Check Reliability Based Bit-Flipping Decoding Algorithms for LDPC Codes

Chi-Yuan Chang, Yu T. Su, Yu-Liang Chen, and Yin-Chen Liu
Institute of Communications Engineering
National Chiao Tung University
Hsinchu, 30056, TAIWAN,
Email: ytsu@mail.nctu.edu.tw

Abstract—We introduce new reliability definitions for bit and check nodes. Maximizing global reliability, which is the sum reliability of all bit nodes, is shown to be equivalent to minimizing a decoding metric which is closely related to the minimum likelihood decoding metric. We then propose novel bit-flipping (BF) decoding algorithms that take into account the check node reliability. Both hard-decision (HD) and soft-decision (SD) versions are considered. The former performs better than the conventional BF algorithm and, in most cases, suffers less than 1 dB performance loss when compared with some well known SD BF decoders. For one particular code it even outperforms those SD BF decoders. The performance of the SD version is superior to that of SD BF decoders and is comparable to or even better than that of the sum-product algorithm (SPA). The latter is achieved with a complexity much less than that required by the SPA.

I. INTRODUCTION

Low-density parity-check (LDPC) codes were first introduced by Gallager [1] in early 1960s and rediscovered by Mackay [2][3] in 1990s. Soft-decision, hard-decision and hybrid decoding algorithms have been proposed for decoding LDPC codes. The sum-product algorithm (SPA) achieves the near-capacity performance asymptotically but its computational complexity is very high. The min-sum algorithm (MSA) [5], which replaces the nonlinear check node operation by a single minimum operation, was proposed to reduce the complexity of the standard SPA at the cost of a noticeable degradation in the decoding performance. Chen and Fossorier [11]-[13] suggested the normalized min-sum algorithm and the offset min-sum algorithm which multiply and add a constant correction factor in the check-to-variable updating equation of MSA. They offer performance compatible to that of the SPA but with lower complexity.

Bit-flipping (BF) decoding algorithm is a hard-decision decoding algorithm which is much simpler than SPA or its modifications but does not perform as well. To reduce the performance gap between SPA and BF based decoders, variants of the latter such as weighted bit-flipping (WBF) [7], modified weighted bit-flipping (MWBF) [8] and improved modified bit-flipping (IMWBF) [9] algorithms have been proposed. They provide tradeoffs between computational complexity and error performance. The reliability ratio based weighted bit-flipping (RRWBF) [10] decoding algorithm needs not to find optimal parameters as variants of the WBF algorithm do but yields better performance.

In this paper, we present novel bit-flipping algorithms called check reliability based bit-flipping (CRBF) decoding algorithms for decoding LDPC codes. Starting with the maximum likelihood (ML) decoding metric, we first relax the parity-check requirements and introduce a global cost which is the sum $N$ local costs, where $N$ is the codeword length. Interpreting each cost as the opposite of reliability, the bit(s) with maximal local cost or least reliability is (are) flipped in each decoding iteration. By reducing the local costs iteratively, we hope that the global cost can also be minimized to approach the ML cost. We define the reliability of a check node and use this reliability to modify and update the bit (node) reliabilities.

Two CRBF algorithms are proposed: the soft check reliability based bit-flipping (soft-CRBF) algorithm, which processes the received channel values when decoding, and its hard-decision counterpart which sends the hard-decision demodulated bit streams to the decoder. The soft-CRBF outperforms the WBF decoding algorithm and its variants and is comparable to SPA for some LDPC codes. We also compare the performance of the hard-CRBF and standard BF decoders and the simulation results prove that the former is a better choice.

The rest of this paper is organized as follows. In Section II-A we introduce the global and local costs as minimizing decoding costs and derive the relation between the global cost and the ML cost. The check node cost and reliability are defined in Section II-B. In Section II-C we describe the proposed CRBF algorithms, and in II-D the cost functions of conventional WBF algorithms are examined. Section III discusses the computational complexities of the proposed algorithms, and Section IV provides some simulated numerical results concerning the performance of our and some well known algorithms. Finally, conclusion remarks are drawn in Section V.

