MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Reduced Latency Iterative Decoding of LDPC Codes

Yige Wang, Juntan Zhang, Marc Fossorier, Jonathan S. Yedidia

TR2005-103 November 2005

Abstract

Reduced latency versions of iterative decoders of low-density parity-check codes are analyzed in this paper. The proposed schemes converge faster than standard approaches. Two methods, density evolution and EXIT charts, are used to analyze the performance of the proposed algorithms. Both theoretical analysis and simulations show that the new schedules offer good performance versus complexity and latency trade-offs.

To appear in the Proceedings of GlobeCom 2005.

Reduced Latency Iterative Decoding of LDPC Codes

Yige Wang, Juntan Zhang and Marc Fossorier
Department of Electrical Engineering
University of Hawaii at Manoa, Honolulu, HI 96822
Email: yige, juntan, marc@spectra.eng.hawaii.edu

Jonathan S. Yedidia
Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge, MA 02139
Email: yedidia@merl.com

***Abstract*— Reduced latency versions of iterative decoders of low-density parity-check codes are analyzed in this paper. The proposed schemes converge faster than standard approaches. Two methods, density evolution and EXIT charts, are used to analyze the performance of the proposed algorithms. Both theoretical analysis and simulations show that the new schedules offer good performance versus complexity and latency trade-offs.**

I. INTRODUCTION

Iterative decoding based on belief propagation (BP) [1] has received significant attention recently, mostly due to its near-Shannon-limit error performance for the decoding of low-density parity-check (LDPC) codes. In order to take advantage of more reliable extrinsic messages, a shuffled BP decoding method was introduced in [2][3]. This method was shown to converge faster than the standard BP decoder. In [4][5], a new approach which further speeds up the convergence of iterative decoders was presented. This method uses replicas of subdecoders working in parallel. In this paper, these algorithms are analyzed by density evolution [6] and extrinsic information transfer (EXIT) charts [8][9]. Both show that shuffled BP converges about twice as fast as standard BP and replica shuffled BP converges faster than plain shuffled BP. The convergence speed of replica shuffled BP is determined by the number of subdecoders and the specific information updating scheme used.

II. ITERATIVE BP DECODING OF LDPC CODES

In general, LDPC codes can be categorized into regular LDPC codes and irregular LDPC codes. Both can be represented by a bipartite graph with N variable nodes on the left and M check nodes on the right. A bipartite graph is specified by sequences $(\lambda_1, \lambda_2, \cdots, \lambda_{d_v})$ and $(\rho_1, \rho_2, \cdots, \rho_{d_c})$, where $\lambda_i(\rho_i)$ represents the fraction of edges with left (right) degree i, and d_v and d_c are the maximum variable degree and check degree, respectively.

A. Standard BP for iterative decoding of LDPC codes

Suppose a regular binary $(N, K)(d_v, d_c)$ LDPC code $\mathbf{C}$ is used for error control over an AWGN channel with zero mean and power spectral density $N_0/2$. Assume BPSK signaling with unit energy, which maps a codeword $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ into a transmitted sequence $\mathbf{q} = (q_1, q_2, \ldots, q_N)$, according to $q_n = 1 - 2w_n$, for $n = 1, 2, \ldots, N$. If $\mathbf{w} = [w_n]$ is a codeword in $\mathbf{C}$ and $\mathbf{q} = [q_n]$ is the corresponding transmitted sequence, then the received sequence is $\mathbf{q} + \mathbf{g} = \mathbf{y} = [y_n]$, with $y_n = q_n + g_n$, where for $1 \le n \le N$, g_n's are statistically independent Gaussian random variables with zero mean and variance $N_0/2$. Let $\mathbf{H} = [H_{mn}]$ be the parity check matrix which defines an LDPC code. We denote the set of bits that participate in check m by $\mathcal{N}(m) = \{n : H_{mn} = 1\}$ and the set of checks in which bit n participates as $\mathcal{M}(n) = \{m : H_{mn} = 1\}$. We also denote $\mathcal{N}(m)\backslash n$ as the set $\mathcal{N}(m)$ with bit n excluded, and $\mathcal{M}(n)\backslash m$ as the set $\mathcal{M}(n)$ with check m excluded. We define the following notations associated with the ith iteration:

- $U_{ch,n}$: The log-likelihood ratio (LLR) of bit n which is derived from the channel output y_n. In BP decoding, we initially set $U_{ch,n} = \frac{4}{N_0} y_n$.
- $U_{mn}^{(i)}$: The LLR of bit n which is sent from check node m to bit node n.
- $V_{mn}^{(i)}$: The LLR of bit n which is sent from the bit node n to check node m.
- $V_n^{(i)}$: The *a posteriori* LLR of bit n.

