Bit-Optimal Decoding of Codes whose Tanner Graphs are Trees ⋆

E. Soljanin

Bell Laboratories, Lucent Technologies, Murray Hill NJ 07974, USA

E. Offer

Munich University of Technology, Institute of Comm. Eng., 80290 Munich, Germany

Abstract

We introduce a group algebra formulation for bit optimal decoding of binary block codes. We use this new framework to give a simple algebraic proof that Pearl's and Gallager's belief propagation decoding algorithms are bit-optimal when the Tanner graph of the code is a tree. We believe that these derivations of known results give new insights into the issues of decoding on graphs from the algebraic coding theorist's point of view.

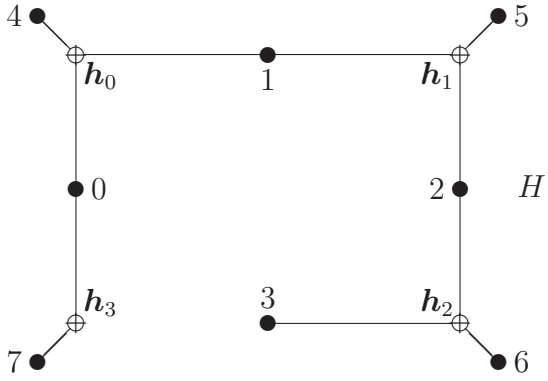
Key words: Bit-optimal decoding, iterative decoding, soft-output decoding, suboptimal decoding
1991 MSC: 94B05, 94B25, 94B35, 94B60

1 Introduction

We consider a linear, binary, block code $\mathcal{C}$ defined by its parity-check matrix $H = \{h_{ij}\}_{(n-k)\times n}$. It is common and useful for describing and analyzing decoding algorithms to represent H by a *Tanner graph* [1]. In this *bipartite* graph, the two groups of nodes are referred to as the *check* and *variable* nodes. Each check node corresponds to a row of the parity-check matrix, and each variable

⋆ This work was supported by the 1999 German-American Networking Research Grant given by the national academics of engineering of Germany and the USA
Email addresses: emina@lucent.com (E. Soljanin), elke.offer@ei.tum.de (E. Offer).
URLs: cm.bell-labs.com/~emina (E. Soljanin), www.lnt.ei.tum.de (E. Offer).

node corresponds to a position (coordinate) in codewords of $\mathcal{C}$. When row $\boldsymbol{h}_i$ has a 1 at position j, then there is a branch in the Tanner graph between the corresponding check and variable nodes. We are concerned only with codes whose Tanner graphs are trees, *i.e.,* have no loops. Figure 1 shows the Tanner graph and the parity check matrix H of an $(8, 4)$ "tree" code.



$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. An $(8, 4)$ tree code.

Tanner graphs (also known as *factor* graphs) have been used to describe and analyze various decoding algorithms in terms of message-passing between the graph nodes. These algorithms include Gallager's algorithms for low density parity check (LDPC) codes, Pearl's belief propagation, the Viterbi algorithm, certain fast Fourier transforms, forward/backward BCJR algorithm, and the turbo decoding algorithm [2]–[6]. In this paper we introduce a group algebra framework and give simple and elegant algebraic proof that the belief propagation decoding is bit-optimal when the Tanner graph of the code is a tree. We believe that this derivation of a result of Pearl [2] gives new insights into the issues of decoding on graphs from the algebraic coding theorist's point of view.

The paper is organized as follows: In Sec. 2, we put the bit-optimal decoding in the group algebra framework. In Sec. 3, we show that the belief propagation is bit-optimal when the Tanner graph of the code is a tree. In Sec. 4, we show how Gallager's belief propagation approximates the optimal algorithm.

