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

Low-Latency Decoding of EG LDPC Codes

Juntan Zhang, Jonathan S. Yedidia, and Marc P.C. Fossorier

TR2008-002 January 2008

Abstract

We describe simple iterative decodes for low-density parity-check codes based on Euclidean geometries, suitable for practical very-large-scale-integration implementation in applications requiring very fast decoders. The decoders are based on shuffled and replica-shuffled versions of iterative bit-flipping (BF) and quantized weighted BF schemes. The proposed decoders converge faster and provide better ultimate performance than standard BF decoders. We present simulations that illustrate the performance versus complexity tradeoffs for these decoders. We can show in some cases through importance sampling that no significant error floor exists.

Journal of Lightwave Technology, September 2007

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

Low-Latency Decoding of EG LDPC Codes

Juntan Zhang, Jonathan S. Yedidia, and Marc P.C. Fossorier

TR-2008-002 January 2008

Abstract

We describe simple iterative decoders for low-density parity-check codes based on Euclidean geometries, suitable for practical very-large-scale-integration implementation in applications requiring very fast decoders. The decoders are based on shuffled and replica-shuffled versions of iterative bit-flipping (BF) and quantized weighted BF schemes. The proposed decoders converge faster and provide better ultimate performance than standard BF decoders. We present simulations that illustrate the performance versus complexity tradeoffs for these decoders. We can show in some cases through importance sampling that no signicant error floor exists.

Published in the Journal of Lightwave Technology, vol. 25, number 9, September 2007, pp. 2879-2886.

Low-Latency Decoding of EG LDPC Codes

Juntan Zhang, Jonathan S. Yedidia, *Member, IEEE*, and Marc P. C. Fossorier, *Fellow, IEEE*

***Abstract*—We describe simple iterative decoders for low-density parity-check codes based on Euclidean geometries, suitable for practical very-large-scale-integration implementation in applications requiring very fast decoders. The decoders are based on shuffled and replica-shuffled versions of iterative bit-flipping (BF) and quantized weighted BF schemes. The proposed decoders converge faster and provide better ultimate performance than standard BF decoders. We present simulations that illustrate the performance versus complexity tradeoffs for these decoders. We can show in some cases through importance sampling that no significant error floor exists.**

***Index Terms*—Bit-flipping (BF) decoding, low-density parity-check (LDPC) codes, optical communications, weighted BF decoding.**

I. Introduction

LOW-DENSITY parity-check (LDPC) codes were first discovered in 1960s [1] and have received significant attention recently because of their excellent performance when decoded using iterative decoders [2], [3]. LDPC codes can be constructed using random or deterministic approaches. In this paper, we focus on a class of LDPC codes known as Euclidean geometric (EG)-LDPC codes, which are constructed deterministically using the points and lines of a Euclidean geometry [4]. The EG-LDPC codes that we consider are cyclic, and consequently, their encoding can be efficiently implemented with linear shift registers. Minimum distances for EG codes are also reasonably good and can be derived analytically. Iteratively, decoded EG-LDPC codes generally do not suffer as much from the error-floor problems that plague some randomly constructed LDPC codes. For these reasons, EG-LDPC codes are good candidates for use in applications like optical communications that require very fast encoders and decoders and very low bit-error rates (BERs).

LDPC codes can be decoded using hard-decision, soft-decision, and hybrid decoding methods. Soft decoding algorithms such as belief propagation (BP) provide good performance, but require high decoding complexity, and are therefore not very suitable for very-large-scale-integration (VLSI) implementations. Instead, hard-decision and hybrid schemes such as bit flipping (BF) and quantized weighted BF (QWBF) offer a better tradeoff between error performance, complexity, and decoding speed or latency.

Most standard iterative decoders of LDPC codes require at least several tens of iterations for the iterative decoding process to converge, which is not always realistic for high-speed VLSI implementations. In [5], a decoding scheme called "shuffled BP" was presented to reduce the required number of iterations of standard BP decoding. Related BF algorithms based on the shuffled BP idea are easy to construct. Recently, an improved but more complex iterative algorithm named "replica-shuffled iterative decoding" was developed to further decrease the required number of decoding iterations [6]. In this paper, we study the performance of shuffled and replica-shuffled versions of BF and QWBF decoders for EG-LDPC codes.

Another problem for VLSI implementations of LDPC decoders is related to the fact that to achieve a good performance, the LDPC code must have a large codeword length and a correspondingly large parity-check matrix. The large parity-check matrix makes it difficult to implement the iterative decoder in a fully parallel manner. To deal with this problem, it makes sense to consider shuffled and replica-shuffled algorithms which divide the codeword bits into groups and update one group of bits at a time [5], [6].

The application of closely related LDPC codes based on projective geometries to optical communications has in fact already been proposed by Djordjevic and Vasic [7], who demonstrated the attractiveness of these codes for this application. Some of the new features present in this paper compared to the work of Djodjevic and Vasic are the study of longer codes which are better suited to the application: the use of replica shuffled BF decoders, which use simpler hardware and are more parallelized compared to the min-sum algorithm and, therefore, generally need fewer iterations, and the analysis of the error floor regime using importance sampling.

This paper is organized as follows. Section II describes the construction of EG-LDPC codes. Section III briefly reviews standard BF, weighted BF, and QWBF decoders. Shuffled and replica-shuffled versions of these decoders are presented in Sections IV and V, respectively. In Section VI, group-shuffled and replica group-shuffled schemes are discussed. Finally, in Sections VII and VIII, we provide simulation results for the various presented decoding methods.

II. Definition of EG LDPC Codes

A binary LDPC code is specified by a parity-check matrix containing mostly zeros and only a small number of ones. A regular binary $(N, K)(d_v, d_c)$ LDPC code has a transmitted codeword block length N, a information block length K, and a parity-check matrix with precisely d_v ones in each column

Manuscript received August 1, 2005; revised August 1, 2006.

