

New Group Shuffled BP Decoding Algorithms for LDPC Codes

Chi-Yuan Chang, Yu-Liang Chen, Chang-Ming Lee, and Yu T. Su

Department of Communications Engineering

National Chiao Tung University

Hsinchu, 30056, TAIWAN

Emails: tofar.cm96g@nctu.edu.tw, tttt1415@gmail.com, cmlee.cm89g@nctu.edu.tw, ytsu@mail.nctu.edu.tw

Abstract—Implementing a belief propagation (BP) based LDPC decoder requires high degrees of parallelism using many component soft-in soft-output (SISO) decoding units to perform message passing from variable nodes to check nodes or vice versa. An obvious complexity-reduction solution is to serialize the decoding process, i.e., dividing a decoding iteration into several serial sub-iterations in which a sub-iteration performs only part of the complete parallel message-passing operation. The group horizontal shuffled BP (GHSBP) and vertical shuffled BP (GVSBP) algorithms respectively partition the check and variable nodes of the code graph into groups to perform group-by-group message-passing decoding. This paper proposes new techniques to improve three key elements of a GHSBP decoding algorithm, namely, the grouping method, the decoding schedule and the log-likelihood updating formulae. The (check nodes) grouping method and decoding schedule optimize certain design criterion. The new normalized min-sum updating formula with a self-adjustable correction (scaling) factor offers better nonlinear approximation. Numerical performance of new GHSBP algorithms that include part or all three new techniques indicate that the combination of the proposed grouping and decoding schedule yields a faster convergence rate and our modified min-sum algorithm gives performance superior to that of the conventional min-sum and normalized min-sum algorithm and is very close to that of the sum-product algorithm.

I. INTRODUCTION

The belief propagation (BP) [4] or sum-product algorithm (SPA) is an efficient inference algorithm on trees and has demonstrated empirical success in numerous applications that involve loopy networks including LDPC codes and turbo codes. The algorithm is naturally suited for parallel processing but, except for short codes, hardware implementation of a parallel decoder requires large memory, high computational complexity and complicated interconnection. A more practical alternative is to partition either the variable nodes or the check nodes of the corresponding bipartite code graph into several groups and perform group-wise parallel decoding in a serial manner. Depending on which class of nodes are partitioned, these two parallel-serial approaches are referred to as the (group) horizontal shuffled BP (GHSBP) [10] [11] and the (group) vertical shuffled BP (GVSBP) algorithms [6]-[9], respectively. The group shuffled BP algorithms have the advantage of passing more reliable messages earlier for subsequent iterative decoding. As a result, they reduce the number of iterations needed to achieve a predetermined performance.

The min-sum algorithm (MSA) [12], which replaces the nonlinear check node operation,

$-\ln \tanh [\sum_i -\ln \tanh (|x_i|/2) / 2]$, by a single minimum operation, was introduced to reduce the complexity of the standard BP algorithm at the cost of a noticeable degradation in the decoding performance. Some modifications are therefore proposed [13]-[15]. The normalized min-sum algorithm (NMSA) and the offset min-sum algorithm multiplies and adds a constant correction factor in the check-to-variable updating of MSA. They offer performance near that of the standard BP algorithm but with lower complexity.

In this paper we introduce a partition-dependent parameter that measures the average degree of variable nodes in a group and present a new grouping method which has better average connection measure, i.e., it results in an increase of the average number of check nodes connected to a variable node in a group. This is achieved by allowing a check node to be in different groups whereas for the conventional grouping is a partition such that each check node belongs to one group only. An associated (group) decoding schedule that makes each subiteration to use most recently updated messages as soon as possible is proposed as well. The combination of grouping and scheduling enables our algorithm to outperform the standard BP and GHSBP algorithm with the same computation complexity.

Besides new grouping and scheduling methods, we also propose a modified version of the normalized min-sum algorithm (M-NMSA) that takes into account the effect of the magnitude summation of variable-to-check messages on the nonlinear function mentioned above. The new approximation induces a self-adjustable correction factor in the corresponding node updating formula.

The paper is organized as follows. In Section II, we present some properties of the LDPC codes. In Section III, the new grouping method and the corresponding decoding schedule are presented. In Section IV, we describe our modified normalized min-sum algorithm and Section V provides some simulated numerical results about the performance of the proposed algorithms. Finally, conclusion remarks are drawn in Section VI.