2 A Group Algebra Formulation of Bit-Optimal Decoding

2.1 A Group Algebra Framework

Clearly, $\mathcal{C}$ is a subgroup of the additive group of $\mathbb{F}_2^n$, and the rows of the parity-check matrix $\boldsymbol{h}_i = \{h_{ij}\}_{1\times n}$, $0 \le i < n-k$, are the group generators of the dual code $\mathcal{C}'$. For the additive groups $\mathcal{C}$, $\mathcal{C}'$, and $\mathbb{F}_2^n$, we define isomorphic multiplicative groups $\mathcal{G}$, $\mathcal{G}'$, and $\mathcal{F}$ as follows: For each $\boldsymbol{c} \in \mathcal{C}$, $\boldsymbol{c} = c_0 c_1 \ldots c_{n-1}$, there is a corresponding $\boldsymbol{\lambda}^{\boldsymbol{c}} \in \mathcal{G}$, $\boldsymbol{\lambda}^{\boldsymbol{c}} = \lambda_0^{c_0}\lambda_1^{c_1}\ldots\lambda_{n-1}^{c_{n-1}}$. The group operation in $\mathcal{G}$ is multiplication $\otimes$: $\boldsymbol{\lambda}^{\boldsymbol{c}_1} \otimes \boldsymbol{\lambda}^{\boldsymbol{c}_2} = \boldsymbol{\lambda}^{\boldsymbol{c}_1 \oplus \boldsymbol{c}_2}$, where $\oplus$ is the addition in $\mathbb{F}_2^n$. It is easy to see that $\mathcal{C} \cong \mathcal{G}$. The groups $\mathcal{G}'$ and $\mathcal{F}$ are defined in the same manner.

Remark 1 *Let $\boldsymbol{e_i}$ denote the unit vector whose all components are 0 except the i-th which is 1. Note that $\lambda_i = \boldsymbol{\lambda}^{\boldsymbol{e}_i} \in \mathcal{F}$, and thus*

$$\lambda_i \otimes \lambda_j = \boldsymbol{\lambda}^{\boldsymbol{e}_i \oplus \boldsymbol{e}_j} = \begin{cases} \lambda_i \cdot \lambda_j, & i \neq j, \\ 1, & i = j. \end{cases}$$

The *group algebra* of a multiplicative group $\mathcal{G}$ over the field of reals $\boldsymbol{R}$, denoted by $\boldsymbol{R}\mathcal{G}$, consists of elements of the form

$$\sum_{g\in\mathcal{G}} \alpha_g g, \ \alpha_g \in \boldsymbol{R}.$$

Note that $\boldsymbol{R}\mathcal{G}$ is a vector space whose basis are the elements of $\mathcal{G}$. The rules for addition and scalar multiplication are the natural ones. The algebra structure on this vector space (the multiplication in $\boldsymbol{R}\mathcal{G}$) is defined based on the product operation in $\mathcal{G}$ as

$$\left(\sum_{f\in\mathcal{G}} \alpha_f f\right) \otimes \left(\sum_{g\in\mathcal{G}} \alpha_g g\right) = \sum_{f,g\in\mathcal{G}} \alpha_f \alpha_g (f \otimes g).$$

Here, the concept of group algebra is introduced similarly as in [7, pp. 132–134], where it was used for deriving the MacWilliams identities for non-linear codes.

Definition 2 *The following polynomial over $\lambda_1, \ldots, \lambda_n$ shall be called the generalized dual weight enumerator of code $\mathcal{C}$:*

$$A_{\mathcal{C}} = \sum_{\boldsymbol{c}' \in \mathcal{C}'} \prod_{j=0}^{n-1} \lambda_j^{c'_j}.$$

From the definition of $\boldsymbol{R}\mathcal{G}$, we see that

$$A_{\mathcal{C}} = \sum_{\boldsymbol{c}' \in \mathcal{C}'} \boldsymbol{\lambda}^{\boldsymbol{c}'} = \prod_{i=0}^{n-k-1}{}_{\otimes} \left[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\right]$$

and

$$\underbrace{A_{\mathcal{C}} \otimes A_{\mathcal{C}} \otimes \cdots \otimes A_{\mathcal{C}}}_{\nu} = 2^{(\nu-1)(n-k)} \cdot A_{\mathcal{C}}.$$

There are similarities between multiplying polynomials over $\lambda_1, \ldots, \lambda_n$ and multiplying elements of $\boldsymbol{R}\mathcal{G}$. Let us take a look at the following example:

Example 1 *Comparing multiplication in $\boldsymbol{R}\mathcal{G}$ and $\boldsymbol{R}$:*

- *product of $1 + \lambda_0\lambda_1$ and $1 + \lambda_2\lambda_3$ is the same in $\boldsymbol{R}\mathcal{G}$ and $\boldsymbol{R}$:*

$$\begin{aligned}(1 + \lambda_0\lambda_1) \otimes (1 + \lambda_2\lambda_3) &= 1 + \lambda_0\lambda_1 + \lambda_2\lambda_3 + \lambda_0\lambda_1\lambda_2\lambda_3 \\ (1 + \lambda_0\lambda_1) \cdot (1 + \lambda_2\lambda_3) &= 1 + \lambda_0\lambda_1 + \lambda_2\lambda_3 + \lambda_0\lambda_1\lambda_2\lambda_3\end{aligned}$$