J. Zhang was with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822 USA. He is now with Availink Inc., Germantown, MD 20874 USA (e-mail: jzhang@availink.com).

J. S. Yedidia is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: yedidia@merl.com).

M. P. C. Fossorier is with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822 USA (e-mail: marc@spectra.eng.hawaii.edu).

Digital Object Identifier 10.1109/JLT.2007.903311

and d_c ones in each row. We refer to the N elements of an LDPC codeword $\mathbf{w} = [w_n]$ as bits, and the M rows of the parity-check matrix $\mathbf{H} = [H_{ml}]$ as checks. Accordingly, in a regular binary LDPC code, every code bit is checked by exactly d_v parity checks, and every parity check involves exactly d_c code bits. 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 $\mathcal{M}(n) = \{m : H_{mn} = 1\}$.

EG-LDPC codes [4], [8] are regular LDPC codes characterized by a parity-check matrix which is constructed using a finite Euclidean geometry. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ be an m-tuple whose component α_i is from the Galois field GF(2^s). The set of all possible αs has cardinality 2^{ms} and forms an m-dimensional Euclidean geometry over GF(2^s), which is denoted by EG($m, 2^s$). Each m-tuple α is called a point in EG($m, 2^s$). The all-zeros point is called the origin. Let α_1 and α_2 be two linearly independent points in EG($m, 2^s$). Then, the collection of $\{\alpha_1 + \beta\alpha_2\}$, with $\beta \in$ GF(2^s), has 2^s points and forms a line (1-flat) in EG($m, 2^s$). There are $J_0 = ((2^{(m-1)s} - 1)(2^{ms} - 1))/(2^s - 1)$ lines in EG($m, 2^s$) that do not contain the origin.

To construct a cyclic EG-LDPC code based on EG($m, 2^s$), we form the parity-check matrix $\mathbf{H}_{\mathrm{EG}}$ whose columns are all of the $2^{ms} - 1$ nonorigin points in EG($m, 2^s$), and whose rows are the incidence vectors of all of the J_0 lines in EG($m, 2^s$) that do not contain the origin. Note that many of the rows of the matrix defined in this way are redundant, so that an important decision that must be made for practical decoders is how many of the redundant rows should be used, and how many should be discarded. In the decoders described in this paper, we actually used a square parity-check matrix with an equal number ($N = M = 2^{ms} - 1$) of rows and columns. Thus, we discard most of the redundant rows, but not all of them.

III. BF AND QWBF DECODING

Assume a codeword $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ is transmitted over an additive white Gaussian noise (AWGN) channel with zero mean and variance $N_0/2$ using binary phase-shift keying (BPSK) signaling, and let $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ be the corresponding received sequence.

A. Standard BF Decoding

The standard BF decoding is a hard decision algorithm. Let $\mathbf{z} = (z_1, z_2, \ldots, z_N)$ be the binary hard decision sequence obtained from $\mathbf{y}$ as follow:

$$z_n = \begin{cases} 1, & \text{if } y_n \leq 0 \\ 0, & \text{if } y_n > 0. \end{cases}$$

Let $\mathbf{s}$ be the vector of syndromes of $\mathbf{z}$

$$\mathbf{s} = (s_1, s_2, \ldots, s_M) = \mathbf{z} \cdot \mathbf{H}^{\mathrm{T}} \tag{1}$$

where

$$s_m = \sum_{n=1}^{N} z_n h_{mn}. \tag{2}$$

The received vector $\mathbf{z}$ is a codeword if and only if $\mathbf{s} = \mathbf{0}$. If $\mathbf{s} \neq \mathbf{0}$, errors are detected, and any nonzero syndrome s_m indicates a parity failure. Let F_n be the set of nonzero syndromes checking on bit n, i.e., $F_n = \{s_m : s_m = 1 \text{ and } m \in \mathcal{M}(n)\}$. In standard BF decoding, the decoder computes all the syndromes and then flips any bits which are involved in more than a fixed number δ of parity failures. Based on these new values, the syndromes are recomputed, and the process is repeated until a codeword is found or the maximum number of iterations is reached.

Thus, the standard BF decoding is carried out as follows.

Step 1) Compute $\mathbf{s} = (s_1, s_2, \ldots, s_M) = \mathbf{z} \cdot \mathbf{H}^{\mathrm{T}}$.

Step 2) For $n = 1, 2, \ldots, N$, flip z_n with $|F_n| \geq \delta$.

Step 3) Repeat Steps 1) and 2) until $\mathbf{s} = \mathbf{0}$ or the maximum number of iterations $I_{\max}$ is reached.

It should be noted that this version of BF decoding can be viewed as a simplified version of the conventional Gallager algorithm B [1]. This simplification can be justified by the large values of d_c and d_v for EG-LDPC codes.

B. WBF Decoding

The performance of standard BF decoding can be improved upon by using a soft-valued reliability measure for the received symbols. The standard weighted bit-flipping (WBF) algorithm [4] first computes for $m = 1, 2, \ldots, M$

$$|y|_{\min - m} = \min_{n: n \in \mathcal{N}(m)} |y_n|. \tag{3}$$

As explained below, $|y|_{\min - m}$ is a measure of the reliability of the mth check.

Next, WBF decoding is carried out as follows.

Step 1) For $m = 1, 2, \ldots, M$, compute the syndrome $s_m = \sum_{n=1}^{N} z_n H_{mn}$ from $\mathbf{z}$.

Step 2) For $n = 1, 2, \ldots, N$, compute

$$E_n = \sum_{m \in \mathcal{M}(n)} (2s_m - 1)|y|_{\min - m}. \tag{4}$$

Step 3) Flip the bit z_n for $n = \arg\max_{1 \leq n \leq N} E_n$.

