$ where $a_j$ is $b$ or $b+1$, the mean of message $m_{v_{S_i}}^{(l)}$ from a variable node to the subset $S_i$ can be expressed as

$$m_{v_{S_i}}^{(l)} = m_{u_0} + (a_i-1)m_{u_{S_i}}^{(l-1)} + \sum_{j=1}^{i-1} a_j m_{u_{S_j}}^{(l)} + \sum_{j=i+1}^{p} a_j m_{u_{S_j}}^{(l-1)} \quad (11)$$

and the probability that the message with the mean value in (11) is passed to the subset $S_i$ can be derived as

$$\Pr\left(m_{v_{S_i}}^{(l)}\right) = \frac{a_i}{b \times_{p-1}\mathrm{C}_r + (b+1)\times_{p-1}\mathrm{C}_{r-1}}. \quad (12)$$

Using (6), (11), and (12), the recursive equation for the mean of message from the check node in the subset $S_i$ to the variable node can be expressed as

$$m_{u_{S_i}}^{(l)} = \phi^{-1}\left(1 - \left[1 - \sum_{(a_1,\cdots,a_p)} \frac{a_i}{b\times_{p-1}\mathrm{C}_r + (b+1)\times_{p-1}\mathrm{C}_{r-1}} \times\phi\left(m_{u_0} + (a_i-1)m_{u_{S_i}}^{(l-1)} + \sum_{j=1}^{i-1} a_j m_{u_{S_j}}^{(l)} + \sum_{j=i+1}^{p} a_j m_{u_{S_j}}^{(l-1)}\right)\right]^{d_c-1}\right).$$

Since there are ${}_p\mathrm{C}_r$ distinct edge distributions for the connections from a variable node to $p$ subsets, the pdf $f(x)$ of message at a variable node can be expressed as the following Gaussian mixture

$$f_v(x) = \sum_{k=1}^{{}_p\mathrm{C}_r} p_k f_{m_k^{(l)}}(x)$$

where $p_k = 1/{}_p\mathrm{C}_r$ and $f_{m_k^{(l)}}(x)$ is the Gaussian pdf with mean $m_k^{(l)} = m_{u_0} + \sum_{j=1}^{p} a_j m_{u_{S_j}}^{(l)}$ and variance $2m_k^{(l)}$. Then, the BER of codeword bits can be written as

$$P_e = \Pr(x<0) = \frac{1}{{}_p\mathrm{C}_r}\sum_{k=1}^{{}_p\mathrm{C}_r} Q\left(\sqrt{\frac{m_k^{(l)}}{2}}\right).$$

*2) $p \geq d_v$:* In this case, the number of distinct edge distributions for connections from a variable node to $p$ subsets becomes ${}_p\mathrm{C}_{d_v}$. Assume that one edge from a variable node is connected to the subset $S_i$. Then, the number of distinct edge distributions is ${}_{p-1}\mathrm{C}_{d_v-1}$. Then, for the given $(a_1, a_2, \cdots, a_p)$ where $a_j$ is zero or one, the mean of message from a variable node to the subset $S_i$ can be expressed as

$$m_{u_0} + \sum_{j=1}^{i-1} a_j m_{u_{S_j}}^{(l)} + \sum_{j=i+1}^{p} a_j m_{u_{S_j}}^{(l-1)}. \tag{13}$$

The probability that the message with the mean value in (13) is passed to the subset $S_i$ becomes $1/{}_{p-1}\mathrm{C}_{d_v-1}$.

