AN ALGEBRAIC DESCRIPTION OF ITERATIVE DECODING SCHEMES*

ELKE OFFER† AND EMINA SOLJANIN‡

Abstract. Several popular, suboptimal algorithms for bit decoding of binary block codes such as turbo decoding, threshold decoding, and message passing for LDPC, were developed almost as a *common sense* approach to decoding of some specially designed codes. After their introduction, these algorithms have been studied by mathematical tools pertinent more to computer science than the conventional algebraic coding theory. We give an algebraic description of the optimal and suboptimal bit decoders and of the optimal and suboptimal message passing. We explain exactly how suboptimal algorithms approximate the optimal, and show how good this approximations are in some special cases.

Key words. Iterative decoding, soft-output decoding, suboptimal decoding, bit-optimal decoding.

AMS(MOS) subject classifications. 94B05, 94B25, 94B35, 94B60.

1. Introduction. We propose an entirely new approach to the problem of iterative decoding, which is algebraic in nature and *derives* the well known suboptimal algorithms from the bit-optimal as a starting point. The approach gives new insights into the issues of iterative decoding, and we hope will make the area more accessible to the algebraic coding theorists.

We are concerned with a binary block code $\mathcal{C}$ defined by its parity-check matrix $H = \{h_{ij}\}_{(n-k)\times n}$, *i.e.,* by the group generators $\boldsymbol{h}_i = \{h_{ij}\}_{1\times n}$, $i \in \mathcal{I}$, of the dual code $\mathcal{C}'$, where $\mathcal{I}$ is used to denote the index set $\mathcal{I} = \{0, 1, \ldots, n-k-1\}$. The channel is assumed to be memoryless.

The bit-optimal decoder computes the probability that the bit at position m is equal to $b \in \{0, 1\}$ given the received word $\boldsymbol{r}$. To do this, it computes and adds probabilities (given $\boldsymbol{r}$) of each codeword in the code $\mathcal{C}$ with b at position m. It can equivalently compute the Fourier Transform of the probabilities and add them up over the dual code $\mathcal{C}'$. This equivalence was first shown by Hartmann and Rudolph in [1], and we point out that it is merely based on the Poisson summation formula. Working with the the dual code is in practice preferred when $\mathcal{C}'$ has fewer codewords than $\mathcal{C}$, *i.e.,* when dealing with high rate codes.

Since the suboptimal algorithms we are concerned with (turbo decoding, threshold decoding, and message passing for LDPC) operate with the generators $\boldsymbol{h}_i$ of the dual code, we derive an expression for optimal decod-

* This work was supported by the 1999 German-American Networking Research Grant given by the the national academies of engineering of Germany and the USA

†Institute for Communications Engineering, Munich University of Technology, D-80290 Munich, Germany, elke@lnt.e-technik.tu-muenchen.de

‡Mathematical Sciences Research Center, Bell Labs, Lucent Technologies, Murray Hill, NJ 07974, USA. emina@lucent.com

ing based on the dual code and then rewrite it so that it explicitly involves only $\boldsymbol{h}_i$. Since sub optimal algorithms operate on subcodes of the dual code, we define L subcodes of $\mathcal{C}'$ by partitioning its set of generators $\boldsymbol{h}_i$, $i \in \mathcal{I}$. Thus each $\mathcal{C}'_l \subset \mathcal{C}'$ is defined by its set of generators $\boldsymbol{h}_{i_l}$, $i_l \in \mathcal{I}_l$, $\mathcal{I}_l \subset I$, where the index sets $\mathcal{I}_l$ are disjoint and $\cup_{l=1}^{L} \mathcal{I}_l = \mathcal{I}$. We now expand our expression for optimal decoding to show the participation of each subcode $\mathcal{C}'_l$. We use this expression to point out the approximation each of the suboptimal schemes performs.

2. Bit Optimal Decoding. Let block code $\mathcal{C}$ be a subgroup of the additive group of $\mathbb{F}_2^n$. 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 $\boldsymbol{r}$. This probability, given by

$$P(c_m = b|\boldsymbol{r}) = \sum_{\boldsymbol{c}\in\mathcal{C}, c_m=b} P(\boldsymbol{c}|\boldsymbol{r}),$$

can be expressed as

$$P(c_m = b|\boldsymbol{r}) = \sum_{\boldsymbol{c}\in\mathcal{C}} \frac{P(\boldsymbol{c})}{p(\boldsymbol{r})} p(\boldsymbol{r}|\boldsymbol{c}) \delta_{b,\boldsymbol{c}\cdot\boldsymbol{e_m}}.$$