- *product of $1 + \lambda_0\lambda_1$ and $1 + \lambda_1\lambda_2$ is different in $\boldsymbol{R}\mathcal{G}$ and $\boldsymbol{R}$:*

$$\begin{aligned}(1 + \lambda_0\lambda_1) \otimes (1 + \lambda_1\lambda_2) &= 1 + \lambda_0\lambda_1 + \lambda_1\lambda_2 + \lambda_0\lambda_2 \\ (1 + \lambda_0\lambda_1) \cdot (1 + \lambda_1\lambda_2) &= 1 + \lambda_0\lambda_1 + \lambda_1\lambda_2 + \boxed{\lambda_1^2}\lambda_0\lambda_2\end{aligned}$$

Now consider two elements of $\boldsymbol{R}\mathcal{G}$: $\boldsymbol{\lambda}^{\boldsymbol{c}_1}$ and $\boldsymbol{\lambda}^{\boldsymbol{c}_2}$ such that $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ have disjoint supports. We have

$$\boldsymbol{\lambda}^{\boldsymbol{c}_1} \otimes \boldsymbol{\lambda}^{\boldsymbol{c}_2} = \boldsymbol{\lambda}^{\boldsymbol{c}_1 \oplus \boldsymbol{c}_2} = \boldsymbol{\lambda}^{\boldsymbol{c}_1 + \boldsymbol{c}_2} = \boldsymbol{\lambda}^{\boldsymbol{c}_1} \cdot \boldsymbol{\lambda}^{\boldsymbol{c}_2}.$$

In general, the following holds:

Lemma 3 *Let $\sum_{\boldsymbol{c} \in \mathcal{S}_1} \alpha_c \boldsymbol{\lambda}^{\boldsymbol{c}}$ and $\sum_{\boldsymbol{c} \in \mathcal{S}_2} \alpha_c \boldsymbol{\lambda}^{\boldsymbol{c}}$ be two elements of $\boldsymbol{R}\mathcal{G}$ associated with subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{C}$ such that the supports of codewords in $\mathcal{S}_1$ and the supports of codewords in $\mathcal{S}_2$ are disjoint. Then,*

$$\Big(\sum_{\boldsymbol{c} \in \mathcal{S}_1} \alpha_c \boldsymbol{\lambda}^{\boldsymbol{c}}\Big) \otimes \Big(\sum_{\boldsymbol{c} \in \mathcal{S}_2} \alpha_c \boldsymbol{\lambda}^{\boldsymbol{c}}\Big) = \Big(\sum_{\boldsymbol{c} \in \mathcal{S}_1} \alpha_c \boldsymbol{\lambda}^{\boldsymbol{c}}\Big) \cdot \Big(\sum_{\boldsymbol{c} \in \mathcal{S}_2} \alpha_c \boldsymbol{\lambda}^{\boldsymbol{c}}\Big).$$

We show later that Gallager's belief propagation would be optimal for any parity check code (not necessarily low-density) if the multiplication were carried out in $\boldsymbol{R}\mathcal{G}'$ rather than in $\boldsymbol{R}$. When there are no loops in the Tanner graph of the code, the results of multiplication in $\boldsymbol{R}\mathcal{G}'$ and $\boldsymbol{R}$ are the same. Consequently, belief propagation for codes whose Tanner graphs are trees is bit optimal. This is all discussed in Sec. 3.

2.2 Bit Optimal Decoding

Since the message passing decoding algorithms on Tanner graphs directly deal only with the generators $\boldsymbol{h}_i$ of the dual code, we start with an expression for bit-optimal decoding based on computations over the dual code $\mathcal{C}'$. The expression involves all codewords in $\mathcal{C}'$. We derive its group algebra formulation which enables us to rewrite it so that it explicitly involves only $\boldsymbol{h}_i$.

The bit optimal decoding rule maximizes $P(c_m = b|\boldsymbol{r})$, the probability that c_m equals $b \in \{0, 1\}$ given the received word $\boldsymbol{r}$. The optimal decoder computes the log-likelihood of bit m over code $\mathcal{C}$, $L_m^{\mathcal{C}}$, as

$$L_m^{\mathcal{C}} = \log \frac{P(c_m = 0|\boldsymbol{r})}{P(c_m = 1|\boldsymbol{r})}. \tag{1}$$

When the codewords are equiprobable and the channel memoryless, a result of Hartmann and Rudolph [9] (derived and presented somewhat differently in [8] and [10]) gives an expression for computing $L_m^{\mathcal{C}}$ over the dual code:

$$L_m^{\mathcal{C}} = \log \frac{p(r_m|0)}{p(r_m|1)} + \log \frac{\displaystyle\sum_{i=0}^{2^{n-k}-1} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \left(\frac{p(r_j|0) - p(r_j|1)}{p(r_j|0) + p(r_j|1)}\right)^{c'_{ij}}}{\displaystyle\sum_{i=0}^{2^{n-k}-1} (-1)^{c'_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \left(\frac{p(r_j|0) - p(r_j|1)}{p(r_i|0) + p(r_j|1)}\right)^{c'_{ij}}}. \tag{2}$$

We find a group algebra interpretation of this expression.