Steps 1) to 3) are repeated until all the parity-check equations are satisfied, or until the maximum number of iterations $I_{\max}$ is reached.

WBF decoding achieves better performance than BF decoding by making more accurate decisions for each bit based on a flipping criteria that considers soft reliability information. For the AWGN channel, a simple measure of the reliability of a received symbol y_n is its magnitude $|y_n|$. The larger the magnitude $|y_n|$ is, the larger the reliability of the corresponding hard-decision digit z_n is. For $m = 1, 2, \ldots, M$, $|y|_{\min - m}$ given in (3) can be viewed as a measure of the reliability of the syndrome s_m computed with z_ns, $n \in \mathcal{N}(m)$. For $n = 1, 2, \ldots, N$, the larger E_n is, the less likely the hard decision z_n is, which is why the bits with the largest values for E_n are flipped first.

C. QWBF Decoding

From the practical point of view, WBF decoding is problematic, because a real number $|y|_{\min-m}$ must be stored for each check, these real numbers must be added to determine the reliability E_n of each bit, and the bits must then be sorted according to their values of E_n. In a more practical version of WBF decoding, which we call QWBF decoding, each bit is assigned a "high" or "low" reliability based on whether $|y_n|$ is greater or less than a preassigned threshold Δ_1. Then, for each check, if all the bits involved in that check have "high" reliability, the check is also considered to have high reliability, but if even a single bit involved in the check has "low" reliability, the check is considered to have low reliability. High-reliability checks are assigned a value of $|y|_{\min-m} = 2$, while low-reliability checks are assigned a value of $|y|_{\min-m} = 1$. Next, QWBF decoding proceeds as follows.

Step 1) For $m = 1, 2, \dots, M$, compute the syndrome $s_m = \sum_{n=1}^{N} z_n H_{mn}$ from $\mathbf{z}$.

Step 2) For $n = 1, 2, \dots, N$, compute

$$E_n = \sum_{m \in \mathcal{M}(n)} (2s_m - 1)|y|_{\min-m}. \qquad (5)$$

Step 3) Flip all bits z_n for which E_n exceeds a predefined threshold Δ_2.

Steps 1) to 3) are repeated until all the parity-check equations are satisfied, or until the maximum number of iterations $I_{\max}$ is reached.

Note that in QWBF decoding, two predetermined thresholds Δ_1 and Δ_2 have to be specified. In the simulations described below, these thresholds were chosen by empirical testing. Our chosen thresholds should be reasonably good, although we cannot be certain of how far they are from the optimal thresholds; a theoretical analysis would certainly be desirable.

Compared to WBF decoding, QWBF decoding has the advantages that only one bit needs to be stored to record the reliability of each bit and check that all the addition is simple integer arithmetic and that no sorting of the bits by reliability needs to be done. For these reasons, we consider QWBF decoding to be much more realistic than WBF decoding for our target applications.

IV. Shuffled BF and QWBF Decoding

As mentioned in Section I, standard BF decoders require an undesirably large number of iterations to converge. Shuffled BF (or shuffled QWBF) decoding is designed to accelerate the decoding process. Let us first assume, for the sake of argument, that the bits are processed serially; one bit is processed in each unit time. During a given iteration of decoding, we assume that the nth bit is processed in the nth unit of time. If the flipping condition is satisfied, this bit is flipped. Generally, the new value of z_n is more likely to be correct than the old one. Consequently, the syndromes $\{s_m : m \in \mathcal{M}(n)\}$ based on the new value of z_n are more reliable than the corresponding old ones. In shuffled BF (or QWBF) decoding, at each iteration, once a bit is flipped, all the syndromes involving in this bit are flipped too, and the processing of the remaining bits are based on these new syndrome values. Because the more reliable bit values are taken advantage of as soon as available, the shuffled version of BF (or WBF or QWBF) decoding is expected to converge faster than the standard one.

Shuffled BF is carried out as follows.

Initialization Compute $\mathbf{s} = (s_1, s_2, \dots, s_M) = \mathbf{z} \cdot \mathbf{H}^{\mathrm{T}}$.

Step 1) For $n = 1, 2, \dots, N$, if $|F_n| \geq \delta$, flip z_n and $\{s_m : m \in \mathcal{M}(n)\}$. Recalculate F_ns.

Step 2) Repeat Step 1) until $\mathbf{s} = \mathbf{0}$ or $I_{\max}$ is reached.

Similarly, the shuffled WBF is carried out as follows:

For $n = 1, 2, \dots, N$, assign each bit a value z_n, and a one-bit reliability $|y_n|$. For $m = 1, 2, \dots, M$, compute the reliability value $|y|_{\min-m}$ using the one-bit reliabilities $|y_n|$ and the threshold Δ_1.

Step 1) For $n = 1, 2, \dots, N$, compute

$$E_n = \sum_{m \in \mathcal{M}(n)} (2s_m - 1)|y|_{\min-m}. \qquad (6)$$

Step 2) For $n = 1, 2, \dots, N$, if $|E_n| \geq \Delta_2$, flip z_n and $\{s_m : m \in \mathcal{M}(n)\}$. Recalculate E_ns.

Steps 1) and 2) are repeated until all the parity-check equations are satisfied or $I_{\max}$ is reached.

V. Replica-Shuffled BF and QWBF Decoding

The plain shuffled BF (or QWBF) decoding presented in the previous section is a bit-based sequential approach, and the scheme just presented is based on a natural increasing order, i.e., flipping decisions of bits are made according to order $n = 1, 2, \dots, N$. The larger the value of n, the more newly delivered values are used in making the flipping decision and the more accurate that decision becomes. Consequently, the reliability of the bits increases and the error rate decreases with increasing n. Clearly then, in a shuffled BF (or QWBF) decoder based on a natural decreasing order, after each iteration, the reliability of the bits would decrease with increasing n.