II. SOME PROPERTIES OF THE LDPC CODES

A regular binary (N, K) (d_v, d_c) LDPC code \mathcal{C} is a linear block code described by an $M \times N$ parity check matrix \mathbf{H} which has d_v ones in each column and d_c ones in each

row. \mathbf{H} can be viewed as a bipartite graph with N variable nodes corresponding to the encoded bits, and M check nodes corresponding to the parity-check functions represented by the rows of \mathbf{H} . The code rate of \mathcal{C} is given by $\mathcal{R} = K/N$.

Let $\mathcal{N}(m)$ be the set of variable nodes that participate in check node m and $\mathcal{M}(n)$ be the set of check nodes that are connected to variable node n in the code graph. $\mathcal{N}(m) \setminus n$ is defined as the set $\mathcal{N}(m)$ with the variable node n excluded while $\mathcal{M}(n) \setminus m$ is the set $\mathcal{M}(n)$ with the check node m excluded. We further define $\mathcal{W}(m)$ as the union of $\mathcal{M}(n^*)$, where $n^* \in \mathcal{N}(m)$, i.e., $\mathcal{W}(m) = \bigcup_{n^* \in \mathcal{N}(m)} \mathcal{M}(n^*)$, and similarly, we define $\mathcal{U}(m)$ as the set of variable nodes that participate in the check nodes in $\mathcal{W}(m)$, i.e., $\mathcal{U}(m) = \bigcup_{m^* \in \mathcal{W}(m)} \mathcal{N}(m^*)$. The following four properties on the cardinalities of $\mathcal{W}(m)$ and $\mathcal{U}(m)$ associated with a regular binary (N, K) (d_v, d_c) LDPC code \mathcal{C} are needed in our subsequent discourse.

Property 1: For any check node m of \mathcal{C} , $|\mathcal{W}(m)| \leq (d_v - 1) \cdot d_c + 1$.

Property 2: If m is a check node of \mathcal{C} which is not involved in any cycle of length 4, then $|\mathcal{W}(m)| = (d_v - 1) \cdot d_c + 1$.

Using the above two properties, we further obtain

Property 3: If the girth of \mathcal{C} is greater than 4, then $|\mathcal{W}(m)| = (d_v - 1) \cdot d_c + 1$ for every check node m .

and

Property 4: For every check node m of \mathcal{C} , $|\mathcal{U}(m)| \leq (d_v - 1) \cdot (d_c^2 - d_c) + d_c$.

III. GROUPING CHECK NODES AND DECODING SCHEDULE

The conventional GHSBP and HSBP algorithms partition the check nodes into groups of equal cardinalities according to their natural order, that is, M checks of a codeword are divided into G groups, and each group contains $M/G = N_c$ checks (assuming $G = 0 \bmod N$ for simplicity). Therefore, check nodes iN_c to $(i+1)N_c - 1$ belong to the i th group, $i = 0, 1, \dots, G-1$.

Let $\mathcal{G}_i(\mathcal{P})$ be the i th group of check nodes resulted from a given partition \mathcal{P} , then the uniform partition of the conventional GHSBP algorithm \mathcal{P}_{con} yields $\mathcal{G}_i(\mathcal{P}_{con}) = \{m | (iN_c \leq m \leq (i+1)N_c - 1)\}$. Although for an uniform partition, each check node belongs to a group only, and the variable nodes are likely to be connected to check nodes in more than one $\mathcal{G}_i(\mathcal{P}_{con})$.

When $G = 1$, the GHSBP algorithm becomes the standard BP algorithm, and if $G = M$, the GHSBP algorithm becomes the horizontal shuffled BP (HSBP) algorithm. The HSBP algorithm allows the more reliable (most recently updated) messages to be used as soon as they become available at the cost of a larger decoding delay due to its fully sequential nature. The GHSBP algorithm reduces the decoding delay with a parallel-serial message-passing approach but may lead to performance inferior to that of the HSBP algorithm.