II. CHECK RELIABILITY BASED BIT-FLIPPING DECODING ALGORITHMS

A. COST FUNCTIONS

A regular binary $(N, K)$ $(d_v, d_c)$ LDPC code $C$ is a linear block code described by an $M \times N$ parity check matrix $H$ which has constant column weight of $d_v$ and row weight of $d_c$. $H$ can be represented by 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 $H$. The code rate of $C$ is given by $R = K/N$.

Assume BPSK signaling with unit energy is used. A codeword $c = (c_0, c_1, \ldots, c_{N-1})$ is mapped into a bipolar sequence $\hat{c} = (\hat{c}_0, \hat{c}_1, \ldots, \hat{c}_{N-1})$ and transmitted over an AWGN channel with noise variance $\sigma^2$. Let $y = (y_0, y_1, \ldots, y_{N-1})$ be the corresponding received soft-decision sequence and the binary hard-decision sequence, $z = (z_0, z_1, \ldots, z_{N-1})$, is obtained as follows:

$$z_i = \begin{cases} 0 & \text{if } y_i \geq 0 \\ 1 & \text{if } y_i < 0 \end{cases} . \tag{1}$$

Let $x^l = (x_0^l, x_1^l, \ldots, x_{N-1}^l)$ be the decoded sequence at the $l$th iteration, and $s^l = (s_0^l, s_1^l, \ldots, s_{M-1}^l)$ be the syndrome vector of $x^l$:

$$s^l = (s_0^l, s_1^l, \ldots, s_{M-1}^l) = x^lH^T. \tag{2}$$

Define $\hat{z} = (\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_{N-1})$, $\hat{x}^l = (\hat{x}_0^l, \hat{x}_1^l, \ldots, \hat{x}_{N-1}^l)$ and $\hat{s}^l = (\hat{s}_0^l, \hat{s}_1^l, \ldots, \hat{s}_{M-1}^l)$ be the bipolar modulated sequences corresponding to $z$, $x^l$ and $s^l$, respectively. We thus have $\hat{z}_i = 1 - 2 \cdot z_i$, $\hat{x}_i^l = 1 - 2 \cdot x_i^l$ for $0 \leq i < N$, $\hat{s}_i^l = 1 - 2 \cdot s_i^l$ for $0 \leq i < M$, and $\hat{x}^l, \hat{z}^l \in \{1, -1\}^N \equiv F_2^N$.

Let $N(m)$ 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. $N(m) \setminus n$ is defined as the set $N(m)$ with the variable node $n$ excluded while $\mathcal{M}(n) \setminus m$ is the set $\mathcal{M}(n)$ with the check node $m$ excluded.

Maximum likelihood (ML) decoding would find the unique bipolar vector $\hat{x} = \arg \max_{x \in C} \sum_{i=0}^{N-1} \hat{x}_i y_i$, where the constraint $\hat{x} \in C$ is equivalent to $\hat{s}_j = \prod_{j' \in \mathcal{M}(j)} \hat{x}_j$, $\forall 0 \leq j \leq M - 1$. Equivalently, the ML decoder solves the unstrained optimization problem

$$\arg \min_{\hat{x} \in F_2^N} -\sum_{i=0}^{N-1} \hat{x}_i y_i + \sum_{j=0}^{M-1} \alpha_j (\hat{s}_j - 1) . \tag{3}$$

where $\alpha_j$ are Lagrange multipliers. Although $\alpha_j$’s can be arbitrary real numbers the fact that each $\hat{s}_j$ can only take two values, +1 and -1, implies that they must be positive for otherwise the cost function will encourage the violation of the constraints $\hat{s}_j = 1$. Instead of solving the above ML problem we attack a simpler problem by relaxing the constraints and define a global cost (GC) associated with a candidate bipolar $N$-tuple $\hat{x}$ as

$$E(\hat{x}) \triangleq -\sum_{i=0}^{N-1} \hat{x}_i y_i - \alpha \sum_{j=0}^{M-1} \hat{s}_j , \tag{4}$$

where $\alpha > 0$. The above cost replaces the multiple constraints $\{\hat{s}_i - 1 = 0, i = 0, 1, \ldots, M - 1\}$ by the single constraint $\sum_{i=0}^{M-1} \hat{s}_i = M$. The second term in the above equation is used to penalize each invalidate check relation and the penalty is minimized if $\hat{x}$ is a valid codeword. Unlike (4) which penalizes each violation $\hat{s}_j \neq 1$ differently, (3) imposes a constant penalty.