Therefore the log-likelihood of bit m over code $\mathcal{C}$, $L_m^{\mathcal{C}}$, is given by

$$L_m^{\mathcal{C}} = \log \frac{P(c_m = 0|\boldsymbol{r})}{P(c_m = 1|\boldsymbol{r})} = \log \frac{\sum_{\boldsymbol{c}\in\mathcal{C}} p(\boldsymbol{r}|\boldsymbol{c}) \delta_{0,\boldsymbol{c}\cdot\boldsymbol{e_m}}}{\sum_{\boldsymbol{c}\in\mathcal{C}} p(\boldsymbol{r}|\boldsymbol{c}) \delta_{1,\boldsymbol{c}\cdot\boldsymbol{e_m}}} \tag{2.1}$$

when the codewords are equiprobable.

Let $f : \mathbb{F}_2^n \to \mathbb{R}$ be a real valued function defined on $\mathbb{F}_2^n$ and $\hat{f}$ its DFT on the additive group $\mathbb{F}_2^n$, *i.e.*,

$$\hat{f}(\boldsymbol{v}) = \sum_{u\in\mathbb{F}_2^n} f(\boldsymbol{u})(-1)^{\boldsymbol{u}\cdot\boldsymbol{v}}.$$

Then, by the Poisson summation formula

$$\frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{c}\in\mathcal{C}} f(\boldsymbol{c}) = \frac{1}{|\mathbb{F}_2^n|} \sum_{\boldsymbol{c}'\in\mathcal{C}'} \hat{f}(\boldsymbol{c}'), \tag{2.2}$$

where $\mathcal{C}'$ is the dual of $\mathcal{C}$ in $\mathbb{F}_2^n$. For $f(\boldsymbol{u})$, $\boldsymbol{u} \in \mathbb{F}_2^n$, defined as

$$f(\boldsymbol{u}) = p(\boldsymbol{r}|\boldsymbol{u}) \delta_{b,\boldsymbol{u}\cdot\boldsymbol{e_m}}$$

and $p(\boldsymbol{r}|\boldsymbol{u}) = \prod_{j=1}^{n} p(r_j|u_j)$, it follows that $\hat{f}(\boldsymbol{v})$, $\boldsymbol{v} \in \mathbb{F}_2^n$, is given by

$$\begin{aligned} \hat{f}(\boldsymbol{v}) &= \sum_{u\in\mathbb{F}_2^n} \delta_{b,u_m} \prod_{j=1}^{n} p(r_j|u_j)(-1)^{u_j v_j} \\ &= p(r_m|b)(-1)^{b v_m} \prod_{\substack{j=1 \\ j\neq m}}^{n} \big[p(r_j|0) + p(r_j|1)(-1)^{v_j}\big]. \end{aligned} \tag{2.3}$$

By using (2.2) with $\hat{f}$ defined by (2.3), equation (2.1) can be expressed as

$$L_m^{\mathcal{C}} = \log \frac{p(r_m|0)}{p(r_m|1)} + \log \frac{\displaystyle\sum_{i=0}^{2^{n-k}-1} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \Big(\frac{p(r_j|0)-p(r_j|1)}{p(r_j|0)+p(r_j|1)}\Big)^{c'_{ij}}}{\displaystyle\sum_{i=0}^{2^{n-k}-1} (-1)^{c'_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \Big(\frac{p(r_j|0)-p(r_j|1)}{p(r_i|0)+p(r_j|1)}\Big)^{c'_{ij}}}, \quad (2.4)$$

which is basically the result of Hartmann and Rudolph derived and presented somewhat differently in [1].

Note that expression (2.4) explicitly involves all codewords of the dual code $\mathcal{C}'$. Recall that suboptimal algorithms we are concerned with (turbo decoding, threshold decoding, and message passing for LDPC) operate with the parity-check matrix $H = \{h_{ij}\}_{(n-k)\times n}$, *i.e.,* with the generators (independent parity checks) of the dual code, $\boldsymbol{h}_i = \{h_{ij}\}_{1\times n}$, $0 \le i < n-k$. Because of that, we derive an expression for optimal decoding (2.4) which explicitly involves only $\boldsymbol{h}_i$. We denote

$$\lambda_j = \frac{p(r_j|0)-p(r_j|1)}{p(r_j|0)+p(r_j|1)} = \tanh(L_j/2), \quad (2.5)$$

which when substituted in (2.4) gives

$$L_m^{\mathcal{C}} = \log \frac{1+\lambda_m}{1-\lambda_m} + \log \frac{\displaystyle\sum_{i=0}^{2^{n-k}-1} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{c'_{ij}}}{\displaystyle\sum_{i=0}^{2^{n-k}-1} (-1)^{c'_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{c'_{ij}}}. \quad (2.6)$$