We denote the set of variable nodes that participate the check nodes in $\mathcal{G}_i(\mathcal{P})$ by $\mathcal{N}_{\mathcal{G}_i}$, i.e., $\mathcal{N}_{\mathcal{G}_i} = \bigcup_{m^* \in \mathcal{G}_i(\mathcal{P})} \mathcal{N}(m^*)$, where we have omitted the dependence of the sets $\mathcal{N}_{\mathcal{G}_i}$ on the partition \mathcal{P} for the sake of notational brevity. The ratio $\frac{N_c \cdot d_c}{|\mathcal{N}_{\mathcal{G}_i}|}$ represents the average number of check nodes connected

TABLE I

MACKAY (504,252) REGULAR CODE, $d_v = 3, d_c = 6$

| Grouping Method | E_G |
|-------------------------|--------|
| New $N_c = 13$ | 1.2187 |
| Conventional $N_c = 12$ | 1.0368 |
| Random $N_c = 12$ | 1.0438 |
| Random $N_c = 13$ | 1.0494 |
| Random $N_c = 14$ | 1.0545 |

TABLE II

MACKAY (816,408) REGULAR CODE, $d_v = 3, d_c = 6$

| Grouping Method | E_G |
|-------------------------|----------|
| New $N_c = 13$ | 1.1996 |
| Conventional $N_c = 12$ | 1.0021 |
| Random $N_c = 12$ | 1.02615 |
| Random $N_c = 13$ | 1.02973. |
| Random $N_c = 14$ | 1.03209. |

to a variable node in $\mathcal{G}_i(\mathcal{P})$. Hence, a larger $\frac{N_c \cdot d_c}{|\mathcal{N}_{\mathcal{G}_i}|}$ means, on the average, the variable nodes in $\mathcal{N}_{\mathcal{G}_i}$ receive messages from more check nodes and more reliable message will be forwarded from the variable nodes in $\mathcal{N}_{\mathcal{G}_i}$. Define the average check number (ACN) for a given partition \mathcal{P} as

$$E_G(\mathcal{P}) = \frac{1}{G} \sum_{i=0}^{G-1} \frac{N_c \cdot d_c}{|\mathcal{N}_{\mathcal{G}_i}|} \quad (1)$$

A. A New Grouping Method

Let \mathcal{P}_{new} be the partition¹(grouping) such that

$$\mathcal{G}_m(\mathcal{P}_{new}) = \mathcal{W}(m), \quad 0 \leq m \leq M-1 \quad (2)$$

Such a grouping method results in M groups with duplicate member nodes, i.e., a check node may belong to several groups.

Property 3 immediately tells us that, for the above grouping method, $N_c = (d_v - 1)d_c + 1$ if the girth of the code is larger than 4. Table I and Table II list the ACN's ($E_G(\mathcal{P})$) for different grouping (partition) methods where the random grouping method refers to the method that select N_c check nodes randomly for each group. The group sizes N_c of other methods are chosen to be close to $(d_v - 1) \cdot d_c + 1$ for fair comparison. The results indicate that the new grouping method offer ACN's larger than other grouping methods. This is partially due to fact that for the proposed grouping method $|\mathcal{N}_{\mathcal{G}_i}| = |\mathcal{U}(m)|$ is upper-bounded (see **Property 4**) while for other methods $|\mathcal{N}_{\mathcal{G}_i}|$ may be larger than $(d_v - 1)(d_c^2 - d_c) + d_c$.

B. A New Decoding Schedule

Having determined the grouping of check nodes, we now consider the associated group-decoding schedule. In order

¹Such a "partition" is not really a partition anymore, whence to avoid abusing the term we will refer to it as grouping henceforth.

that more reliable messages be used as soon as they become available, we schedule the group message-passing order in an iteration by $O(0) \rightarrow O(1) \rightarrow \dots \rightarrow O(M-1)$, where $O(j)$ defined by (3) is the j th group to be processed. That is, having selected the $O(j-1)$ th group to be processed, we assign the one which has not been processed and whose connecting variable nodes set $\mathcal{U}(O(j))$ has the largest intersection with $\mathcal{U}(O(j-1))$ as the next group to be processed. Note that the indexes for the check node groups induced by a grouping method say, (2), and those associated with a decoding schedule is very likely to be different.

Assume a codeword $\mathbf{C} = [c_0, c_1, \dots, c_{N-1}]$ is BPSK-modulated and transmitted over an AWGN channel with noise variance σ^2 . Let $\mathbf{R} = [r_0, r_1, \dots, r_{N-1}]$ be the corresponding received sequence and $L(n)$ be the log-likelihood ratio (LLR) of the variable node n with the initial value given by $L(n) = \frac{2}{\sigma^2} r_n$. $\varepsilon_{m \rightarrow n}$ denotes the check-to-variable message from check node m to variable node n and $z_{n \rightarrow m}$ denotes the variable-to-check message from variable node n and to check node m . $z_n^{(i)}$ represents the a posteriori LLR of the variable node n at the i th iteration. The proposed algorithm with the new grouping method and new scheduling of groups is carried out as follows.