To take advantage of this property of plain shuffled BF (or QWBF) decoders, in a replica-shuffled BF (or QWBF) decoder [6], two shuffled subdecoders ("replicas") based on different updating orders operate simultaneously and cooperatively. After each iteration, the replica subdecoders exchange bit values with each other (according to the rule that each replica is "responsible" for those bit values that it decoded later), and the next decoding iteration is based on these new values. It is also straightforward to extend replica-shuffled BF (or QWBF) decoding to cases in which more than two replica subdecoders are employed. It will require more study to determine the optimal ordering of the updates for each of the subdecoders, even for just two replicas. In general, replica-shuffled decoders obtain faster decoding at the price of duplicating subdecoders.

Let $\overrightarrow{D}$ and $\overleftarrow{D}$ be two replica subdecoders using increasing and decreasing updating orders, respectively. Let $\overrightarrow{\mathbf{s}}$, $\overrightarrow{\mathbf{z}}$, $\overrightarrow{F_n}$ be the corresponding notations associated with decoder $\overrightarrow{D}$.

Notations associated with decoder $\overleftarrow{D}$ are defined in a similar way. Replica-shuffled BF decoding with two subdecoders is carried out as follows.

```
Initialization Compute $\overrightarrow{\mathbf{s}} = \overleftarrow{\mathbf{s}} = \mathbf{z} \cdot \mathbf{H}^{\mathrm{T}}$. Let $\overrightarrow{\mathbf{z}} = \overleftarrow{\mathbf{z}} = \mathbf{z}$.
    Step 1) For $n = 1, 2, \ldots, N$, subdecoders $\overleftarrow{D}$ and $\overrightarrow{D}$ simultaneously operate, respectively, according to the following rules.
            If $|\overrightarrow{F_n}| \geq \delta$, flip $\overrightarrow{z_n}$ and $\{\overrightarrow{s_m} : m \in \mathcal{M}(n)\}$.
            If $|\overleftarrow{F}_{N-n}| \geq \delta$, flip $\overleftarrow{z}_{N-n}$ and $\{\overleftarrow{s}_m : m \in \mathcal{M}(N-n)\}$.
    Step 2) For $n = 1, 2, \ldots, N/2$, let $\overrightarrow{z}_n = \overleftarrow{z}_n$, $\overleftarrow{z}_{N-n} = \overrightarrow{z}_{N-n}$ and update $\overrightarrow{\mathbf{s}}$ and $\overleftarrow{\mathbf{s}}$.
    Step 3) Repeat Step 1) and Step 2) until $\overrightarrow{\mathbf{s}} = \mathbf{0}$ (or $\overleftarrow{\mathbf{s}} = \mathbf{0}$) or $I_{\max}$ is reached.
```

Note that Step 2) is the stage at which information is exchanged between the replicas. Each replica is responsible for those bit values that it updated more recently.

Replica shuffled QWBF decoding is carried out in an analogous way; the full description is omitted here.

VI. Group-Shuffled and Replica Group-Shuffled Schemes

An entirely serial implementation of replica-shuffled BF decoding would be desirable from the point of view of performance, but is not very realistic in terms of VLSI implementation. On the other hand, an entirely parallel implementation is not so desirable, nor even realistic given the large block-lengths that one would expect to use.

Thus, a more realistic scenario is for the bits in an EG-LDPC code to be processed in groups of bits, where the groups are processed serially, but the bits within a group are processed in parallel. We call such decoding schemes "group-shuffled" decoding or "replica group-shuffled" decoding, depending on whether replica subdecoders are used.

Assume the N bits of a codeword are divided into G groups and each group contains $(N/G) = N_G$ bits (assuming $N \bmod G = 0$ for simplicity). Grouped shuffled BF decoding is carried out as follows.

```
Initialization Compute $\mathbf{s} = (s_1, s_2, \ldots, s_M) = \mathbf{z} \cdot \mathbf{H}^{\mathrm{T}}$.
    Step 1) For $g = 1, 2, \ldots, G$
        a) process the following step in parallel: for $g \cdot N_G + 1 \leq n \leq (g+1) \cdot N_G + 1$, if $|F_n| \geq \delta$, flip $z_n$;
        b) process the following step in parallel: for $g \cdot N_G + 1 \leq n \leq (g+1) \cdot N_G + 1$, if $|F_n| \geq \delta$, flip $\{s_m : m \in \mathcal{M}(n)\}$.
    Step 2) Repeat Step 1) until $\mathbf{s} = \mathbf{0}$ or $I_{\max}$ is reached.
```

Replica group-shuffled BF (or QWBF) decoders operate in an analogous way; the full details are omitted here.

Note that in all these decoders, including the replica group-shuffled decoders, the order in which all the steps proceed is always fixed ahead of time, which is convenient for a hardware implementation. It is possible to consider other ideas, such as ordering the updates based on the strength of the channel information, but we preferred to avoid any approach that would require sorting or would interfere with the natural cyclic ordering of the bits.

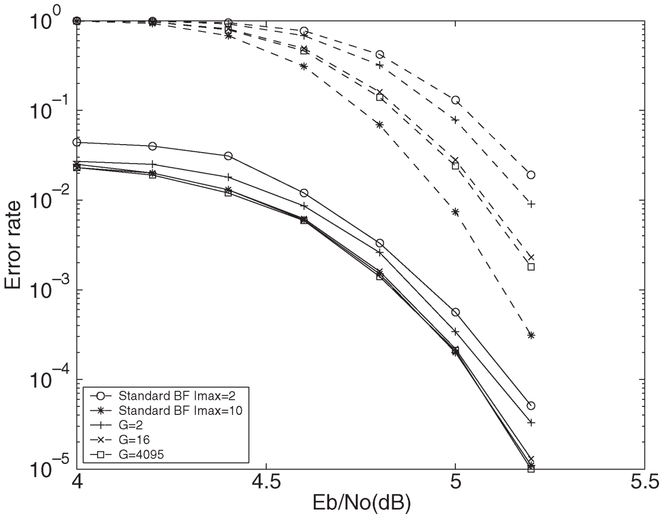

Fig. 1. Error rate of the (4095, 3367) EG-LDPC code with the standard BF and group shuffled BF algorithm, for $G = 2, 16, 4095$, and at most, two iterations.