We introduce special multiplication $\otimes$ as follows:

$$\lambda_i \otimes \lambda_j = (\lambda_i \cdot \lambda_j)^{1-\delta_{i,j}} = \begin{cases} \lambda_i \cdot \lambda_j, & i \neq j, \\ 1, & i = j. \end{cases}$$

It is easy to see that equation (2.4) can be formally written as

$$L_m^{\mathcal{C}} = \log \frac{1+\lambda_m}{1-\lambda_m} + \log \frac{\displaystyle\prod_{i\in\mathcal{I}}{}_{\otimes} \Big[1 + \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}{\displaystyle\prod_{i\in\mathcal{I}}{}_{\otimes} \Big[1 + (-1)^{h_{im}} \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}, \quad (2.7)$$

where $\mathcal{I}$ is used to denote the index set $\mathcal{I} = \{0, 1, \ldots, n-k-1\}$. We now derive a formal companion expression for $\lambda_m^{\mathcal{C}} = \tanh(L_m^{\mathcal{C}}/2)$.

THEOREM 2.1. *Let λ_j defined by (2.5) 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*

$$\lambda_m^{\mathcal{C}} = \tanh(L_m^{\mathcal{C}}/2) = \frac{\lambda_m \otimes \prod_{i\in\mathcal{I}\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}. \tag{2.8}$$

Proof. Let

$$A_m^+ = \prod_{i\in\mathcal{I}\otimes}\Big[1+\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big] \quad \text{and} \quad A_m^- = \prod_{i\in\mathcal{I}\otimes}\Big[1+(-1)^{h_{im}}\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big].$$

Then

$$L_m^{\mathcal{C}} = \log\frac{(1+\lambda_m)A_m^+}{(1-\lambda_m)A_m^-}.$$

By using the definition $\tanh(x/2) = (e^x+1)/(e^x-1)$, we get

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

Note that $(A_m^+ + A_m^-)/2$ represents the summation over all dual codewords with a '0' at position m, where $\lambda_m(A_m^+ - A_m^-)/2$ is the summation over all dual codewords with a '1' at position m. Therefore,

$$\lambda_m^{\mathcal{C}} = \frac{\lambda_m \otimes [(A_m^+ + A_m^-) + \lambda_m(A_m^+ - A_m^-)]}{(A_m^+ + A_m^-) + \lambda_m(A_m^+ - A_m^-)} = \frac{\lambda_m \otimes \sum_{i=0}^{2^{n-k}-1}\prod_{j=0}^{n-1}\lambda_j^{c'_{ij}}}{\sum_{i=0}^{2^{n-k}-1}\prod_{j=0}^{n-1}\lambda_j^{c'_{ij}}}.$$

which can formally be written as (2.8). □

3. Optimal Message Passing Bit Decoders. We define L subcodes of the dual code $\mathcal{C}'$ by partitioning its set of generators $\boldsymbol{h}_i$, $i \in \mathcal{I}$. Thus each $\mathcal{C}'_l \subset \mathcal{C}'$ is defined by its set of generators $\boldsymbol{h}_{i_l}$, $i_l \in \mathcal{I}_l$, $\mathcal{I}_l \subset \mathcal{I}$,

where the index sets $\mathcal{I}_l$ are disjoint and $\cup_{l=1}^{L}\mathcal{I}_l = \mathcal{I}$. Note that in this scenario equation (2.8) becomes

$$\lambda_m^{\mathcal{C}} = \frac{\lambda_m \otimes \prod_{i\in\mathcal{I}_1\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]\otimes\cdots\otimes\prod_{i\in\mathcal{I}_L\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_1\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]\otimes\cdots\otimes\prod_{i\in\mathcal{I}_L\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}. \tag{3.1}$$

THEOREM 3.1. *Let $\mathcal{C}_1'^M$ be the subcode of $\mathcal{C}'$ defined by its index set $\mathcal{I}_1^M = \cup_{l=1}^{M}\mathcal{I}_l$. Then*

$$\lambda_m^{\mathcal{C}_1^{M+1}} = \frac{\lambda_m^{\mathcal{C}_1^M} \otimes \prod_{i\in\mathcal{I}_{M+1}\otimes}\Big[1+\prod_{j=0\otimes}^{n-1}(\lambda_j^{\mathcal{C}_1^M})^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_{M+1}\otimes}\Big[1+\prod_{j=0\otimes}^{n-1}(\lambda_j^{\mathcal{C}_1^M})^{h_{ij}}\Big]}. \tag{3.2}$$