Initialization: Let $\psi = \{l | 1 \leq l \leq M-1\}$, $O(l) = 0$ for $0 \leq l \leq M-1$. Set $i = 1$ and maximum number of iterations to I_{Max} . For each m, n , set $z_{n \rightarrow m} = L(n)$.

Step 1: For $j = 1 : M-1$

$$\begin{aligned} O(j) &= \arg \max_{l \in \psi} (|\mathcal{U}(O(j-1)) \cap \mathcal{U}(l)|), \\ \psi &\leftarrow \psi \setminus O(j) \end{aligned} \quad (3)$$

Step 2: For $k = 0 : M-1$

a) Horizontal Step: $\forall m \in \mathcal{W}(O(k)), n \in \mathcal{U}(O(k))$

$$\begin{aligned} \varepsilon_{m \rightarrow n} &= \left\{ \prod_{n' \in \mathcal{N}(m) \setminus n} \text{sgn}(z_{n' \rightarrow m}) \right\} \times \\ &\quad \phi \left\{ \sum_{n' \in \mathcal{N}(m) \setminus n} \phi(|z_{n' \rightarrow m}|) \right\} \end{aligned} \quad (4)$$

where $\text{sgn}(x)$ is equal to the sign of the argument x and

$$\phi(x) = \phi^{-1}(x) = -\ln \tanh(x/2) \quad (5)$$

b) Vertical Step: $\forall n \in \mathcal{U}(O(k+1))$ if $k < M-1$ or $n \in \mathcal{U}(O(0))$ if $k = M-1$, and each $m \in \mathcal{M}(n)$

$$z_{n \rightarrow m} = L(n) + \sum_{m' \in \mathcal{M}(n) \setminus m} \varepsilon_{m' \rightarrow n} \quad (6)$$

Step 3: $\forall 0 \leq n \leq N-1$,

$$z_n^{(i)} = L(n) + \sum_{m' \in \mathcal{M}(n)} \varepsilon_{m' \rightarrow n} \quad (7)$$

TABLE III

COMPARISON OF NORMALIZED UPDATING COMPLEXITIES

| Method | Horizontal Updating | Vertical Updating |
|--------|--|--|
| BP | M | N |
| GHSBP | M | N |
| New | $\leq M \cdot (d_v - 1) \cdot d_c + M$ | $\leq N \cdot (d_v - 1) \cdot d_c + N$ |

Step 4: Hard decision and stopping criterion test:

- Create $\mathbf{D}^{(i)} = [d_0^{(i)}, d_1^{(i)}, \dots, d_{N-1}^{(i)}]$ such that $d_n^{(i)} = 0$ if $z_n^{(i)} \geq 0$ and $d_n^{(i)} = 1$ if $z_n^{(i)} < 0$.
- If $\mathbf{D}^{(i)} \mathbf{H}^T = \mathbf{0}$ or I_{Max} is reached, stop decoding and output $\mathbf{D}^{(i)}$ as the decoded codeword. Otherwise, set $i = i + 1$ and go to **Step 2**.

C. Computation Complexity

For the proposed algorithm, each decoding iteration consists of M subiterations and the number of check nodes in a group depends on d_c , d_v , and the girth of the code. The GHSBP algorithm requires $\lfloor M/N_c \rfloor \leq M$ subiterations with no overlapping check node updating in an iteration while the BP algorithm processes all node updates in parallel.

Table III shows the numbers of normalized (by the degrees of the nodes of concern) updates per iteration for different algorithms. When the girth of \mathcal{C} is larger than 4, the inequalities in Table III will become equalities. Although the proposed algorithm has higher per-iteration computation complexity because of the overlapped grouping, its convergence speed and performance, as will be shown in Section V, are superior to those of the other two algorithms such that it outperforms BP and GHSBP with identical computation complexity constraint.