VII. Simulation Results of Shuffled Schemes

Two-dimensional EG-LDPC codes are decoded with various schemes to show the performance of the proposed schemes. For any positive integer $s \geq 2$, the 2-D EG-LDPC code has length, dimension, and minimum distance $2^{2s} - 1$, $2^{2s} - 3^s$, and $2^s + 1$, respectively. The geometry EG$(2, 2^s)$ contains $2^{2s} - 1$ lines that do not pass through the origin. The parity-check matrix $\mathbf{H}$ therefore is a $(2^{2s} - 1) \times (2^{2s} - 1)$ square matrix.

Fig. 1 depicts the error rate of iterative decoding of the (4095, 3367) EG-LDPC code with standard BF and group shuffled BF algorithm, for $G = 2, 16, 4095$. The maximum number of iterations for group shuffled BF was set to be only two. The word error rates and bit-error rates (BERs) are shown simultaneously in many of our plots; obviously, the bit-error rates are the lower curves.

Note in Fig. 1 that for group-shuffled BF decoding, using 16 groups is nearly as good as 4095 (fully serial operation). Note also that the word error rate performance using 16 groups and two iterations in group-shuffled decoding is nearly as good as that using ten iterations in standard BF decoding.

Fig. 2 depicts the error rate of iterative decoding of the (4095, 3367) EG-LDPC code with standard QWBF and group shuffled QWBF algorithm, for $G = 2, 16, 4095$. The maximum number of iterations for group shuffled BF was set to be five.

The threshold parameters used were $\Delta_1 = 0.09$ and $\Delta_2 = 8.0$. There is clearly a performance gain from using QWBF instead of BF (about 0.5 dB at a BER of 10^{-5}).

Fig. 3 depicts the error rate of iterative decoding of the (16 383, 14 179) EG-LDPC code with the standard BF and the group shuffled BF algorithm, for $G = 2, 16, 4095$. The maximum number of iterations for group shuffled BF was set to be 2.

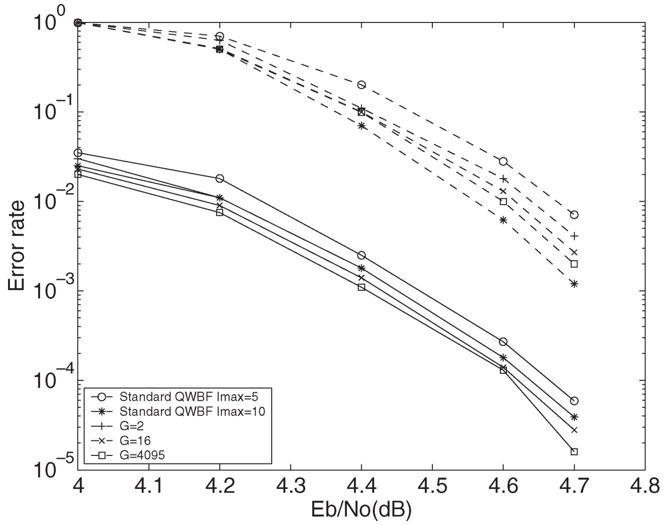

Fig. 2. Error rate of the (4095, 3367) EG-LDPC code with the standard QWBF and group shuffled QWBF algorithm, for $G = 2, 16, 4095$, and at most, five iterations.

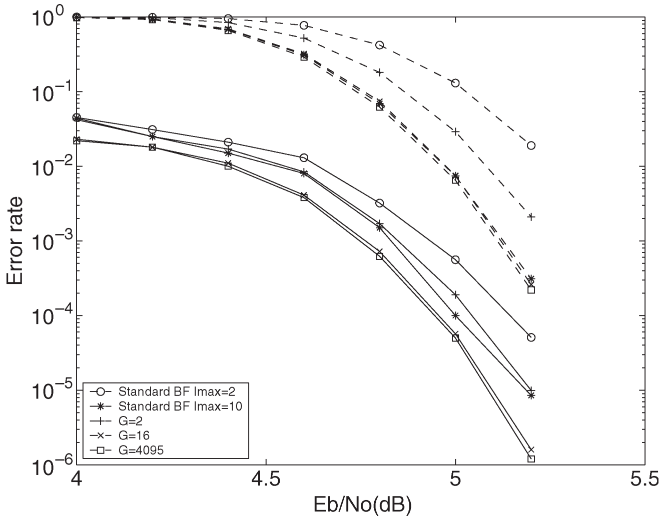

Fig. 4. Error rate of the (4095, 3367) EG-LDPC code with the standard BF and group replica shuffled BF algorithm with four subdecoders, for $G = 2, 16, 4095$, and at most two iterations.

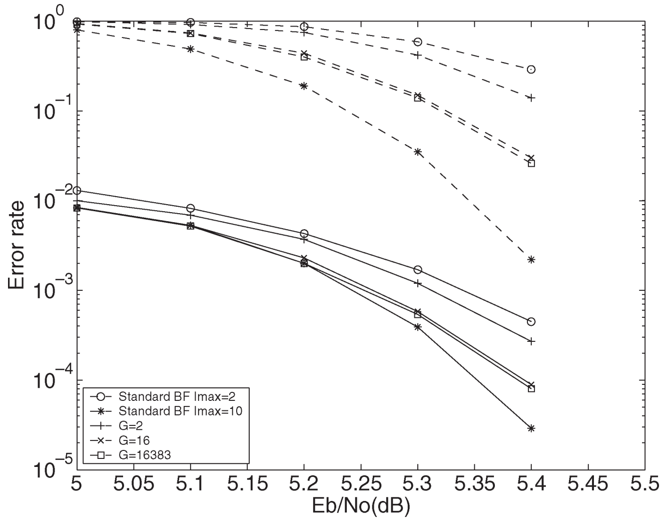

Fig. 3. Error rate of the (16 383, 14 179) EG-LDPC code with the standard BF and the group shuffled BF algorithm, for $G = 2, 16, 4095$, and at most two iterations.