The recursive equation for the mean of message from the check node in $S_i$ to the variable node can be expressed as

$$m_{u_{S_i}}^{(l)} = \phi^{-1}\left(1 - \left[1 - \frac{1}{{}_{p-1}\mathrm{C}_{d_v-1}} \sum_{(a_1,\cdots,a_p)} \phi\left(m_{u_0} + \sum_{j=1}^{i-1} a_j m_{u_{S_j}}^{(l)} + \sum_{j=i+1}^{p} a_j m_{u_{S_j}}^{(l-1)}\right)\right]^{d_c-1}\right).$$

Since there are ${}_p\mathrm{C}_{d_v}$ distinct edge distributions for the connections from a variable node to $p$ subsets, the pdf $f(x)$ of message at a variable node can be expressed as

$$f_v(x) = \sum_{k=1}^{{}_p\mathrm{C}_{d_v}} p_k f_{m_k^{(l)}}(x)$$

where $p_k = 1/{}_p\mathrm{C}_{d_v}$ and $f_{m_k^{(l)}}(x)$ is the Gaussian pdf with mean $m_k^{(l)} = m_{u_0} + \sum_{j=1}^{p} a_j m_{u_{S_j}}^{(l)}$ and variance $2m_k^{(l)}$. Then, the BER of codeword bits can be derived as

$$P_e = \Pr(x < 0) = \frac{1}{{}_p\mathrm{C}_{d_v}} \sum_{k=1}^{{}_p\mathrm{C}_{d_v}} Q\left(\sqrt{\frac{m_k^{(l)}}{2}}\right).$$

## C. Threshold and BER

In this subsection, we will compare the threshold values of LDPC codes when flooding schedule, uniformly at random partitioning, and bimodal partitioning are used. We also compare the BER performances of them.

*1) Threshold:* We calculated the threshold values of various regular LDPC codes when flooding schedule, uniformly at random partitioning, and bimodal partitioning are used. Since these threshold values for three schedules are identical, we can conclude that the threshold is the same regardless of the schedule, as expected. For example, the threshold value of (3, 6) regular LDPC code for three schedules is 0.87476. Note that the threshold values we have obtained are the same as those in [22].

*2) BER:* By using the results in the previous subsections, the BER curves for the cases of $p = 2, 3,$ and $4$ are obtained and compared. Equation (5) can be simplified as $\phi(x) \approx e^{-0.4527x^{0.86}+0.0218}$ [19]. We consider binary phase shift keying (BPSK) modulation and the additive white Gaussian noise (AWGN) channels with the standard deviation values 0.81 and 0.85 which are less than the threshold value 0.87476 of (3, 6) regular LDPC code.

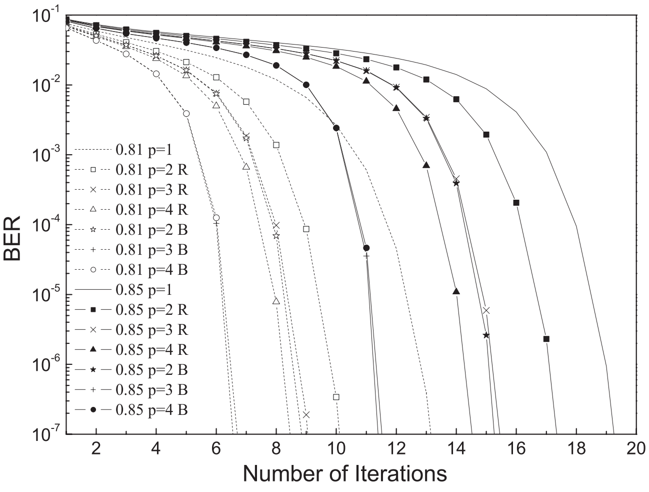

Fig. 2. Analytical BER of (3, 6) regular LDPC code using various schedules in the AWGN channel with $\sigma = 0.81$ and 0.85.

Fig. 2 shows the analytical BER of $(3, 6)$ regular LDPC code for the uniformly at random partitioning and bimodal partitioning with $p = 2, 3,$ and $4$, where R and B stand for the uniformly at random partitioning and bimodal partitioning, respectively. It is shown in Fig. 2 that BER performance of the sequential decoding algorithm is better than that of the conventional decoding ($p = 1$) and the sequential decoding algorithm with bimodal partitioning is superior to that with the uniformly at random partitioning. The bimodal partitioning seems to make more effective message update in one iteration than the uniformly at random partitioning does because edges of all variable nodes in the bimodal partitioning are connected to the subsets as evenly as possible and the message update for each subset utilizes the similar number of messages from variable nodes.