The standard BP algorithm is carried out as follows [1]:

Initialization: Set $i = 1$, and the maximum number of iteration to I_{Max}. For each m,n, set $V_{mn}^{(0)} = U_{ch,n}$.

Step 1: (i) Horizontal Step, for $1 \le n \le N$ and each $m \in \mathcal{M}(n)$, process:

$$U_{mn}^{(i)} = 2 \tanh^{-1} \prod_{n' \in \mathcal{N}(m)\backslash n} \tanh \frac{V_{mn'}^{(i-1)}}{2} \qquad (1)$$

(ii) Vertical Step, for $1 \le n \le N$ and each $m \in \mathcal{M}(n)$, process:

$$V_{mn}^{(i)} = U_{ch,n} + \sum_{m' \in \mathcal{M}(n)\backslash m} U_{m'n}^{(i)} \qquad (2)$$

$$V_n^{(i)} = U_{ch,n} + \sum_{m \in \mathcal{M}(n)} U_{mn}^{(i)}$$

Step 2: Hard decision and stopping criterion test:

(i) Create $\hat{\boldsymbol{w}}^{(i)} = [\hat{w}_n^{(i)}]$ such that $\hat{w}_n^{(i)} = 1$ if $V_n^{(i)} < 0$, and $\hat{w}_n^{(i)} = 0$ if $V_n^{(i)} \ge 0$.

(ii) If $\boldsymbol{H}\hat{\boldsymbol{w}}^{(i)} = \mathbf{0}$ or the maximum iteration number I_{Max} is reached, stop the decoding iteration and go to Step 3. Otherwise set $i := i+1$ and go to Step 1.

Step 3: Output $\hat{\boldsymbol{w}}^{(i)}$ as the decoded codeword.

B. Plain shuffled BP for iterative decoding of LDPC codes

In general, for both the check-to-bit messages and bit-to-check messages, the more independent information is used to update the messages, the more reliable they become. Iteration-i of the standard two step implementation of the BP algorithm uses all values $V_{mn'}^{(i-1)}$ computed at the previous iteration in (1). However certain values $V_{mn'}^{(i)}$ could already be computed based on a partial computation of the values $U_{mn}^{(i)}$ obtained from (2), and then be used instead of $V_{mn'}^{(i-1)}$ in (1) to compute the remaining values $U_{mn}^{(i)}$. This suggests a shuffling of the horizontal and vertical steps of the standard BP decoding. Hence we refer to this new version as shuffled BP decoding.

In the shuffled BP algorithm, the initialization, stopping criterion test and output steps remain the same as in the standard BP algorithm. The only difference between the two algorithms lies in the updating procedure. Step 1 of the shuffled BP algorithm is modified as: for $1 \le n \le N$ and each $m \in \mathcal{M}(n)$, process the horizontal step and vertical step **jointly**, with (1) modified as [2][3]:

$$U_{mn}^{(i)} = 2\tanh^{-1}\Bigg(\prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'<n}} \tanh\frac{V_{mn'}^{(i)}}{2} \prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'>n}} \tanh\frac{V_{mn'}^{(i-1)}}{2}\Bigg) \quad (3)$$

C. Replica shuffled BP decoding for LDPC codes

In replica shuffled BP decoding, several shuffled subdecoders based on different updating orders operate simultaneously and cooperatively. After each iteration, each subdecoder receives more reliable messages from and sends more reliable messages to other subdecoders [4][5]. Based on these more reliable messages, all replica subdecoders begin the next iteration. For two replica, let $\overrightarrow{D}$ and $\overleftarrow{D}$ denote the subdecoders with natural increasing and decreasing updating order, respectively. Let $\overrightarrow{U}_{mn}^{(i)}$ and $\overrightarrow{V}_{mn}^{i}$ be the variables associated with $\overrightarrow{D}$ at iteration i. The variables associated with $\overleftarrow{D}$ are defined in a similar way. Replica shuffled BP decoding with two replica subdecoders is carried out as follows:

Initialization: Set $i = 1$, maximum number of iteration to I_{Max}. For each m,n, set $\overrightarrow{V}_{mn}^{(0)} = \overleftarrow{V}_{mn}^{(0)} = U_{ch,n}$.