IV. MODIFICATIONS OF MIN-SUM ALGORITHM

A. Min-Sum Algorithm and Normalized Min-Sum Algorithm

For conventional BP or SPA, the horizontal updating, (4), (5), involves some nonlinear operations which can be simplified by approximating the composite nonlinearity in the horizontal step. The min-sum algorithm (MSA) is perhaps the most common simplification which replaces (4) by

$$\begin{aligned} \varepsilon_{m \rightarrow n} &= \left\{ \prod_{n' \in \mathcal{N}(m) \setminus n} \text{sgn}(z_{n' \rightarrow m}) \right\} \\ &\quad \times \min_{n' \in \mathcal{N}(m) \setminus n} |z_{n' \rightarrow m}|. \end{aligned} \quad (8)$$

The approximation

$$\phi \left\{ \sum_{n' \in \mathcal{N}(m) \setminus n} \phi(|z_{n' \rightarrow m}|) \right\} \approx \min_{n' \in \mathcal{N}(m) \setminus n} |z_{n' \rightarrow m}| \quad (9)$$

used in (8) brings about some performance degradation, hence several improved approximations were suggested. The normal-

ized min-sum algorithm (NMSA) scales (8) by a factor α :

$$\varepsilon_{m \rightarrow n} = \alpha \left\{ \prod_{n' \in \mathcal{N}(m) \setminus n} \text{sgn}(z_{n' \rightarrow m}) \right\} \times \min_{n' \in \mathcal{N}(m) \setminus n} |z_{n' \rightarrow m}| \quad (10)$$

where $0 < \alpha \leq 1$.

B. Modified Normalized Min-Sum Algorithm

Let X, Y be two sets of positive numbers with the same cardinality J and assume x^* and y^* are the smallest number in X and Y , respectively. Since $\phi(x)$ is a monotonic decreasing function for positive x , we have

$$\phi \left(\sum_{x_i \in X} \phi(x_i) \right) < x^*, \quad \phi \left(\sum_{y_i \in Y} \phi(y_i) \right) < y^* \quad (11)$$

Define

$$x_{sum} = \sum_i x_i, \quad y_{sum} = \sum_i y_i \quad (12)$$

It can be shown that if $x^* = y^*$ and $x_{sum} < y_{sum}$ then the following inequality holds with high probability.

$$\phi \left(\sum_i \phi(x_i) \right) < \phi \left(\sum_i \phi(y_i) \right) < x^* = y^* \quad (13)$$

Define

$$z_{min} = \min_{n' \in \mathcal{N}(m) \setminus n} |z_{n' \rightarrow m}| \quad (14)$$

$$z_{sum} = \sum_{n' \in \mathcal{N}(m) \setminus n} |z_{n' \rightarrow m}| \quad (15)$$

(13) and the fact that only the minimum value z_{min} is considered in (8) and (10) implies that in many cases when MSA and NMSA cannot distinguish between two sets of variable-to-check messages, z_{sum} can provide the information about their relative magnitudes of updating messages computed at a check node. (13) then suggests that we modify the factor α in (10) as

$$\alpha = 1 - \beta \cdot \frac{z_{min}}{z_{sum}} \quad (16)$$

where $\beta > 0$ is an empirical scalar that ensures $0 < \alpha \leq 1$. The resulting updating formula called the modified NMSA (M-NMSA), will have a variable correction factor that depends on z_{sum} and z_{min} .

V. SIMULATION RESULTS

A. Notations and assumptions

The frame error rate (FER) and the bit error rate (BER) performance of various combinations of the grouping/scheduling method and the message-updating formula in decoding Mackay's (504,252) regular LDPC code [16] with $d_c = 6$, $d_v = 3$ are examined. The frame duration is assumed to be equal to the codeword length, hence FER is actually the same as the codeword error probability. To distinguish these combinations we use the following shorthands. GHS-MSA,

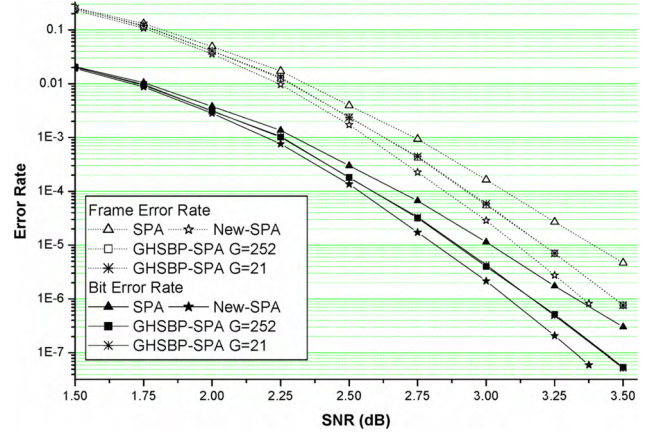


Fig. 1. FER performance of Mackay's (504,252) regular LDPC code using GHSBP, and New-SPA and fully-parallel BP decodings.