Let

$$L_j = \log \frac{p(r_j|0)}{p(r_j|1)} \quad \text{and} \quad \lambda_j = \frac{p(r_j|0) - p(r_j|1)}{p(r_j|0) + p(r_j|1)} = \tanh(L_j/2). \tag{3}$$

Note that L_j is the log-likelihood of bit j given only the received value r_j; it could be used for the best guess for bit j without the knowledge of the code. By substituting (3) in (2), we obtain

$$L_m^{\mathcal{C}} = \log \frac{1 + \lambda_m}{1 - \lambda_m} + \log \frac{\displaystyle\sum_{i=0}^{2^{n-k}-1} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{c'_{ij}}}{\displaystyle\sum_{i=0}^{2^{n-k}-1} (-1)^{c'_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{c'_{ij}}}. \tag{4}$$

Now, the expressions in the numerators and denominators in equation (4) are elements of group algebra $\boldsymbol{R}\mathcal{F}$, since $1 = \lambda^{\mathbf{0}}$, $\lambda_m = \boldsymbol{\lambda}^{\boldsymbol{e}_m}$, and $\prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{c'_{ij}}$ are all in $\mathcal{F}$. It is easy to see that

$$\sum_{i=0}^{2^{n-k}-1} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{c'_{ij}} = \prod_{i=0}^{n-k-1}{}_{\otimes}(1 + \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}) \quad \text{and}$$

$$\sum_{i=0}^{2^{n-k}-1} (-1)^{c'_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{c'_{ij}} = \prod_{i=0}^{n-k-1}{}_{\otimes}(1 + (-1)^{h_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}).$$

Therefore equation (4) can be written as

$$L_m^{\mathcal{C}} = \log \frac{1+\lambda_m}{1-\lambda_m} + \log \frac{\prod_{i=0}^{n-k-1}{}_{\otimes}\Big[1 + \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}{\prod_{i=0}^{n-k-1}{}_{\otimes}\Big[1 + (-1)^{h_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}. \tag{5}$$

The corresponding $\lambda_m^{\mathcal{C}} = \tanh(L_m^{\mathcal{C}}/2)$ can be found as follows [10]:

Theorem 4 *Let λ_j defined by (3) be the j-th soft input to the bit optimal decoder for code $\mathcal{C}$. Then its soft output for bit m is given by*

$$\lambda_m^{\mathcal{C}} = \tanh(L_m^{\mathcal{C}}/2) = \frac{\boldsymbol{\lambda}^{\boldsymbol{e_m}} \otimes \prod_{i=0}^{n-k-1}{}_{\otimes}\big[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\big]}{\prod_{i=0}^{n-k-1}{}_{\otimes}\big[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\big]}. \tag{6}$$

Note that in equations (4) and (6), the expressions in the numerators and denominators are elements of group algebra $\boldsymbol{R}\mathcal{F}$, and have to be evaluated in $\boldsymbol{R}$ before the divisions and logarithms are performed.

3 Message Passing Bit Decoders

3.1 Optimal Message Passing

We first consider general parity check codes. Let $\mathcal{C}'_1$ and $\mathcal{C}'_2$ be two subcodes of the dual code $\mathcal{C}'$ obtained by partitioning its set of generators $\boldsymbol{h}_i$, $0 \leq i < n-k$. Let $\mathcal{I}$ be the index set $\mathcal{I} = \{0, 1, \ldots, n-k-1\}$, and $\mathcal{I}_1$ and $\mathcal{I}_2$ its two disjoint subsets such that $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$. Thus each $\mathcal{C}'_l \subset \mathcal{C}'$ is defined by its set of generators $\boldsymbol{h}_{i_l}$, $i_l \in \mathcal{I}_l$, $l = 1, 2$. We show that the belief propagation is optimal (albeit computationally complex) for general parity check codes when certain products in $\boldsymbol{R}$ are replaced by products in $\boldsymbol{R}\mathcal{G}'$. Equation (6) for the soft