(4) can be rewritten as

$$E(\hat{x}) = -\sum_{i=0}^{N-1} \hat{x}_i y_i + \frac{\alpha}{d^c} \sum_{j \in \mathcal{M}(i)} \hat{s}_j$$

$$= -\sum_{i=0}^{N-1} \hat{x}_i y_i + \gamma \sum_{j \in \mathcal{M}(i)} \hat{s}_j$$

$$= \sum_{i=0}^{N-1} E_i$$, \tag{5}$$

where $\gamma = \frac{\alpha}{d^c}$ is also a positive constant. We accordingly define the local cost (LC) for variable node $i$ by

$$E_i \triangleq -\left( \hat{x}_i y_i + \gamma \sum_{j \in \mathcal{M}(i)} \hat{s}_j \right)$$

and interpret the quantity $-E_i$, if positive, as the reliability of variable node $i$. Minimizing the global cost is thus equal to maximizing the total (variable node) reliability. Note that the LC’s are not independent and related through $\hat{s}_i$ unless $\mathcal{M}(i) \cap \mathcal{M}(j) = \emptyset, \forall j \neq i$.

B. Reliability of a Check Node

Given the reliability and local cost of a variable node, we now define the corresponding reliability and local cost of a check node by

$$R_{mn} = \max(-R_{mn}'$, 0) \tag{7}$$

where

$$R_{mn}' = \max_{n' \in N(m) \setminus n} E_{n'}$$ \tag{8}$$

$R_{mn}$ is the maximum local cost amongst those of the variable nodes connecting to the check node $m$ except node $n$. In other words, $R_{mn}$ is used as a measure of the unreliability of the massage check node $m$ intends to pass to variable node $n$. This unreliability is equal to the maximal cost of the variable nodes in $N(m) \setminus n$. If the maximum local cost (unreliability) is too high, (7) will return a constant zero, which means the check node $m$ is totally useless for variable node $n$. In contrast, the reliability $R_{mn}$ is a positive weighting factor which indicates whether the check node (relation) $m$ can provide proper reliability information for variable node $n$. Note that a bit-flipping on variable node $i$ will result in a magnitude change of $E_i$. The maximum magnitude change is $|y_i| + |\mathcal{M}(i)|$, where $|\mathcal{M}(i)|$ denotes the cardinality of $\mathcal{M}(i)$. The definition of the check reliability in (7) and (8) indicates this maximum magnitude change is also the maximum candidate value for $R_{mn}$ which occurs when a incorrect bit decision is made and all check relations are violated.
The resulting algorithm is called the decoding algorithm is summarized in Table I.

The first, calculate the reliability of each check node, and update decision bit(s) which is (are) most unreliable (largest LC) of iterations.

Let $R^0_{ji} = 1, \forall i \in \mathcal{N}(j), 0 \leq j < M.\ E^0_{ij} = -y_i - \gamma \sum_{j \in \mathcal{M}(i)} \hat{s}^0_{ij},$ for $0 \leq i < N.$

Step 1:

$l = l + 1.\ \hat{x}^l = \hat{x}^{l-1}, 0 \leq i < N.\ e = \arg\max_j E^l_{ji}$, then let $\hat{x}_e = -\hat{x}^{l-1}_e.$

Step 2:

Compute $s^l.$ If $s^l = 0$ or $l = I_{max},$ stop decoding and output $\hat{x}^l$ as the decoded sequence. Otherwise, go to step 3.

Step 3:

$\forall i \in \mathcal{N}(j)$ and $0 \leq j < M,$ compute $R^l_{ji}.$

Step 4:

For $0 \leq i < N,$ compute $E^l_{ij}$ and go to step 1.