Step 1: Each replica subdecoder processes the following two steps simultaneously. For $1 \le n \le N$ and each $m \in \mathcal{M}(n)$, process

(i) Horizontal Step

$$\overrightarrow{U}_{mn}^{(i)} = 2\tanh^{-1}\Bigg(\prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'<n}} \tanh\frac{\overrightarrow{V}_{mn'}^{(i)}}{2} \prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'>n}} \tanh\frac{\overrightarrow{V}_{mn'}^{(i-1)}}{2}\Bigg)$$

$$\overleftarrow{U}_{mn}^{(i)} = 2\tanh^{-1}\Bigg(\prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'>n}} \tanh\frac{\overleftarrow{V}_{mn'}^{(i)}}{2} \prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'<n}} \tanh\frac{\overleftarrow{V}_{mn'}^{(i-1)}}{2}\Bigg)$$

(ii) Vertical Step

$$\overrightarrow{V}_{mn}^{(i)} = U_{ch,n} + \sum_{m'\in\mathcal{M}(n)\backslash m} \overrightarrow{U}_{m'n}^{(i)}$$

$$\overleftarrow{V}_{mn}^{(i)} = U_{ch,n} + \sum_{m'\in\mathcal{M}(n)\backslash m} \overleftarrow{U}_{m'n}^{(i)}$$

Step 2: Set $\overrightarrow{V}_{mn}^{(i)} = \overleftarrow{V}_{mn}^{(i)}$ for $1 \le n \le N/2$ and $\overleftarrow{V}_{mn}^{(i)} = \overrightarrow{V}_{mn}^{(i)}$ for $N/2 < n \le N$.

Step 3: Hard decision and stopping criterion test:

(i) Create $\hat{\boldsymbol{w}}^{(i)} = [\hat{w}_n^{(i)}]$ such that for $1 \le n \le N/2$, $\hat{w}_n^{(i)} = 1$ if $U_{ch,n} + \sum_{m\in\mathcal{M}(n)} \overleftarrow{U}_{mn}^{(i)} < 0$, and $\hat{w}_n^{(i)} = 0$ otherwise; for $N/2 < n \le N$, $\hat{w}_n^{(i)} = 1$ if $U_{ch,n} + \sum_{m\in\mathcal{M}(n)} \overrightarrow{U}_{mn}^{(i)} < 0$, and $\hat{w}_n^{(i)} = 0$ otherwise.

(ii) If $\boldsymbol{H}\hat{\boldsymbol{w}}^{(i)} = \mathbf{0}$ or the maximum iteration number I_{Max} is reached, stop the decoding iteration and go to Step 4. Otherwise set $i := i+1$ and go to Step 1.

Step 4: Output $\hat{\boldsymbol{w}}^{(i)}$ as the decoded codeword.

Another possible approach that can be used is for the two subdecoders to exchange more reliable messages synchronously with each other during the decoding process. Define $R(n) = \{n'|1 \le n' \le N, n \le n' \le N-n\}$, and $\overline{R}(n) = \{n'|1 \le n' \le N, n' \notin R(n)\}$, for $1 \le n \le N$. In synchronous scheme, the updating and exchanging procedures operate simultaneously:

Step 1: For $1 \le n \le N$ and each $m \in \mathcal{M}(n)$, for $p = N-n$ and $q \in \mathcal{M}(p)$, two replica subdecoders process the following two steps simultaneously

(i) Horizontal Step

$$U_{mn}^{(i)} = 2\tanh^{-1}\Bigg(\prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'\in\overline{R}(n)}} \tanh\frac{V_{mn'}^{(i)}}{2} \prod_{\substack{n'\in\mathcal{N}(m)\backslash n\\ n'\in R(n)}} \tanh\frac{V_{mn'}^{(i-1)}}{2}\Bigg)$$

$$U_{qp}^{(i)} = 2\tanh^{-1}\Bigg(\prod_{\substack{p'\in\mathcal{N}(q)\backslash p\\ p'\in\overline{R}(N-p)}} \tanh\frac{V_{qp'}^{(i)}}{2} \prod_{\substack{p'\in\mathcal{N}(q)\backslash p\\ p'\in R(N-p)}} \tanh\frac{V_{qp'}^{(i-1)}}{2}\Bigg)$$

(ii) Vertical Step

$$V_{mn}^{(i)} = U_{ch,n} + \sum_{m'\in\mathcal{M}(n)\backslash m} U_{m'n}^{(i)}$$

$$V_{qp}^{(i)} = U_{ch,p} + \sum_{q'\in\mathcal{M}(p)\backslash q} U_{q'p}^{(i)}$$

Notice that in this case the two replica subdecoders use the same set of bit-to-check LLR values. It is also straightforward to extend the replica shuffled BP decoding to the cases in which more than two replica subdecoders are used. In order to decrease decoding delay of plain shuffled BP decoding, a parallel version of shuffled BP named group shuffled BP was developed in [2]. In a similar way, group replica shuffled BP can also preserve the parallelism advantage of the standard BP algorithm [4][5].

III. Analysis by Density Evolution

A. Density evolution of replica shuffled BP

Density evolution [6] is an effective numerical method to analyze the performance of message passing iterative decoding algorithms. It has been shown that for a given message-passing decoding, if the channel and the decoder satisfy symmetry conditions [6], then the decoding bit error rate is independent of the transmitted sequence. The process of density evolution therefore can be greatly simplified by assuming the all-zero sequence was transmitted. It is straightforward to verify that the shuffled and replica shuffled BP decoder satisfy the symmetry condition thus the all-zero transmitted codeword assumption is valid. In density evolution of shuffled and replica shuffled BP, cycle-free structure of LDPC code is assumed as in [6]. In this case, the incoming messages to any bit or check node are independent, which also simplifies the derivation of the pdf of the outgoing messages. Density evolution results for serial BP decoding of LDPC codes can also be found in [7].