output of the optimal decoder can now be written as

$$\lambda_m^{\mathcal{C}} = \frac{\boldsymbol{\lambda}^{\boldsymbol{e_m}} \otimes \prod_{i\in\mathcal{I}_1 \otimes} \left[1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}\right] \otimes \prod_{i\in\mathcal{I}_2 \otimes} \left[1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}\right]}{\prod_{i\in\mathcal{I}_1 \otimes} \left[1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}\right] \otimes \prod_{i\in\mathcal{I}_2 \otimes} \left[1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}\right]} = \frac{\lambda_m \otimes A_{\mathcal{C}_1} \otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_1} \otimes A_{\mathcal{C}_2}} \tag{7}$$

Theorem 5 *[10] Let*

$$\lambda_m^{\mathcal{C}_1} = \frac{\lambda_m \otimes \prod_{i\in\mathcal{I}_1 \otimes} \left[1+\prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod_{i\in\mathcal{I}_1 \otimes} \left[1+\prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right]} \tag{8}$$

be the soft output of the bit optimal decoder for code $\mathcal{C}_1$, and

$$\lambda_j^{\mathcal{C}_2} = \frac{\lambda_j \otimes \prod_{i\in\mathcal{I}_2 \otimes} \left[1+\prod_{l=0}^{n-1} \lambda_l^{h_{il}}\right]}{\prod_{i\in\mathcal{I}_2 \otimes} \left[1+\prod_{l=0}^{n-1} \lambda_l^{h_{il}}\right]} = \frac{\lambda_j \otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_2}} \tag{9}$$

be the soft outputs of the bit optimal decoder for code $\mathcal{C}_2$, Then if we replace (update) each λ_j in equation (8) by $\lambda_j^{\mathcal{C}_2}$ given by (9), and change the real product in equation (8) by the group algebra product, we obtain the soft output of the bit optimal decoder for code $\mathcal{C}$, given by (7):

$$\lambda_m^{\mathcal{C}} = \frac{\lambda_m^{\mathcal{C}_2} \otimes \prod_{i\in\mathcal{I}_1 \otimes} \left[1+\prod_{j=0\otimes}^{n-1} (\lambda_j^{\mathcal{C}_2})^{h_{ij}}\right]}{\prod_{i\in\mathcal{I}_1 \otimes} \left[1+\prod_{j=0\otimes}^{n-1} (\lambda_j^{\mathcal{C}_2})^{h_{ij}}\right]}. \tag{10}$$

Remark 6 *The optimal message passing (10) from decoder of $\mathcal{C}_2$ to decoder of $\mathcal{C}_1$ does not simplify the computations required for the optimal solution (7). It only shows how the information from decoder of $\mathcal{C}_2$ should be used by decoder of $\mathcal{C}_1$ to obtain the optimal solution, which we need as a reference for comparison with suboptimal algorithms. It also shows that belief propagation is optimal when the real product of soft outputs is changed into the group algebra product.*

3.2 Codes Whose Tanner Graphs are Trees

Let $\mathcal{C}'_1$ and $\mathcal{C}'_2$ be two subcodes of the dual code $\mathcal{C}'$ obtained by partitioning its set of generators $\boldsymbol{h}_i$, $0 \leq i < n-k$. Let $\mathcal{I}$ be the index set $\mathcal{I} = \{0, 1, \ldots, n-k-1\}$, and $\mathcal{I}_1$ and $\mathcal{I}_2$ its two disjoint subsets obtained by *cutting* through the l-th variable node of the Tanner graph of $\mathcal{C}'$. Thus $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$, and each $\mathcal{C}'_j \subset \mathcal{C}'$ is defined by its set of generators $\boldsymbol{h}_{i_j}$, $i_j \in \mathcal{I}_j$, $j = 1, 2$. This is possible since the Tanner graph of $\mathcal{C}'$ is a tree. The resulting Tanner graphs of $\mathcal{C}'_1$ and $\mathcal{C}'_2$ have only the variable node l in common.

The following theorem establishes the result of Pearl [2] that the belief propagation is optimal when the Tanner graph of the code is a tree.