### C. Check Reliability Based Bit-Flipping Decoding Algorithms

When using an iterative algorithm to find the minimum GC, $\hat{x}_j$ and $\hat{s}_i$ are replaced by their $l$th iteration values $\hat{x}^l_j, \hat{s}^l_i$ in computing (4). To use the check reliability information we update the LC at the $l$th iteration by

$$E^l_{ij} \overset{def}{=} -\left(\hat{x}^l_j y_i + \gamma \sum_{j \in \mathcal{M}(i)} R^l_{ji} \hat{s}^l_j\right),$$

where

$$R^l_{ji} = \max(-R^l_{ji}, 0)$$

is the reliability of check node $j$ for variable node $i$ in the $l$th iteration and

$$R^l_{ji} = \max_{\ell \in \mathcal{N}(j) \setminus i} E^{\ell-1}_{\ell,i} - \gamma \hat{s}^{\ell-1}_j R^{\ell-1}_{ji},$$

is the modified unreliability of the check node $j$ for variable node $i.$ Since $\gamma > 0,$ (9) and (11) indicate that we have put larger weights on more reliable checks.

The basic procedure of the proposed CRBF decoding algorithms works as follows. At each iteration we flip the decision bit(s) which is (are) most unreliable (largest LC) first, calculate the reliability of each check node, and update the LC of each variable node if needed. The procedure stops if a valid codeword is found or if the maximum number of iterations $I_{max}$ is reached. The soft CRBF (soft-CRBF) decoding algorithm is summarized in Table I.

If we replace $y_i$ with $\hat{z}_i$ in the Step 4 of Table I, the resulting algorithm is called the hard CRBF (hard-CRBF) decoding algorithm.

### D. Cost Functions of Known WBF Decoding Algorithms

For the weighted bit-flipping (WBF) decoding, the cost function or the objective function for $\hat{x}^l_j$ in the $l$th iteration is defined by

$$E^l_{i,WBF} \triangleq -\sum_{j \in \mathcal{M}(i)} \hat{s}^l_j \cdot w_{ji},$$

where

$$w_{ji} = \min_{i' \in \mathcal{N}(j) \setminus i} |y_{i'}|.$$  

For modified WBF (MWBF) and improved modified WBF (IMWBF) decoding algorithms, the cost functions for $\hat{x}^l_j$ in the $l$th iteration is defined by

$$E^l_{i,MWBF} \triangleq -\sum_{j \in \mathcal{M}(i)} \hat{s}^l_j \cdot w_{ji} - |y_i|$$

and

$$E^l_{i,IMWBF} \triangleq -\sum_{j \in \mathcal{M}(i)} \hat{s}^l_j \cdot w_{ji} - \alpha |y_i|$$

respectively, where $\alpha$ is a positive constant and can be optimized by simulations. The bit with maximal cost will be flipped in the WBF, MWBF and IMWBF decoding algorithms.

### III. Computational Complexity

For the proposed algorithm, each decoding iteration consists of check-reliability and the cost updates. The number of check-reliability updates in the proposed algorithms is highly dependent on the node degrees of the code used. Suppose only one bit is flipped in each iteration, we will need $\min\{M+1, d_v \cdot (d_c - 1) + 1\}$ standard operations, each includes the check-reliability updates and the selection of the bit with the maximum local cost, in one iteration. $\min\{N, d_v \cdot d_c\}$ cost calculations are also needed.

For the WBF, MWBF and IMWBF decoding algorithms, $w_{ji}$ is used to adjust the reliability of check node $j$ for variable node $i.$ $w_{ji}$ will not be modified in the course of iterative decoding, so the computational complexity is lower than that of the proposed algorithms. The standard BF algorithm can be regarded as a special case of the WBF decoder with $w_{ji} = 1,$ which has the lowest complexity but gives relatively poor performance.

### IV. Numerical Results

Computer-simulated bit error rate (BER) performance of various BF decoding methods, SPA and proposed decoding algorithms are reported in this section. Mackay’s (504,252)(3,6) 0.5-rate LDPC code, (255,175)(16,16) 0.69-rate EG-LDPC code, the (1440,1344)(3,45) 0.93-rate LDPC code defined in 802.15.3c and the (2048,1723)(6,32) 0.84-rate LDPC code defined in 802.3a/n are used for comparison in our simulations.

Fig[11] depicts the BER performance of the standard BF, WBF, MWBF, soft-CRBF and hard-CRBF decoding algorithms with $I_{max} = 30.$ The performance of IMWBF decoding is not shown in this figure because the optimal parameter value for $\alpha$ equals to 1, that is, the two algorithms, IMWBF and MWBF, become identical for this code. The proposed soft-CRBF algorithm outperforms the SPA, WBF...
and MWBF by about 0.35 dB, 0.5 dB, and 0.8 dB, respectively at BER $= 10^{-5}$. The hard-CRBF decoding algorithm outperforms the standard decoding algorithm, which is also a pure hard-decision decoding algorithm, by about 0.5 dB at BER $= 4 \times 10^{-4}$.