In shuffled and replica shuffled BP decoding, pdf's of outgoing and incoming messages of bit nodes are dependent on the bit index number n. Let $f^{(i)}_{U_n}(u)$ and $f^{(i)}_{V_n}(v)$ be the pdf's of the incoming and outgoing messages of bit node n at iteration i, respectively. In standard BP, infinite codeword length is assumed while in shuffled BP we consider a large enough codeword length N and assume the cycle-free condition still holds.

For $n = 1, 2, \ldots, N$, for bit node processor of shuffled BP, the density evolution is the same as that of standard BP

$$f^{(i)}_{V_n} = \mathcal{F}^{-1}\left(\mathcal{F}(f_{U_{ch}}) \cdot \left(\mathcal{F}(f^{(i)}_{U_n})\right)^{dc-1}\right) \quad (4)$$

where $\mathcal{F}$ denotes Fourier transform.

From (3), $U^{(i)}_n$ depends on both $V^{(i)}_{n'}$ for $n' < n$ and $V^{(i-1)}_{n'}$ for $n' > n$. To avoid brute force calculation of all possible combinatorial formats of $V^{(i)}_{n'}$ and $V^{(i-1)}_{n'}$, we let the average pdf of the newly delivered incoming messages to check nodes adjacent to bit node n at iteration i be

$$f^{(i)}_{\overline{V}_{n'<n}}(v) = \frac{1}{n-1}\sum_{n'=1}^{n-1} f^{(i)}_{V_{n'}}(v). \quad (5)$$

Similarly, we let the average pdf of the incoming messages from bit node $\{b_{n'} | n' > n\}$ to check nodes adjacent to bit node n be

$$f^{(i-1)}_{\overline{V}_{n'>n}}(v) = \frac{1}{N-n}\sum_{n'=n+1}^{N} f^{(i-1)}_{V_{n'}}(v). \quad (6)$$

Note that the check node processing can be implemented in a recursive way [10]. Define a core operation as

$$\Psi(V_1, V_2) = 2\tanh^{-1}\left(\tanh\left(\frac{V_1}{2}\right)\tanh\left(\frac{V_2}{2}\right)\right) \quad (7)$$

Then (1) can be calculated by applying (7) recursively:

$$U = \Psi(\ldots\Psi(\Psi(V_1, V_2), V_3), \ldots, V_{d_c-1}). \quad (8)$$

If the incoming messages are i.i.d. random variables with pdf $f_V(v)$, the pdf of the outgoing message can be efficient computed as [10]

$$f_U = \Psi^{d_c-1} f_V. \quad (9)$$

Let us consider plain shuffled BP with natural increasing ordering. For a belief message incoming to bit node n, the incoming messages incoming to check node adjacent to bit node n have in total $\binom{N-1}{d_c-1}$ possible formats. For each $j = 0, 1, \ldots, d_c - 1$, there are $\binom{n-1}{j} \cdot \binom{N-n}{d_c-1-j}$ possible formats which contain j newly delivered bit-to-check messages at the current iteration and d_c-1-j bit-to-check messages delivered at the previous iteration. The average pdf incoming to bit node b_n at iteration i becomes

$$f^{(i)}_{U_n} = \sum_{j=0}^{d_c-1} \frac{\binom{n-1}{j} \cdot \binom{N-n}{d_c-1-j}}{\binom{N-1}{d_c-1}} \cdot \Psi^j f^{(i)}_{\overline{V}_{n'<n}} \cdot \Psi^{d_c-1-j} f^{(i-1)}_{\overline{V}_{n'>n}} \quad (10)$$

It is straightforward to extend these updating rules of pdf's in shuffled BP to replica shuffled BP. For instance, in replica shuffled BP with two subdecoders, the updating rule of the pdf of outgoing belief messages from bit nodes are the same as in plain shuffled BP, while the pdf's of incoming belief messages to bit nodes are modified as

$$f^{(i)}_{V_n} \leftarrow f^{(i)}_{V_{N-n}} \quad (11)$$

for $1 \leq n \leq N/2$. The density evolution of replica shuffled BP with more than two subdecoders can be obtained in a similar way. The extension of density evolution of shuffled and replica shuffled BP for decoding irregular LDPC codes is also straightforward. Consider an irregular LDPC code with degree distributions $\lambda(x) = \sum_{l=1}^{dv} \lambda_l x^{l-1}$ and $\rho(x) = \sum_{l=1}^{dc} \rho_l x^{l-1}$. Consider plain shuffled BP decoding in natural increasing order. From the above analysis, at iteration i the pdf of incoming messages to bit node n from a check node with degree l is

$$f^{(i)}_{U_{n,l}} = \sum_{j=0}^{l-1} \frac{\binom{n-1}{j} \cdot \binom{N-n}{l-1-j}}{\binom{N-1}{l-1}} \cdot \Psi^j f^{(i)}_{\overline{V}_{n'<n}} \cdot \Psi^{l-1-j} f^{(i-1)}_{\overline{V}_{n'>n}}. \quad (12)$$