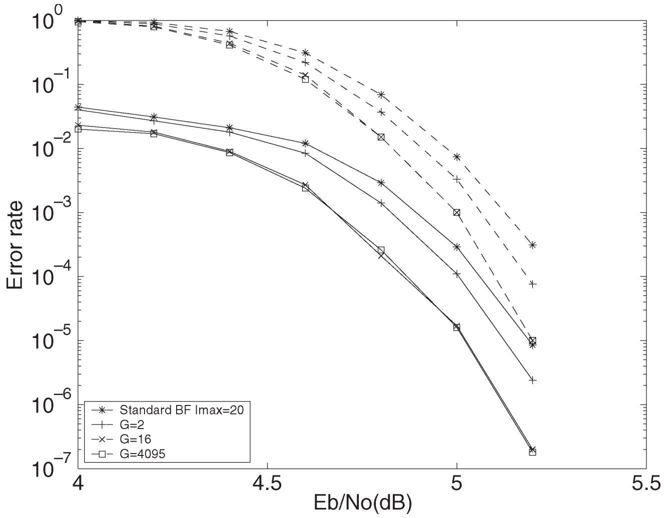

Fig. 5. Error rate of the (4095, 3367) EG-LDPC code with the standard BF and group replica shuffled BF algorithm with four subdecoders, for $G = 2, 16, 4095$, and at most, ten iterations.

VIII. Simulation Results of Replica Shuffled Schemes

Fig. 4 depicts the error rate of iterative decoding of the (4095, 3367) EG-LDPC code with standard BF and group replica shuffled BF algorithm with four subdecoders, for $G = 2, 16, 4095$. The maximum number of iterations $I_{\max}$ for group replica shuffled BF was set to be two. We observed that the word error rate (WER) performance of group replica shuffled BF decoding with four subdecoders and $I_{\max} = 2$, and group number, larger or equal to four, was approximately the same as that of the standard BF with $I_{\max} = 10$, while the BER performance is even better.

Fig. 5 depicts the error rate of the same code decoded by the standard and replica shuffled BF methods, with $I_{\max} = 20$ and $I_{\max} = 10$, respectively. The point of this figure is more theoretical than practical; to demonstrate that replica-group shuffled BF decoding outperforms BF decoding when $I_{\max}$ is large enough.

An important issue in optical communications systems and in storage systems is the performance at very low error rates. The question that must be answered is whether there is a hidden error floor in the high signal-to-noise ratio (SNR) regime. This question is difficult to answer through simulations, but we can make some progress, and set worst case bounds on the error floor, by using importance sampling for the case of BF decoding.

To obtain performance for very low error rates, we generated random errors of fixed weight n, and for each weight n, we evaluated the corresponding bit-error performance $P(n)$. Fortunately, we can assume that no errors at all will occur if $n \leq t$, where t is the bounded error correcting capability of the

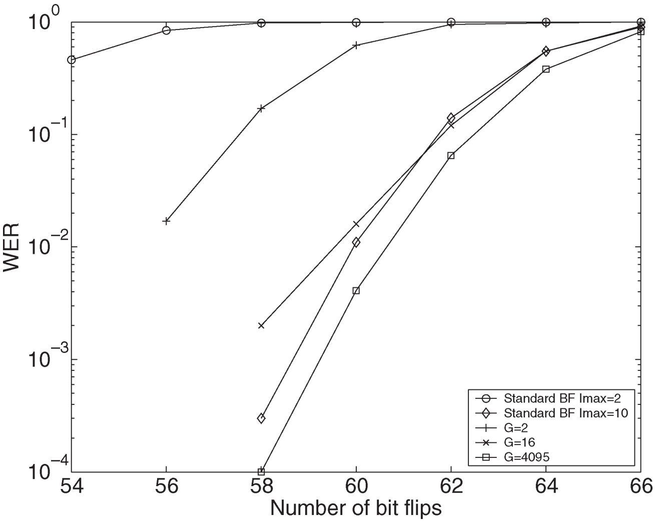

Fig. 6. WER of the (4095, 3367) EG-LDPC code with the standard BF and group replica shuffled BF algorithm with four subdecoders, for $G = 2, 16, 4095$, and at most, two iterations for fixed number of errors.

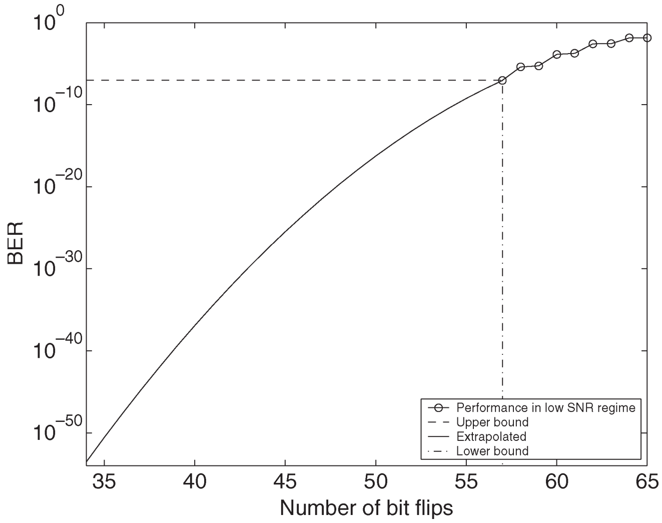

Fig. 7. Group replica shuffled BF decoding of the (4095, 3367) EG-LDPC code for fixed number of errors with four subdecoders and $G = 16$, $I_{\max} = 2$. The point of this figure is to show the extrapolations used in constructing the worst case (upper bound), best case (lower bound), and likely case (extrapolated) performance curves at very low error rates.

code $t = \lfloor (d_{\min} - 1)/2 \rfloor$ (this corresponds to the reasonable assumption that the decoder has a built-in low-complexity bounded-distance decoder; such decoders were developed for these codes already in the 1960s [8]). For the (4095, 3367) code, $d_{\min} = 65$, and $t = 32$.