Theorem 7 *Let variable node m belong to the Tanner graph of $\mathcal{C}'_1$, and*

$$\lambda_m^{\mathcal{C}_1} = \frac{\lambda_m \otimes \prod_{i \in \mathcal{I}_1 \otimes} \Big[1 + \lambda_l^{h_{il}} \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{h_{ij}}\Big]}{\prod_{i \in \mathcal{I}_1 \otimes} \Big[1 + \lambda_l^{h_{il}} \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{h_{ij}}\Big]} = \frac{\lambda_m \otimes \sum_{\boldsymbol{c'} \in \mathcal{C}'_1} \lambda_l^{c'_l} \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{c'_j}}{\sum_{\boldsymbol{c'} \in \mathcal{C}'_1} \lambda_l^{c'_l} \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{c'_j}} \tag{11}$$

be the soft output of the bit optimal decoder for code $\mathcal{C}_1$, and

$$\lambda_l^{\mathcal{C}_2} = \frac{\lambda_l \otimes \prod_{i \in \mathcal{I}_2 \otimes} \Big[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\Big]}{\prod_{i \in \mathcal{I}_2 \otimes} \Big[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\Big]} = \frac{\lambda_l \otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_2}}. \tag{12}$$

be the soft output of the bit optimal decoder for code $\mathcal{C}_2$, Then if we replace (update) λ_l in equation (11) by $\lambda_l^{\mathcal{C}_2}$ given by (12), we obtain $\lambda_m^{\mathcal{C}}$.

PROOF. By substituting $\lambda_l^{\mathcal{C}_2}$ for λ_l in equation (11), we obtain

$$[\lambda_m^{\mathcal{C}_1}]_2 = \frac{\lambda_m \otimes \sum_{\boldsymbol{c'} \in \mathcal{C}'_1} \Big(\frac{\lambda_l \otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_2}}\Big)^{c'_l} \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{c'_j}}{\sum_{\boldsymbol{c'} \in \mathcal{C}'_1} \Big(\frac{\lambda_l \otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_2}}\Big)^{c'_l} \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{c'_j}}$$

Now let $\mathcal{C}'_{1,0}$ denote the set of codewords $\boldsymbol{c'}$ with $c'_l = 0$ and $\mathcal{C}'_{1,1}$ the set of

codewords $c'_l = 1$. We have

$$\begin{aligned}
\sum_{\boldsymbol{c'}\in\mathcal{C}'_1}\Big(\frac{\lambda_l\otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_2}}\Big)^{c'_l}\prod_{\substack{j=0\\ j\neq l}}^{n-1}\lambda_j^{c'_j} &= \sum_{\boldsymbol{c'}\in\mathcal{C}'_{1,0}}\prod_{j=0}^{n-1}\lambda_j^{c'_j}+\sum_{\boldsymbol{c'}\in\mathcal{C}'_{1,1}}\Big(\frac{\lambda_l\otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_2}}\Big)\prod_{\substack{j=0\\ j\neq l}}^{n-1}\lambda_j^{c'_j}\\
&= \frac{1}{A_{\mathcal{C}_2}}\Big[A_{\mathcal{C}_2}\cdot\sum_{\boldsymbol{c'}\in\mathcal{C}'_{1,0}}\prod_{j=0}^{n-1}\lambda_j^{c'_j}+(\lambda_l\otimes A_{\mathcal{C}_2})\sum_{\boldsymbol{c'}\in\mathcal{C}'_{1,0}}\prod_{\substack{j=0\\ j\neq l}}^{n-1}\lambda_j^{c'_j}\Big]\\
&= \frac{1}{A_{\mathcal{C}_2}}\cdot A_{\mathcal{C}_2}\otimes\Big[\sum_{\boldsymbol{c'}\in\mathcal{C}'_{1,0}}\prod_{j=0}^{n-1}\lambda_j^{c'_j}+\sum_{\boldsymbol{c'}\in\mathcal{C}'_{1,1}}\prod_{j=0}^{n-1}\lambda_j^{c'_j}\Big] \quad (13)\\
&= \frac{1}{A_{\mathcal{C}_2}}(A_{\mathcal{C}_2}\otimes A_{\mathcal{C}_1}),
\end{aligned}$$

where (13) follows from Lemma 3. Thus

$$[\lambda_m^{\mathcal{C}_1}]_2 = \frac{\lambda_m\otimes A_{\mathcal{C}_1}\otimes A_{\mathcal{C}_2}}{A_{\mathcal{C}_1}\otimes A_{\mathcal{C}_2}},$$

which is precisely (7). Therefore $\lambda_m^{\mathcal{C}} = [\lambda_m^{\mathcal{C}_1}]_2$, and the optimal solution is reached in 2 steps. $\square$