It is also shown that as the number $p$ of subsets increases, the sequential decoding converges faster. However, the gain in the convergence speed becomes negligible for $p \geq 4$. This can be explained that since the (maximum) degree of variable nodes is 3, even if $p$ goes beyond 3, the overall bimodal partitioning has the similar structure as the case of $p = 3$ and therefore the performance becomes similar. For example, if $d_v$ is 4, then the performance will substantially improve up to $p = 4$. The similar trends as those of the regular codes have also been observed for the tested irregular codes of the same code rate and length.

## IV. LDPC Codes Suitable for the Sequential Decoding with Bimodal Partitioning

The sequential message-passing decoding algorithm can be applied to any code which can be represented by Tanner graph by using uniformly at random partitioning, bimodal partitioning, or some partitioning between them. Especially, the sequential decoding using bimodal partitioning can be efficiently applied to the following types of LDPC codes.

The $(J, L)$ regular quasi-cyclic (QC) LDPC codes have the parity-check matrix constructed from $q \times q$ circulant permutation matrices [20]. This parity-check matrix can be

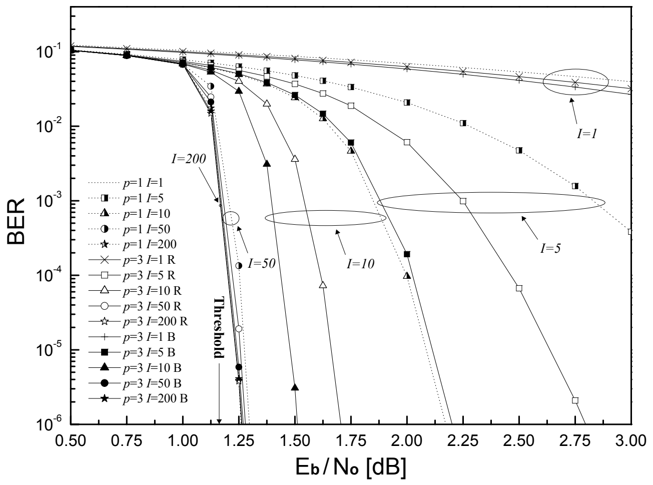

Fig. 3. Simulated BER performance of (3, 6) regular LDPC code with length 100008 and rate 1/2 (threshold = 1.162 dB).

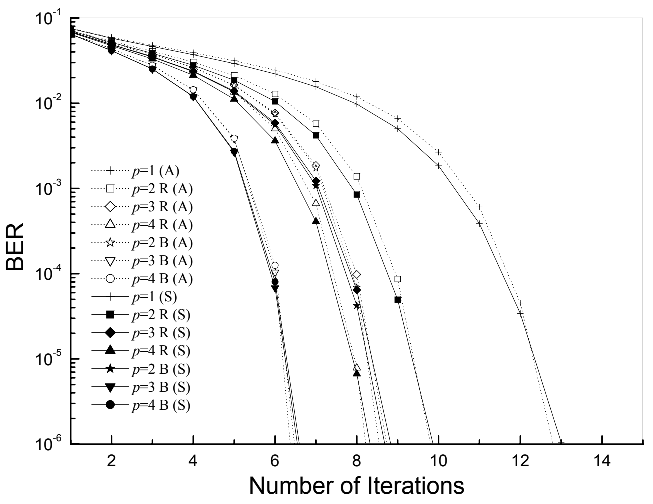

Fig. 4. Performance comparison of analytical and simulation results of bimodal and uniformly at random partitionings of (3, 6) regular LDPC code of length 100008 in the AWGN channel with $\sigma = 0.81$.