New-SPA, New-MSA, New-NMSA, and New-M-NMSA stand for the group horizontal shuffled min-sum algorithm and the GHS algorithm using the proposed grouping/scheduling method with SPA, MSA, NMSA and M-NMSA, respectively. If a shorthand contains no prefix, e.g., SPA, MSA, NMSA, M-NMSA, then the simulated decoding is assumed to be carried out with regular fully parallel processing.

The simulation results reported in this section assume $I_{Max} = 50$ for the proposed algorithms. In order to have a fair performance comparison with the conventional BP algorithm and the GHSBP algorithm under the same computation complexity, we set $I_{Max} = 650$ for the standard BP algorithm, the GHSBP algorithm, the min-sum algorithm and the GHS-MSA because of the complexity multiplicity $(d_v - 1) \cdot d_c + 1 = 13$ shown in Table III.

B. Numerical examples

Fig. 1 depicts the FER and BER performance of the standard BP algorithm, the GHSBP algorithm and the algorithm using the proposed grouping and scheduling. With the same computation complexity, our algorithm is about 0.23 dB better than the standard BP algorithm at the FER $\approx 10^{-4}$ and outperforms the GHSBP algorithm with $N_c = 12$ ($G = 21$) by about 0.1 dB at the FER $= 10^{-5}$. Our algorithm provides better performance but requires less decoding delay because $N_c = 13$. Performance curves in Fig. 2 show that New-MSA is about 0.25 dB better than MSA at the FER $= 10^{-4}$ and outperforms GHS-MSA 0.12 dB at the FER $= 10^{-5}$.

Fig. 3 compares the FER performance of the MSA, the NMSA, the M-NMSA and the standard BP algorithm. [13] suggested $\alpha = 0.8$ for the code with $d_v = 3$ and $d_c = 6$ as the best scaling factor and we set $\beta = 1.1$ for M-NMSA. The simulation results shows that M-NMSA gives FER performance better than that of NMSA and MSA under the same computation complexity constraint.

Fig. 4 compares the FER performance of the proposed algorithm using min-sum and its variations, i.e., performances of New-MSA, New-NMSA, New-M-NMSA and New-SPA are

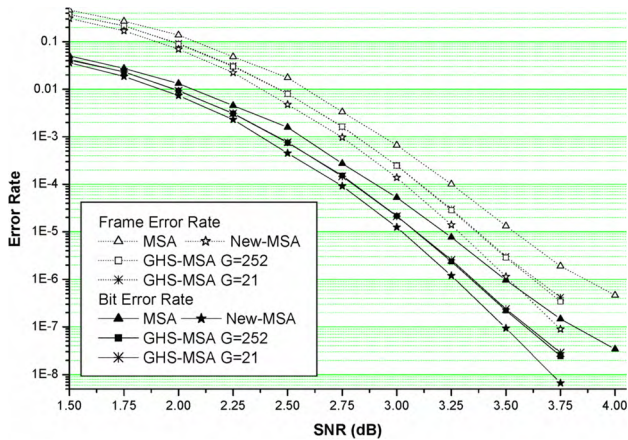


Fig. 2. FER performance of Mackay (504,252) regular LDPC code using the decoding algorithms: GHS-MSA, New-MSA and fully parallel MSA.

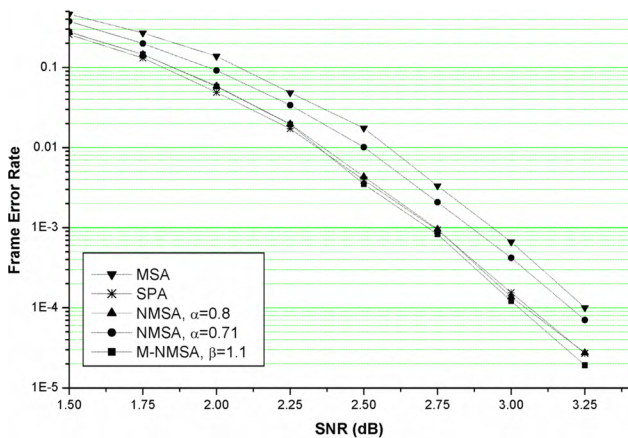


Fig. 3. FER performance of Mackay's (504,252) regular LDPC code using the decoding algorithms: MSA, NMSA, M-NMSA, and BP.