4 Low Density Parity-Check Codes

We illustrate Gallager's belief propagation algorithm in our group algebra terminology on a special one-step case. Note first that the expression for the log-likelihood of the optimal decoder given by (5) can be re-written as

$$L_m + \log\frac{\prod_{\substack{i=0\\ h_{im}=0}}^{n-k-1}{}_{\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]\otimes\prod_{\substack{i=0\\ h_{im}=1}}^{n-k-1}{}_{\otimes}\Big[1+\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod_{\substack{i=0\\ h_{im}=0}}^{n-k-1}{}_{\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]\otimes\prod_{\substack{i=0\\ h_{im}=1}}^{n-k-1}{}_{\otimes}\Big[1-\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}. \quad (14)$$

Now, let $\mathcal{C}$ be an LDPC code such that, for each pair of bits m and l, there is at most one common parity check equation, *i.e.,* there is at most one i, $0\le i\le n-k-1$ for which $h_{im}=1$ and $h_{il}=1$. In the Tanner graph terminology, we say that, for any two variable nodes m and l, there is at most one common check node $c(m,l)$, *i.e.,* there are no loops of length 4.

The optimal soft output for bit m based solely on the channel information L_m and the knowledge of the parity check equations in which it participates is given by

$$L_m + \log \frac{\prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left(1 + \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right)}{\prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left(1 - \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right)}. \tag{15}$$

Because of the low-density assumption (no loops of length 4), the group multiplication above is equal to the real multiplication. Therefore, in the first iteration the soft output of the Gallager's algorithm is equal to the soft output (15), which is optimal for the code defined by the parity check equations in which bit m participates.

Let λ_{ml} denote the message passed from variable node l to variable node m through their unique common check node $c(m,l)$, *i.e.,*

$$\lambda_{ml} = \tanh(L_{ml}/2),$$

where (as proposed by Gallager)

$$L_{ml} = \log \frac{1+\lambda_l}{1-\lambda_l} + \log \frac{\prod\limits_{\substack{i=0 \\ h_{il}=1 \\ i\neq c(m,l)}}^{n-k-1} \left(1 + \prod\limits_{\substack{j=0 \\ j\neq l}}^{n-1} \lambda_j^{h_{ij}}\right)}{\prod\limits_{\substack{i=0 \\ h_{il}=1 \\ i\neq c(m,l)}}^{n-k-1} \left(1 - \prod\limits_{\substack{j=0 \\ j\neq l}}^{n-1} \lambda_j^{h_{ij}}\right)}.$$

Using the low density assumption and Theorem 7, we obtain

$$\lambda_{ml} = \frac{\lambda_l \otimes \prod\limits_{\substack{i=0 \\ h_{im}=0, h_{il}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod\limits_{\substack{i=0 \\ h_{im}=0, h_{il}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\right]}.$$

Upon receiving this message, the update for bit m is computed by (15) with

λ_{ml} in the place of λ_l, giving the following

$$L_m + \log \frac{\prod\limits_{\substack{i=0 \\ h_{im}=0,h_{il}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod\limits_{\substack{i=0 \\ h_{im}=0,h_{il}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 - \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}. \tag{16}$$

At this point the soft output of the Gallager's algorithm, given by (16), is the optimal soft output for the code defined by the parity check equations in which both bit m and bit l participate. This brings us a step *closer* to the expression for optimal decoding of bit m given by (14), which is based on all parity checks.

Example 2 *Gallager's decoding of bit $m = 0$ for the $(8, 4)$ tree code in Fig. 1:*

initialization:

$$L_j = \log \frac{p(r_j|0)}{p(r_j|1)} \quad \textit{and} \quad \lambda_j = \frac{p(r_j|0) - p(r_j|1)}{p(r_j|0) + p(r_j|1)} = \tanh(L_j/2)$$

iteration #1

$$[L_0]_1 = L_0 + \log \frac{(1+\lambda_7)\cdot(1+\lambda_1\lambda_4)}{(1-\lambda_7)\cdot(1-\lambda_1\lambda_4)} \tag{17}$$

$$[L_{01}]_1 = L_1 + \log \frac{1+\lambda_2\lambda_5}{1-\lambda_2\lambda_5} \textit{ and } [\lambda_{01}]_1 = \frac{\lambda_1+\lambda_2\lambda_5}{1+\lambda_1\lambda_2\lambda_5} \tag{18}$$

$$[L_{12}]_1 = L_2 + \log \frac{1+\lambda_3\lambda_6}{1-\lambda_3\lambda_6} \textit{ and } [\lambda_{12}]_1 = \frac{\lambda_2+\lambda_3\lambda_6}{1+\lambda_2\lambda_3\lambda_6}$$