The overall bit-error performance P_s was then obtained by the average

$$P_s = \sum_{n=t+1}^{N} P(n) \binom{N}{n} p_e^n (1 - p_e)^{N-n}. \qquad (7)$$

For BPSK signaling over AWGN channel, the transition probability $p_e = Q(\sqrt{RE_b/N_0})$, where R is the code rate and E_b/N_0 is the SNR per information bit.

Fig. 6 depicts the error performance of the standard and replica group-shuffled BF decoding methods with four subdecoders for decoding the (4095, 3367) EG-LDPC code with a fixed number of errors. The maximum number of iterations for replica shuffled BF was set to 2.

Using the results from Fig. 6, we can determine worst case, best case, and extrapolated performances for this decoder down to very low error rates.

For word error rates (WERs) smaller than 10^{-4} (BER smaller than 10^{-7}), no reliable evaluation of $P(n)$ was possible, so we computed: 1) a worst case upper bound on (7) by assuming the same $P(n_{\min})$ as the smallest simulated for weights n', $t < n' < n_{\min}$, 2) a best case lower bound on (7) by assuming $P(n) = 0$ for weights n', $t < n' < n_{\min}$, and 3) a likely case approximation by extrapolating $P(n')$ for weights n', $t < n' < n_{\min}$.

Fig. 7 depicts these extrapolations. The worst case upper bound is derived using the horizontal line in this figure, the best case lower bound is derived using the vertical line, and the likely case extrapolation is derived using the curved line.

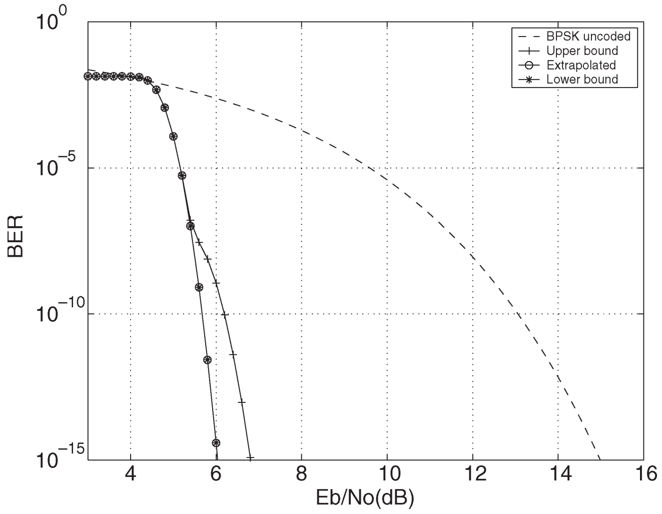

Fig. 8. Group replica shuffled BF decoding of the (4095, 3367) EG-LDPC code in high SNR regime with four subdecoders and $G = 16$, $I_{\max} = 2$. Note that the worst case performance is never more than 1 dB worse than the likely case performance, indicating that there is no error floor.

Fig. 8 depicts the performance of the replica shuffled BF decoding for the (4095, 3367) EG-LDPC codes in the high SNR regime based on Fig. 7. Note that the best case and likely case scenarios are nearly identical. More importantly, the worst case scenario is only slightly worse (less than 1 dB) than the likely case scenario for BERs between 10^{-10} and 10^{-15}. This means that we can be confident that there will be no significant error floor using this decoding method.

Fig. 9 depicts the error rate of iterative decoding of the (16 383, 14 179) EG-LDPC code with standard BF and group replica shuffled BF algorithm with four subdecoders, for $G = 2, 16, 16\,383$. The maximum number of iterations for group replica shuffled BF was set to be two. We observe that the WER

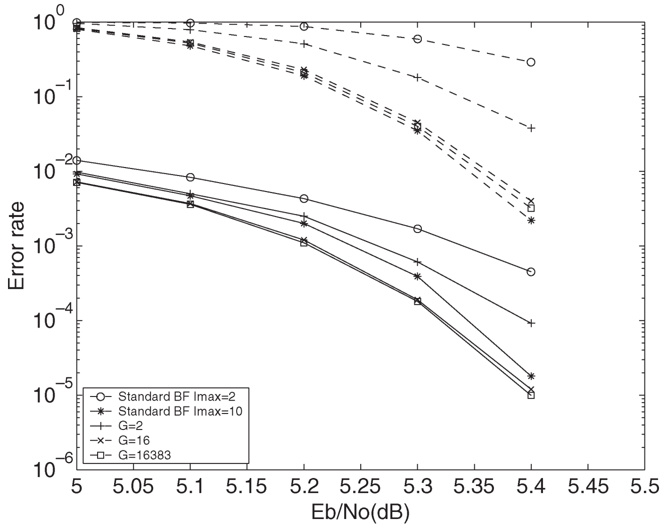

Fig. 9. Error rate of the (16 383, 14 179) EG-LDPC code with the standard BF and group replica shuffled BF algorithm with four subdecoders, for $G = 2, 16, 16\,383$, and at most, two iterations.