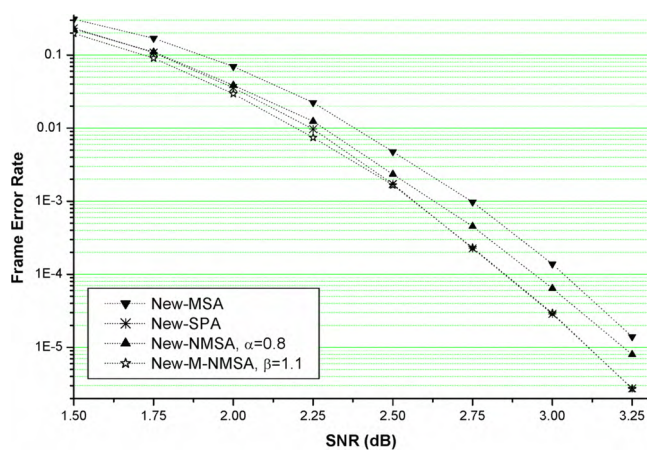


Fig. 4. FER performance of Mackay's (504,252) regular LDPC code using New-MSA, New-NMSA, and New-M-NMSA and New-SPA.

given. $\beta = 1.1$ is used in New-M-NMSA. The simulation curves indicate that NMSA is not suitable for use in conjunction with the proposed grouping and scheduling and New-M-NMSA outperforms New-NMSA by about 0.12 dB at the FER $\approx 5 \times 10^{-5}$.

VI. CONCLUSION

We have presented several new techniques to improve the performance of SPA and GHSBP algorithm for LDPC codes. The new grouping method is based on the ACN that measures the average degree of variable nodes within a group while the corresponding decoding schedule tries to use the newly updated messages as soon as they become available. Although the proposed grouping method results in multiple nodes-to-nodes message subiteration updates within each iteration, numerical results show that the increased complexity is more than compensated for by improved performance when used in conjunction with the proposed decoding schedule. That is, better performance is obtained with the same complexity.

To reduce the decoding complexity with minimum or no performance degradation, we also propose a modification of the normalized min-sum algorithm with a self-adjustable correction factor that take into account the information provided by the sum of variable-to-check messages. Simulated performance indicate that the new updating formula gives the FER performance almost as good as that of the classic SPA.

REFERENCES

- [1] R. G. Gallager, *Low-density parity-check codes*, Cambridge, MA: M.I.T Press, 1963.
- [2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Lett.*, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.
- [3] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399-431, Mar. 1999.
- [4] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan Kaufmann, 1998.
- [5] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, pp. 533-547, Sept. 1981.
- [6] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," in *Proc. 36th Annu. Asilomar Conf. Signals, Syst., Computers*, pp. 8-15, Nov. 2002.
- [7] J. Zhang and M.P.C. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.*, vol.53, no.2, pp.209-213, Feb. 2005.
- [8] J. Zhang, Y. Wang, and M. P. C. Fossorier, J. S. Yedidia, "Replica shuffled iterative decoding," *Proc. 2005 IEEE Int. Symp. on Information Theory*, pp. 454V-458, Adelaide, Australia, Sept. 2005.
- [9] J. Zhang, Y. Wang, and M. P. C. Fossorier, J. S. Yedidia, "Iterative decoding with replica," *IEEE Trans. Inform. Theory*, vol.53, no.5, pp.1644-1663, May 2007.
- [10] E. Yeo, P. Pakzad, B. Nikolic, V. Anantharam, "High throughput low-density parity-check decoder architectures," in *Proc. GLOBECOM '01*, pp. 3019-3024, Nov. 2001.
- [11] A. Ségard, F. Verdier, D. Declercq and P. Urard, "A DVB-S2 compliant LDPC decoder integrating the horizontal shuffle schedule," in *ISAPCS '06*, pp. 1013-1016, Dec. 2006.
- [12] M. P. C. Fossorier, M. J. Mihaljevic and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47 pp. 673-680, May, 1999.
- [13] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-Based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6 pp. 208-210, May, 2002.
- [14] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity-check codes," *IEEE Trans. Commun.*, vol. 50, pp. 406-414, Mar. 2002.
- [15] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, pp. 1288-1299, Aug. 2005.
- [16] D. J. C. MacKay, *Encyclopedia of sparse graph codes* [Online]. Available: <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>