Performance curves in Fig. 2 show that at BER $\approx 10^{-4}$, the proposed soft-CRBF algorithm gives 3 dB perform gain with respect to the variants of WBF algorithms. The hard-CRBF algorithm offers 2 dB perform gain against the standard BF decoder at BER $= 2 \times 10^{-4}$ and provides better performance than the variants of WBF algorithms; $I_{max}$ being set as 70 for this code.

Fig. 3 compares the BER performance of the SPA, variants of the WBF algorithm, and the proposed algorithms with $I_{max} = 30$. The soft-CRBF decoding algorithm outperforms the IMWBF and the MWBF decoders by about 0.9 dB at BER $= 10^{-5}$ and yields 1.3 dB gain against the WBF decoder at the same BER. The hard-CRBF decoding algorithm is about 0.6 dB better than the standard BF at BER $= 1.2 \times 10^{-5}$.

Fig. 4 indicates that the proposed soft-CRBF decoder yields near-SPA performance and outperform the WBF, MWBF and IMWBF decoder by a margin larger than 1 dB at BER $= 10^{-5}$. The hard-CRBF not only outperforms the standard BF but is also superior to the MWBF and IMWBF algorithms at BER $= 10^{-6}$ when $I_{max} = 70$.

V. Conclusion

We have presented two novel check reliability based soft-decision bit-flipping decoding algorithms to improve the per-
formance of the WBF algorithm and its variants for decoding LDPC codes. At each iteration, the cost/reliability for each bit is computed and the bit with least reliability is flipped. The check reliability is also defined for each check node and is used to update the related bit node reliabilities. The sum of bit cost/reliability is shown to be a relaxed version of the ML decoding metric. Our algorithms are iterative approaches for minimizing the sum reliability. Numerical results show that the proposed soft-decision decoding algorithm outperforms the conventional WBF algorithm and its variants. On the other hand, the hard-decision version outperforms the standard bit-flipping decoder and, for some codes, even offers performance better than that of the WBF decoding algorithm.

ACKNOWLEDGEMENT

This work was supported in part by Taiwan’s National Science Council under Grant NSC-96-2221-E-009-076-MY3.

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, 1988.
[5] 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.
[6] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. IT-27, pp. 533-547, Sept. 1981.
[7] Y. Kou, S. Lin and M. P. C. Fossorier, “Low-density parity-check codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 47, pp.2711-236, Nov. 2001.
[8] J. Zhang and M. Fossorier, “A modified weighted bit-flipping decoding of low-density parity-check codes,” *IEEE Commun. Lett.*, vol. 8, pp. 165-167, Mar. 2004.
[9] M. Jiang, C. Zhao, Z. Shi, and Y. Chen, “An improvement on the modified weighted bit flipping decoding algorithm for LDPC codes,” *IEEE Commun. Lett.*, vol. 9, pp. 814-816, Sept. 2005.
[10] Feng Guo and Laios Hanzo “Reliability ratio based weighted bit-flipping decoding for LDPC codes,” *VTC Spring 2005. IEEE*, vol. 1, pp. 709-713, May 2005.
[11] 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.
[12] 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.
[13] 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.
[14] Guiqiang Dong, Yanan Li, Ningde Xie, Tong Zhang, and Huaping Liu, “Candidate Bit Based Bit-Flipping Decoding Algorithm for LDPC Codes,” Proc. 2009 IEEE Int. Symp. on Information Theory, pp. 2166-2168, Seoul, Korea, Jun. 2009.
[15] Xiaofu Wu, Cong Ling, Ming Jiang, Enyang Xu, Chunming Zhao, and Xiaohu You, “New Insights in Weighted Bit-Flipping Decoding,” *IEEE Trans. Commun.*, vol. 57, No. 8, pp. 2177-2180, Aug. 2009.
[16] D. J. C. MacKay, *Encyclopedia of sparse graph codes* [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html