used to construct regular LDPC code of length $n = Lq$. Let $I(q_{j,l})$ be the $q \times q$ circulant permutation matrix with '1' at the column $(r + q_{j,l}) \mod q$ for the row $r$, $0 \le r \le q-1$, and '0', elsewhere. Then the parity-check matrix $H$ of a $(J, L)$ QC LDPC code can be represented as

$$H = \begin{bmatrix} I(0) & I(0) & \cdots & I(0) \\ I(0) & I(q_{1,1}) & \cdots & I(q_{1,L-1}) \\ \vdots & & \cdots & \vdots \\ I(0) & I(q_{J-1,1}) & \cdots & I(q_{J-1,L-1}) \end{bmatrix}.$$

For the above $(J, L)$ QC LDPC code, we can perform the bimodal partitioning by dividing the $Jq$ rows (or the check nodes) of $H$ into $p$ subsets such that each of them contains the $\lfloor Jq/p \rfloor$ or $\lfloor Jq/p \rfloor$+1 consecutive rows where $\lfloor x \rfloor$ is the biggest integer which is less than or equal to $x$. For the cases of $p = 2$, $p|J$, or $J|p$, it is obvious that edges from variable nodes are evenly distributed over $p$ subsets, i.e., satisfying the bimodal partitioning.

A protograph was introduced in [21], which can be any Tanner graph, typically one with a relatively small number of nodes. By using the protograph approach, we can easily construct LDPC codes such that check nodes can be easily partitioned to satisfy bimodal partitioning. If we want to construct an LDPC code of which the check nodes can be evenly partitioned into $p$ subsets, a protograph with $p$ check nodes can be used.

For the protograph, parallel edges are permitted and the numbers of variable nodes and edges are determined according to the code rate and degree distribution of the desired LDPC code. Then, by copying this protograph $w$ times, we have the overall graph with $w$ disconnected subgraphs, where the code length of the desired LDPC code is $wpd_c/d_v$. Let $K_i$ be the set of all edges connected to the $i$-th check node in each subgraph. The $w$ subgraphs are interconnected by a permutation $\Pi = (\Pi_1, \cdots, \Pi_p)$ on all edges, where $\Pi_i$ is a permutation performed only on the edges contained in $K_i$. Then the edges from any variable node are connected to the $p$ subsets of check nodes as evenly as possible. Such an LDPC code keeps the features of protograph and can be efficiently decoded by using the sequential decoding algorithm.

## V. Simulation Results

Simulation is performed to compare the BER performance of sequential decoding algorithm with uniformly at random partitioning and bimodal partitioning for the various numbers of subsets and iterations. For the simulation, a (3, 6) regular LDPC code of length 100008 and rate 1/2 is constructed to have girth 6, and BPSK modulation and AWGN channel are assumed.

Fig. 3 shows that the BER performance of the sequential decoding algorithm is better than that of the conventional decoding algorithm, especially for 5 and 10 iterations. Note that R and B denote the uniformly at random partitioning and bimodal partitioning, respectively, and *I* stands for the number of iterations. Also, the BER performance of the bimodal partitioning is better than that of the uniformly at random partitioning for the small number of iterations. However, for 50 or more iterations, the BER improvement decreases because three schedules have enough iteration gain. For 200 iterations, the performances of three schedules are almost identical, that is well matched with the fact that three schedules have the same threshold value, and the gap between the simulated performance and the threshold value (1.162 dB) is less than 0.1 dB, which confirms the validity of the analytical results.