Proof. Without loss of generality, we assume $M = 2$. Thus $\mathcal{C}_1^1 = \mathcal{C}_1$ and $\mathcal{C}_1^2 = \mathcal{C}$. From equation (2.8), for C_1 and C_2 we have

$$\lambda_j^{\mathcal{C}_1} = \frac{\lambda_j \otimes \prod_{i\in\mathcal{I}_1\otimes}\Big[1+\prod_{l=0}^{n-1}\lambda_l^{h_{il}}\Big]}{\prod_{i\in\mathcal{I}_1\otimes}\Big[1+\prod_{l=0}^{n-1}\lambda_l^{h_{il}}\Big]} = \frac{\lambda_j \otimes A_{\mathcal{C}_1}}{A_{\mathcal{C}_1}} \tag{3.3}$$

and

$$\lambda_m^{\mathcal{C}_2} = \frac{\lambda_m \otimes \prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}. \tag{3.4}$$

where $A_{\mathcal{C}_1} = \prod_{\otimes i\in\mathcal{I}_1}\Big[1+\prod_{l=0}^{n-1}\lambda_l^{h_{il}}\Big]$. We replace the priors λ_j in equation (3.4) by $\lambda_j^{\mathcal{C}_1}$ given by (3.3), and change the real product in equation (3.4) by the special multiplication product, thus obtaining

$$[\lambda_m^{\mathcal{C}_2}]_2 = \frac{\frac{\lambda_m \otimes A_{\mathcal{C}_1}}{A_{\mathcal{C}_1}} \otimes \prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0\otimes}^{n-1}\Big(\frac{\lambda_j \otimes A_{\mathcal{C}_1}}{A_{\mathcal{C}_1}}\Big)^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0\otimes}^{n-1}\Big(\frac{\lambda_j \otimes A_{\mathcal{C}_1}}{A_{\mathcal{C}_1}}\Big)^{h_{ij}}\Big]}, \tag{3.5}$$

where $[\lambda_m^{\mathcal{C}_2}]_2$ denotes the result for bit m of the *second iteration* in decoding of $\mathcal{C}_2$. We can further transform the above equation by taking into account that

$$\underbrace{A_{\mathcal{C}_1} \otimes A_{\mathcal{C}_1} \otimes \cdots \otimes A_{\mathcal{C}_1}}_{\nu} = 2^{(\nu-1)\cdot|\mathcal{I}_1|} \cdot A_{\mathcal{C}_1} \tag{3.6}$$

and obtain

$$[\lambda_m^{\mathcal{C}_2}]_2 = \frac{\lambda_m \otimes \prod\limits_{i\in\mathcal{I}_1\otimes}\Big[1+\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big] \otimes \prod\limits_{i\in\mathcal{I}_2\otimes}\Big[1+\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod\limits_{i\in\mathcal{I}_1\otimes}\Big[1+\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big] \otimes \prod\limits_{i\in\mathcal{I}_2\otimes}\Big[1+\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}. \tag{3.7}$$

which is precisely (3.1) for $M = 2$. Therefore $\lambda_{\mathcal{C}} = [\lambda_m^{\mathcal{C}_2}]_2$, and the optimal solution is reached in 2 steps. □

With optimal message passing in the L subcode case, we obtain $\lambda_m^{\mathcal{C}}$ in L steps. We start with computing $\lambda_m^{\mathcal{C}_1}$ by using (3.3), and continue with computing $\lambda_m^{\mathcal{C}_1^{M+1}}$ by optimal message passing (3.2) for $1 \le M \le L$.

REMARK 3.1. *The optimal message passing (3.5) from decoder of $\mathcal{C}_1$ to decoder of $\mathcal{C}_2$ doesn't simplify the computations required for the optimal solution (3.7). It only shows how the information from decoder of $\mathcal{C}_1$ should be used by decoder of $\mathcal{C}_2$ to obtain the optimal solution, which we need as a reference for comparison with suboptimal algorithms.*

REMARK 3.2. *Once $\lambda_m^{\mathcal{C}}$ is computed, further iterations will leave it unchanged. Namely, $\lambda_j^{\mathcal{C}}$ used as new priors to compute $[\lambda_m^{\mathcal{C}_1}]_2$ will not change its value, starting with $\mathcal{C}_1$ and so on. This is again easily shown by using equalities of type (3.6).*

In what follows, we consider several popular, suboptimal bit detection algorithms. For each algorithm, we give an expression of (2.4) or (2.8) which is the most suitable for the analysis of its relation to the optimal algorithm.