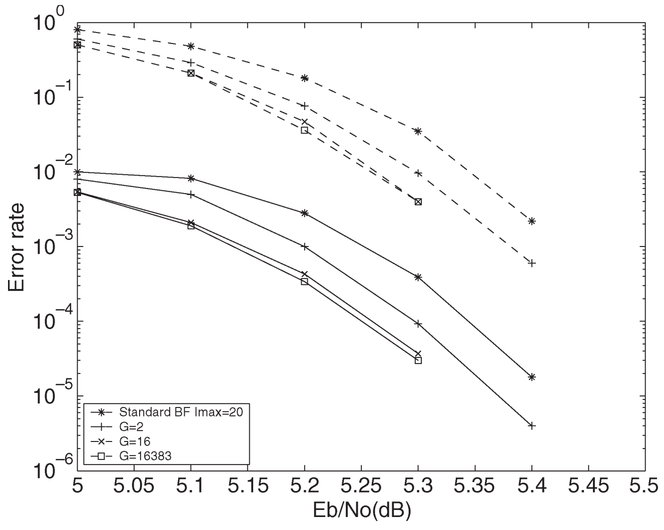

Fig. 10. Error rate of the (16 383, 14 179) EG-LDPC code with the standard BF and group replica shuffled BF algorithm with four subdecoders, for $G = 2, 16, 18\,383$, and at most ten iterations.

performance of group replica shuffled BF decoding with four subdecoders and maximum number of two, and group number larger or equal to 16, are approximately the same as that of the standard BF with maximum number of ten.

Fig. 10 depicts the error rate of the same code decoded by the standard and replica shuffled BF methods, with $I_{\max} = 20$ and $I_{\max} = 10$, respectively. With respect to the ultimate performance after full convergence, the replica shuffled BF decoding is observed to outperform the standard BF. This can be partly explained by less error propagation involved in replica shuffled method.

Although not presented here, other geometry LDPC codes were also simulated with the standard and the replica decoding schemes. Based on the simulation results, replica decoding shows similar improvements in the required number of decoding iterations for codes with different codeword lengths, either shorter or longer than those shown here.

IX. Conclusion

We have described many different decoders, and it may be worthwhile to give some final pointers to orient the reader. The major conclusion is that group-shuffled BF decoders and replica group-shuffled BF or QWBF decoders give good performance using a very small number (two) of iterations and a relatively small number (16, but in fact four is often sufficient) of groups, comparable or better than that of standard decoders using ten or more iterations. This reduction in the reduction of the number of iterations will translate into a better decoding throughput and/or latency. The gain obtained by upgrading from BF to QWBF is rather large (probably about 0.5 dB), but one must be able to quantize the channel output using one bit (“high” versus “low” reliablility. Replica group-shuffled decoders using four replicas also have a further performance advantage compared to ordinary group-shuffled decoders, but the gain is not very large. Finally, we have demonstrated that group-shuffled and replica-group shuffled BF decoders will not have a significant error floor. Although it would be harder to demonstrate using importance sampling, there is no reason to expect error floors for QWBF decoders either.

References

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

[2] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

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

[4] Y. Kou, S. Lin, and M. Fossorier, “Low density parity check codes based on finite geometries: A rediscovery and more,” *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[5] J. Zhang and M. Fossorier, “Shuffled belief propagation decoding,” in *Proc. 36th Annu. Asilomar Conf. Signals, Syst. and Comput.*, Nov. 2002, pp. 8–15.

[6] J. Zhang, Y. Wang, M. Fossorier, and J. Yedidia, “Replica shuffled iterative decoding,” MERL, Cambridge, MA, MERL Tech. Rep. 2005-063, Dec. 2004.

[7] I. Djordjevic and B. Vasic, “Projective geometry low-density parity-check codes for ultra-long haul WDM high-speed transmission,” *IEEE Photon. Technol. Lett.*, vol. 15, no. 5, pp. 784–786, May 2003.

[8] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.

Juntan Zhang received the Ph.D. degree in electrical engineering from the University of Hawaii, Honolulu.

He is currently with Availink, Inc., Germantown, MD.

Jonathan S. Yedidia (M’01) received the Ph.D. degree in physics from Princeton University, Princeton, NJ, in 1990. His graduate work and his postdoctoral work focused on the statistical mechanics of disordered systems.

During 1990–1993, he was a Junior Fellow with Harvard’s Society of Fellows, Cambridge. From 1993 to 1997, he was a Teacher. He is currently a Research Scientist with Mitsubishi Electric Research Laboratories (MERL), Cambridge. His main research interests since joining MERL in 1998 have been in algorithms for probabilistic inference and their applications in communications, signal processing, and artificial intelligence.

Marc P. C. Fossorier (S'90–M'95–SM'00–F'06) was born in Annemasse, France, on March 8, 1964. He received the B.E. degree from the National Institute of Applied Sciences, Lyon, France, in 1987 and the M.S. and Ph.D. degrees from the University of Hawaii, Honolulu, in 1991 and 1994, respectively, all in electrical engineering.

He is a Professor with the Department of Electrical Engineering of the University of Hawaii. His research interests include decoding techniques for linear codes, communication algorithms, combining coding and equalization for ISI channels, magnetic recording, and statistics. He coauthored (with S. Lin, T. Kasami, and T. Fujiwara) the book *Trellises and Trellis-Based Decoding Algorithms* (New York: Kluwer Academic Publishers, 1998).

Dr. Fossorier is a recipient of a 1998 National Science Foundation Career Development award. He has been serving as Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS since 1996, as Associate Editor for the IEEE COMMUNICATIONS LETTERS since 1999, and is currently the First Vice President of the IEEE Information Theory Society. He was Program Cochairman for the 2000 International Symposium on Information Theory and its Applications and Editor for the *Proceedings of the 2003 and 1999 Symposium on Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC)*. He is a member of the IEEE Information Theory and IEEE Communications Societies.