Since the pdf's of the outgoing messages of check node with different degree are distinct, the expectation of these pdf's is the overall pdf of message incoming to bit node n

$$f^{(i)}_{U_n} = \sum_{l=1}^{dc} \rho_l \sum_{j=0}^{l-1} \frac{\binom{n-1}{j} \cdot \binom{N-n}{l-1-j}}{\binom{N-1}{l-1}} \cdot \Psi^j f^{(i)}_{\overline{V}_{n'<n}} \cdot \Psi^{l-1-j} f^{(i-1)}_{\overline{V}_{n'>n}}. \quad (13)$$

Similarly, the pdf of outgoing messages from bit node n at iteration i becomes

$$f_{V_n}^{(i)} = \sum_{l=1}^{dv} \lambda_l \mathcal{F}^{-1}\left(\mathcal{F}(f_{U_{ch}}) \cdot \left(\mathcal{F}(f_{U_n}^{(i)})\right)^{l-1}\right) \quad (14)$$

B. Simulation results

Fig. 1 depicts the BER as a function of the numbers of decoding iterations predicted by density evolution with standard BP, shuffled BP, replica shuffled BP with two and four subdecoders (synchronous exchanging) methods, for decoding a rate-1/2 $(3,6)$ regular LDPC codes with $E_b/N_o = 1.111dB$. We observe that shuffled BP converges about twice as fast as the standard BP decoding while replica shuffled BP converges faster than plain shuffled BP. As expected, we observe that the larger the number of subdecoders in replica shuffled BP, the faster the convergence of decoding.

Fig. 2 depicts the BER versus the number of iterations predicted by density evolution with replica shuffled BP decoder of two subdecoders using non-synchronous and synchronous exchanging schemes, for a $(3,6)$ regular LDPC code. We observe that replica shuffled BP under synchronous exchanging scheme converges faster than under non-synchronous exchanging schedule. It is also worth mentioning that the synchronous scheme requires less memory than the non-synchronous scheme, but more frequent memory access.

Similar results have been obtained for irregular LDPC codes [4].

IV. Analysis by EXIT chart

A. EXIT chart of replica shuffled BP

EXIT chart [8][9] is another effective technique to study the convergence behavior of iterative decoding. It is easy to visualize and program and is a good complement to the density evolution. Both the variable node and check node EXIT curves can be computed in closed form [11] for the standard BP decoding. Let I_U be the average mutual information between the bits on the graph edges and the a priori (extrinsic) LLRs of the variable (check) nodes and let I_V be that between the bits on the edges and the extrinsic (a priori) LLRs of the variable (check) nodes. Then the EXIT functions of a degree$-d_v$ variable node and a degree$-d_c$ check node are respectively

$$I_{V,STD}\left(I_U, d_v, \frac{E_b}{N_0}, R\right) = J\left(\sqrt{(d_v-1)[J^{-1}(I_U)]^2 + \sigma_{ch}^2}\right) \quad (15)$$

$$I_{U,STD}\left(I_V, d_c\right) \approx 1 - J\left(\sqrt{d_c-1} \cdot J^{-1}(1-I_V)\right) \quad (16)$$

where $\sigma_{ch}^2 = 8R \cdot \frac{E_b}{N_0}$ and the functions $J(\cdot)$ and $J^{-1}(\cdot)$ are given in the Appendix of [11].

In order to find a closed form for the shuffled BP decoding, the following ideal model is built. Suppose the variable nodes can be divided into d_c sets and those in the ith set only connect to the ith edges of the check nodes. This kind of structure can be approximately obtained when the progressive edge-growth (PEG) method [12] is used to construct the Tanner graph. Since all the edges of the variable nodes in the same set connect to different check nodes, they can not benefit from one another. But they can equally make use of the updated information of the previous edges. So the processing of each check node is identical. Let the mutual information between the bits on any edges of a check node and their corresponding a priori LLRs be equal to the average input mutual information I_V. Let I'_{V_i} be the updated mutual information between the bit on the ith edge of the same check node and its a priori LLRs. Denote I_{U_i} as the mutual information between the bit on the ith edge of this check node and its extrinsic LLRs. Then the EXIT function for a check node of a (d_v, d_c) regular LDPC code in the shuffled BP decoding is