iteration #2

Substituting $[\lambda_{01}]_1$ for λ_1 in (17), and $[\lambda_{12}]_1$ for λ_2 in (18):

$$\begin{aligned}
[L_0]_2 &= L_0 + \log \frac{(1+\lambda_7)\cdot(1+\lambda_1\lambda_2\lambda_5+\lambda_1\lambda_4+\lambda_2\lambda_4\lambda_5)}{(1-\lambda_7)\cdot(1+\lambda_1\lambda_2\lambda_5-\lambda_1\lambda_4-\lambda_2\lambda_4\lambda_5)} \\
&= L_0 + \log \frac{(1+\lambda_7)\otimes(1+\lambda_1\lambda_4)\otimes(1+\lambda_1\lambda_2\lambda_5)}{(1-\lambda_7)\otimes(1-\lambda_1\lambda_4)\otimes(1+\lambda_1\lambda_2\lambda_5)} \\
[L_{01}]_2 &= L_1 + \log \frac{1+\lambda_2\lambda_3\lambda_6+\lambda_2\lambda_5+\lambda_3\lambda_5\lambda_6}{1+\lambda_2\lambda_3\lambda_6-\lambda_2\lambda_5-\lambda_3\lambda_5\lambda_6} \\
&= L_0 + \log \frac{(1+\lambda_2\lambda_5)\otimes(1+\lambda_2\lambda_3\lambda_6)}{(1-\lambda_2\lambda_5)\otimes(1+\lambda_2\lambda_3\lambda_6)} \\
[\lambda_{01}]_2 &= \frac{\lambda_1+\lambda_2\lambda_5+\lambda_1\lambda_2\lambda_3\lambda_6+\lambda_3\lambda_5\lambda_6}{1+\lambda_1\lambda_2\lambda_5+\lambda_2\lambda_3\lambda_6+\lambda_1\lambda_3\lambda_5\lambda_6}
\end{aligned}$$

iteration #3
Substituting $[\lambda_{01}]_2$ *for* λ_1 *in (17):*

$$[L_0]_3 = L_0 + \log \frac{(1+\lambda_7) \otimes (1+\lambda_1\lambda_4) \otimes (1+\lambda_1\lambda_2\lambda_5) \otimes (1+\lambda_2\lambda_3\lambda_6)}{(1-\lambda_7) \otimes (1-\lambda_1\lambda_4) \otimes (1+\lambda_1\lambda_2\lambda_5) \otimes (1+\lambda_2\lambda_3\lambda_6)}$$

Thus in iteration #3 we attain the log-likelihood of the optimal decoder.

5 Conclusions

After the rediscovery of LDPC codes, their decoding algorithms have been studied by mathematical tools pertinent more to computer science than the conventional algebraic coding theory. We proposed an entirely new approach to this problem in [10], which gives new insights into several issues of iterative decoding. Here we introduced a group algebra formulation for bit optimal decoding of parity check codes. In particular, we used this new framework to give simple algebraic proofs that Pearl's and Gallager's belief propagation decoding algorithms are bit-optimal when the Tanner graph of the code is a tree.

References

[1] R. M. Tanner, "A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory*, Vol. IT-27, pp. 533–547, Sep. 1981.

[2] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial Intelligence*, Vol. 29, pp. 241–288, 1986.

[3] N. Wiberg, H-A. Loeliger, and R. Kötter, "Codes and iterative decoding on general graphs," *Europ. Trans. Telecomm.,* vol. 6, pp. 513–525, Sept/Oct. 1995.

[4] S. M. Aji and R. J. McEliece, "The generalized distributive Law," *IEEE Trans. Inform. Theory,* Vol. 46, pp. 325–343, March 2000.

[5] F. R. Kschischang, B. J. Frey, and H-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, pp. 498–519, Feb. 2001.

[6] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, pp. 599–618, Feb. 2001.

[7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes* Amsterdam: North-Holland, 1996.

[8] G. Battail, M. C. Decouvelaere, and P. J. Godlewski, "Replication decoding," *IEEE Trans. Inform. Theory*, pp. 332–345, May 1979.

[9] C. R. P. Hartmann and L. D. Rudolph, "An optimum symbol by symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, Vol. IT-22, pp. 514–517, Sept. 1976.

[10] E. Offer and E. Soljanin, "An algebraic description of iterative decoding schemes," *Codes, Systems, and Graphical Models,* IMA Volumes in Mathematics and its Applications v. 123, Springer-Verlag, 2001, B. Marcus and J. Rosenthal, eds.

[11] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, Vol. IT-8, pp. 21–28, Jan. 1962.