Fig. 4 compares the simulation results with the analytical results for three schedules using a (3, 6) LDPC code of length 100008 in the AWGN channel with $\sigma = 0.81$. In Fig. 4, A and S stand for the analytical result and the simulation result, respectively. We can see that the simulation results are fairly well matched with the analytical results. The slight difference may be due to the effect of the short cycles and the confidence interval of the simulation. It is well known that the belief propagation decoding algorithm is suboptimal when LDPC codes have cycles and there appears an error floor due to the short cycles. Therefore, in general, as the number of iterations increases, the performance gap between the analytical result and the simulation result also increases because of the short cycles and error floor. We observed 23346 bit errors (and 8581 codeword errors) to obtain the simulated BER = $6.67 \times 10^{-6}$ (and FER = $2.45 \times 10^{-1}$) for the uniformly at random partitioning with $p = 4$ subsets. For all other simulation points, we observed more bit errors (and codeword errors) to obtain BER values in Fig. 4. This indicates the confidence interval which may result in the slight difference between the analytical and simulation results. However, the simulation results in Fig. 4 are good enough to confirm the analytical results for the uniformly at random partitioning and bimodal partitioning.

## VI. Conclusions

The sequential message-passing decoding algorithm with check node partitioning outperforms the conventional decoding algorithm, especially for the small number of iterations. This implies that the sequential algorithm improves the convergence speed without increasing the decoding complexity. By using density evolution with a Gaussian approximation, we investigated the reason why the sequential decoding algorithm has faster convergence speed. Moreover, the sequential decoding algorithm can be easily applied to LDPC codes and is useful to implement the practical decoder when the computing power is limited.

## References

[1] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.

[2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEEE Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[3] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[4] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[5] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 533–547, Feb. 2001.

[6] M. Cocco, J. Dielissen, M. Heijligers, A. Hekstra, and J. Huisken, "A scalable architecture for LDPC decoding," in *Proc. DATE'04*, Feb. 2004, pp. 88–93.

[7] J. Zhang and M. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.* vol. 53, no. 2, pp. 209–213, Feb. 2005.

[8] H. Kfir and I. Kanter, "Parallel versus sequential updating for belief propagation decoding," *Physica A, Elsevier*, vol. 330, pp. 259–270, 2003.

[9] J. Li, E. Kurtas, K. R. Narayanan, and C. N. Georghiades, "On the performance of turbo product codes over partial response channels," *IEEE Trans. Magnetics*, vol. 37, no. 4, pp. 1932–1943, July 2001.

[10] C. Argon and S. W. McLaughlin, "A parallel decoder for low latency decoding of turbo product codes," *IEEE Commun. Lett.*, vol. 6, no. 2, pp. 70–72, Feb. 2002.

[11] J. Li, K. R. Narayanan, E. Kurtas, and C. N. Georghiades, "On the performance of high-rate TPC/SPC codes and LDPC codes over partial response channels," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 723–734, May 2002.

[12] Y. Mao and A. H. Banihashemi, "Decoding low-density parity-check codes with probabilistic schedule," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 414–416, Oct. 2001.

[13] H. Xiao and A. H. Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2098–2105, Dec. 2004.

[14] R. Tanner, "A recursive approach to low complexity code," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.

[15] A. Nouh and A. H. Banihashemi, "Reliability-based schedule for bit-flipping decoding of low-density parity-check codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2038–2040, Dec. 2004.

[16] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "High throughput low-density parity-check decoder architectures," in *Proc. GLOBECOM'01*, Nov. 2001, pp. 3019–3024.

[17] M. M. Mansour and N. R. Shanbhag, "Turbo decoder architecture for low-density parity-check codes," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 2002, pp. 1383–1388.

[18] E. Sharon, S. Litsyn, and J. Goldberger, "An efficient message-passing schedule for LDPC decoding," in *Proc. Electrical and Electronics Engineers in Israel*, Sept. 2004, pp. 223–226.

[19] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambrideg, MA, Sept. 2000.

[20] M. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[21] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protograph," *IPN Progress Report 42–154*, JPL, Aug. 2003.

[22] S.-Y. Chung, "Analysis of sum-product decoding of low-density parity-check codes using a Guassian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.