$$I_{U,SHF}\left(I_V, d_c\right) = \frac{1}{d_c}\sum_{i=1}^{d_c} I_{U_i} \quad (17)$$

$$I_{U_i} = I_{U,STD}\left(\frac{(d_c-i)I_V + \sum_{k=1}^{i-1} I'_{V_k}}{d_c-1}, d_c\right) \quad (18)$$

$$I'_{V_i} = I_{V,STD}\left(I_{U_i}, d_v, \frac{E_b}{N_0}, R\right) \quad (19)$$

Since the input mutual information of the variable nodes in different sets are different, denote them as $I_{U_1}, \ldots, I_{U_{d_c}}$, respectively. Then the average input mutual information of all the variable nodes is $I_{U_{av}} = \sum_{i=1}^{d_c} I_{U_i}/d_c$ and the average output mutual information is $I_{V_{av}} = \sum_{i=1}^{d_c} I_{V,STD}(I_{U_i}, d_v, \frac{E_b}{N_0}, R)/d_c$. The EXIT function for a variable node in the shuffled BP decoding is given by

$$I_{V_{av}} = I_{V,SHF}\left(I_{U_{av}}, d_v, \frac{E_b}{N_0}, R\right) \quad (20)$$

Let $J_1(\sigma^2) = J(\sigma)$ and $I_{U_i} = J_1(\sigma_i^2)$. Since $J_1(\sigma^2)$ is approximately linear with σ^2 when σ^2 is within a small range, we can get $I_{U_{av}} = \sum_{i=1}^{d_c} I_{U_i}/d_c = \sum_{i=1}^{d_c} J_1(\sigma_i^2)/d_c \approx J_1(\frac{1}{d_c}\sum_{i=1}^{d_c}\sigma_i^2)$. Therefore, we obtain [4]

$$I_{V,STD}\left(I_{U_{av}}, d_v, \frac{E_b}{N_0}, R\right) \approx \frac{1}{d_c}\sum_{i=1}^{d_c} I_{V,STD}\left(I_{U_i}, d_v, \frac{E_b}{N_0}, R\right)$$

$$= I_{V,SHF}\left(I_{U_{av}}, d_v, \frac{E_b}{N_0}, R\right)$$

From the simulation, we observe that the variances σ_i^2 of the a priori inputs to different variable nodes in one iteration vary within a small range. Hence the EXIT function for a variable node in the shuffled BP decoding is almost the same as that in the standard BP decoding.

It is straightforward to extend this method to replica shuffled BP. Using a similar analysis, we can prove that the EXIT function for a variable node in replica shuffled BP decoding is also almost the same as that in standard BP decoding. Since in the non-synchronous scheme subdecoders only exchange information at the end of each iteration, the EXIT function for a check node in replica shuffled BP with two subdecoders and the non-synchronous scheme can be written as

$$I_{U,REP_2,NS}\left(I_V,d_c\right)=\frac{1}{d_c}\sum_{i=d_c/2}^{d_c}2I_{U_i}\quad (even\ d_c) \tag{21}$$

$$I_{U,REP_2,NS}\left(I_V,d_c\right)=\frac{1}{d_c}\left(\sum_{i=\lceil d_c/2\rceil+1}^{d_c}2I_{U_i}+I_{U_{\lceil d_c/2\rceil}}\right)\quad (odd\ d_c) \tag{22}$$

The EXIT function for a check node in replica shuffled BP with more than two subdecoders can be obtained in a similar way. In the synchronous scheme, subdecoders exchange information immediately. Suppose *K* subdecoders are used. Then we can divide each set in the ideal model into *K* subsets. Each subdecoder processes the variable nodes in a distinct subset of the same set at the same time. After all the variable nodes are processed once, the subdecoders go back to the first set and have a cyclic shift of their positions so that they can process a subset different from what they have already processed previously. Then replica shuffled BP can be regarded as applying shuffled BP *K* times. Therefore the EXIT function for a check node in the synchronous scheme with *K* subdecoders is

$$I_{U,REP_K,S}\left(I_V,d_c\right)=I_{U,SHF}\left(I_{V_K},d_c\right) \tag{23}$$

$$I_{V_i}=I_{V,SHF}\left(I_{U,SHF}\left(I_{V_{i-1}},d_c\right),d_v,\frac{E_b}{N_0},R\right) \tag{24}$$

with $I_{V_1}=I_V$.

While these derivations allow to model the convergence of each method, the following theorem shows that the threshold value remains the same for all methods.

Theorem 1: Based on EXIT chart analysis, the threshold of a code decoded by BP is not improved by shuffled BP or replica shuffled BP.

Proof: Let the threshold in the standard BP decoding be γ. When SNR$\leq\ \gamma$, the EXIT curves of variable and check nodes cross each other at some point, saying *A*. If $I_E=I_{V,STD}\left(I_A,d_v,\frac{E_b}{N_0},R\right)$, then $I_A=I_{U,STD}\left(I_E,d_c\right)$. In (17)−(19), $I_V=I_E$, $I_{U_i}\equiv I_A$ and $I'_{V_i}\equiv I_E$. So $I_{U,SHF}\left(I_E,d_c\right)=I_A$. Since I_{U_i} is constant, $I_{V,STD}=I_{V,SHF}$ at point *A*. Then $I_E=I_{V,SHF}\left(I_A,d_v,\frac{E_b}{N_0},R\right)$. Therefore the EXIT curves of variable and check nodes in shuffled BP also cross each other at point *A* . The same result can be proved for replica shuffled BP. ■

This theorem provides a formal proof of the observations made in [13]. Indeed it was expected that the threshold derived on a tree can not be changed by modifying the scheduling of the algorithm only.

All the results above can be easily extended to other iteratively decodable codes as turbo codes. The details are included in [4].

B. Simulation results

In general the actual graph can not satisfy all the constraints of this ideal model, but its convergence behavior of the corresponding code can be well approximated by the ideal model. Fig. 3 compares the EXIT functions obtained from the method in [9] and those in closed form. Both methods assume the input LLRs have Gaussian distribution. We observe that the EXIT functions of these two methods are almost the same, which validates the EXIT functions derived in this paper.

We also verified by EXIT charts that the non-synchronous scheduling converges slower than the synchronous one, as also found in Fig. 2. Fig. 4 depicts EXIT curves superimposed to constant-BER curves [14, Chapter 9]. For the same BER, the iteration number of standard BP is twice that of shuffled BP.

Fig. 5 depicts the EXIT curves of different decoding methods at SNR 1.11dB, which is the threshold of the $(3,6)$ regular LDPC code. We observe that the EXIT curves of variable and check nodes cross each other for all the methods, so they have the same threshold as expected from Theorem 1.

Fig. 6 depicts the WER of standard and replica shuffled BP decoding of a $(16200,7200)$ irregular LDPC code which was constructed in a semi-random manner for an early version of the DVB-S2 standard [15]. The number of replica subdecoders was four. We observe that replica shuffled BP with $I_{max}=10$ and $G=32$ provides a similar performance as that of standard BP with $I_{max}=70$, as expected from the analysis presented.

REFERENCES

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

[2] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," *IEEE Trans. Commun.*, vol. 53, pp. 209-213, Feb. 2005.

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

[4] J. Zhang, Y. Wang, M. Fossorier, and J. S. Yedidia, "Iterative decoding with replica," submitted to *IEEE Trans. Inform. Theory*, Jun. 2005.

[5] J. Zhang, Y. Wang, M. Fossorier, and J. S. Yedidia, "Replica shuffled iterative decoding," *2005 IEEE Int. Symp. Inform. Theory*, to appear.

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

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

[8] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1727-1737, Oct. 2001.

[9] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," *Proc. 2002 Conf. Information Sciences and Systems*, Princeton, NJ, pp. 748-753, Mar. 2002.

[10] S. Chung, G. D. Forney, T. J. Richardson and R. Urbanke, "On the design of Low-Density Parity-Check codes within 0.0045dB of the Shannon Limit," *IEEE Commun. Lett.*, vol. 5, No. 2, Feb. 2001.

[11] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670-678, Apr. 2004.

[12] X. Hu, E. Eleftheriou, and D. Arnold, "Progressive edge-growth Tanner graphs," *IEEE Global Telecommun. Conf. 2001*, vol. 2, pp. 995-1001, Nov. 2001.

[13] S. Tong and X. Wang, "Convergence analysis of Gallager codes under different message-passing schedules," *IEEE Commun. Lett.*, vol. 9, pp. 249-251, Mar. 2005.

[14] E. Biglieri, "Coding of Wireless Channels," Springer, preprint.

[15] "Draft DVB-S2 Standard," available at *http://www.dvb.org*.

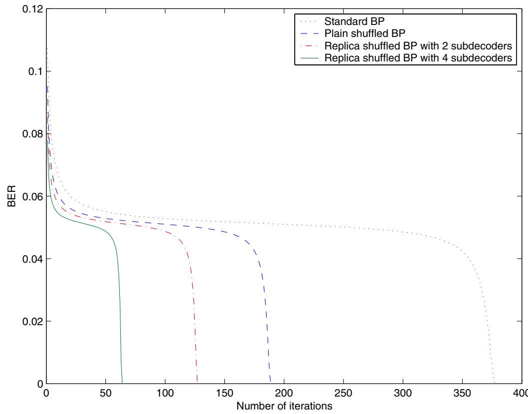

Fig. 1. BER versus number of iterations predicted by density evolution with the standard BP, plain shuffled BP, replica shuffled BP with two and four subdecoders, for decoding $(3, 6)$ regular LDPC code at SNR $1.111dB$.

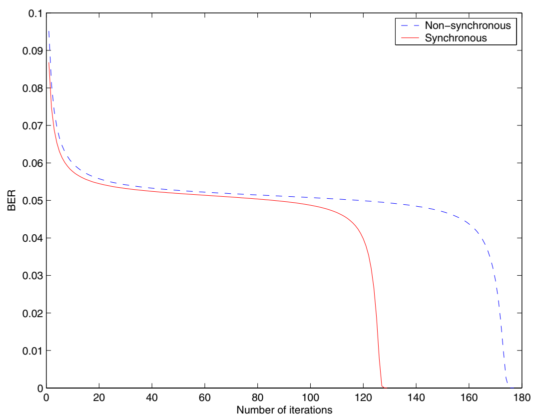

Fig. 2. BER versus number of iterations predicted by density evolution with replica shuffled BP with two subdecoders under non-synchronous and synchronous exchanging schemes, for decoding $(3, 6)$ regular LDPC code.

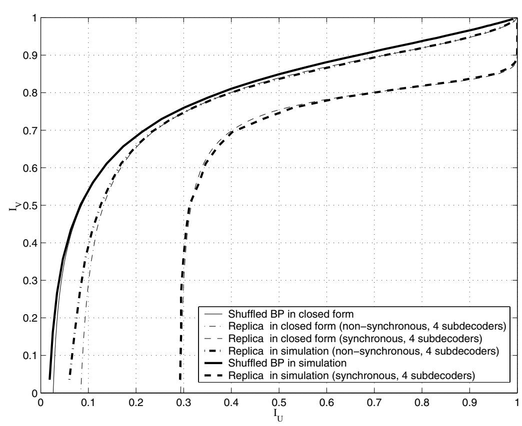

Fig. 3. The comparison between the EXIT functions obtained from the method in [9] and closed form for the $(3, 6)$ regular LDPC code at SNR 1.5dB.

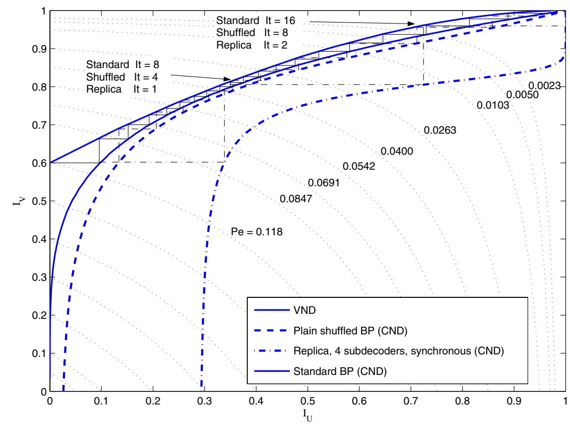

Fig. 4. EXIT curves (in closed form) for standard BP, shuffled BP and replica shuffled BP with four subdecoders and synchronous exchanging scheme at SNR 1.5dB, superimposed to constant-BER curves.

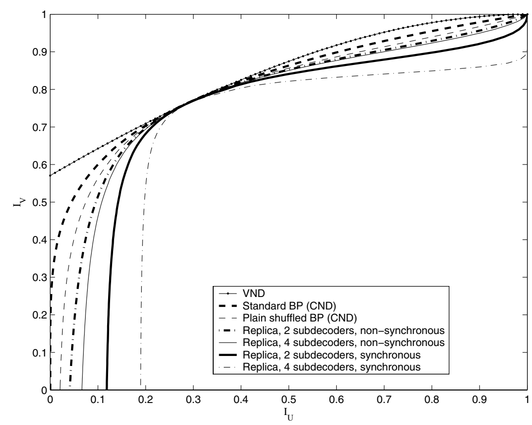

Fig. 5. EXIT curves (in closed form) for standard BP, shuffled BP and four types of replica shuffled BP at 1.11dB.

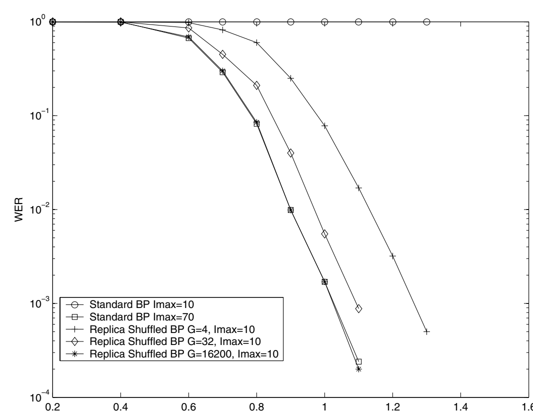

Fig. 6. Error performance for iterative decoding of a $(16200, 7200)$ irregular LDPC code.