4. Turbo Codes. We consider a turbo coding scheme with two component codes $\mathcal{C}_1$ and $\mathcal{C}_2$ as introduced in [2], [3]. The bit-optimal decoder computes the log-likelihood for bit m by (2.7), which can be expressed as

$$L_m^{\mathcal{C}} = \log\frac{1-\lambda_m}{1+\lambda_m} + \log\frac{\prod\limits_{i\in\mathcal{I}_1\otimes}\Big[1+\prod\limits_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big] \otimes \prod\limits_{i,\mathcal{I}_2\otimes}\Big[1+\prod\limits_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod\limits_{i,\in\mathcal{I}_1\otimes}\Big[1+(-1)^{h_{im}}\prod\limits_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big] \otimes \prod\limits_{i\in\mathcal{I}_2\otimes}\Big[1+(-1)^{h_{im}}\prod\limits_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}.$$

The corresponding soft output $\lambda_m^{\mathcal{C}}$ can be written as

$$\lambda_m^{\mathcal{C}} = \frac{\lambda_m \otimes \prod_{i\in\mathcal{I}_1}{}_{\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big] \otimes \prod_{i\in\mathcal{I}_2}{}_{\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_1}{}_{\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big] \otimes \prod_{i\in\mathcal{I}_2}{}_{\otimes}\Big[1+\prod_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}. \tag{4.1}$$

As described in the previous section, the optimal decoding (4.1) can be performed sequentially. First, the optimal decoder of $\mathcal{C}_1$ computes soft information (3.3). Then, decoder of $\mathcal{C}_2$ uses the soft information in the optimal message passing equation (3.5) to obtain the optimal solution (4.1).

Turbo decoding also starts with the optimal decoding of $\mathcal{C}_1$ by computing the *first iteration* log-likelihood for bit m as

$$[L_m^{\mathcal{C}_1}]_1 = \log\frac{1+\lambda_m}{1-\lambda_m} + \underbrace{\log\frac{\prod_{i\in\mathcal{I}_1}{}_{\otimes}\Big[1+\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_1}{}_{\otimes}\Big[1+\lambda_m^{h_{im}}\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}}_{[L_{m,e}^{\mathcal{C}_1}]_1} \tag{4.2}$$

with corresponding

$$[\lambda_m^{\mathcal{C}_1}]_1 = \tanh([L_m^{\mathcal{C}_1}]_1/2) = \frac{\lambda_m \otimes \prod_{i\in\mathcal{I}_1}{}_{\otimes}\Big[1+\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_1}{}_{\otimes}\Big[1+\prod_{\substack{j=0\\ j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big]}.$$

The second term in equation (4.2) is known as (first iteration) *extrinsic* information of decoder $\mathcal{C}_1$ for bit m. It is denoted by $[L_{m,e}^{\mathcal{C}_1}]_1$ with corresponding $[\lambda_{m,e}^{\mathcal{C}_1}]_1 = \tanh([L_{m,e}^{\mathcal{C}_1}]_1/2)$, and transfered to the decoder of $\mathcal{C}_2$ for all bit positions m checked by both codes. Note that

$$[\lambda_m^{\mathcal{C}_1}]_1 = \frac{\lambda_m + [\lambda_{m,e}^{\mathcal{C}_1}]_1}{1+\lambda_m[\lambda_{m,e}^{\mathcal{C}_1}]_1}.$$

Turbo decoding continues with the decoding of $\mathcal{C}_2$ by computing

$$[L_m^{\mathcal{C}_2}]_2 = \log\frac{1+\lambda_m}{1-\lambda_m} + \underbrace{\log\frac{\prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{\substack{j=0\\ j\neq m}}^{n-1}[\lambda_j^{\mathcal{C}_1}]_1^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_2\otimes}\Big[1+(-1)^{h_{im}}\prod_{\substack{j=0\\ j\neq m}}^{n-1}[\lambda_j^{\mathcal{C}_1}]_1^{h_{ij}}\Big]}}_{[L_{m,e}^{\mathcal{C}_2}]_2}, \tag{4.3}$$

where $[\lambda_j^{\mathcal{C}_1}]_1 = \lambda_j$ if position j is checked only by $\mathcal{C}_2$. Now, the extrinsic information $[L_{m,e}^{\mathcal{C}_1}]_2$ is transfered to the decoder of $\mathcal{C}_1$ for the bit positions m checked by both codes, and the process continues iteratively, as illustrated in Fig. 1. In the iteration ν, when component decoder $\mathcal{C}_1$ is active, we

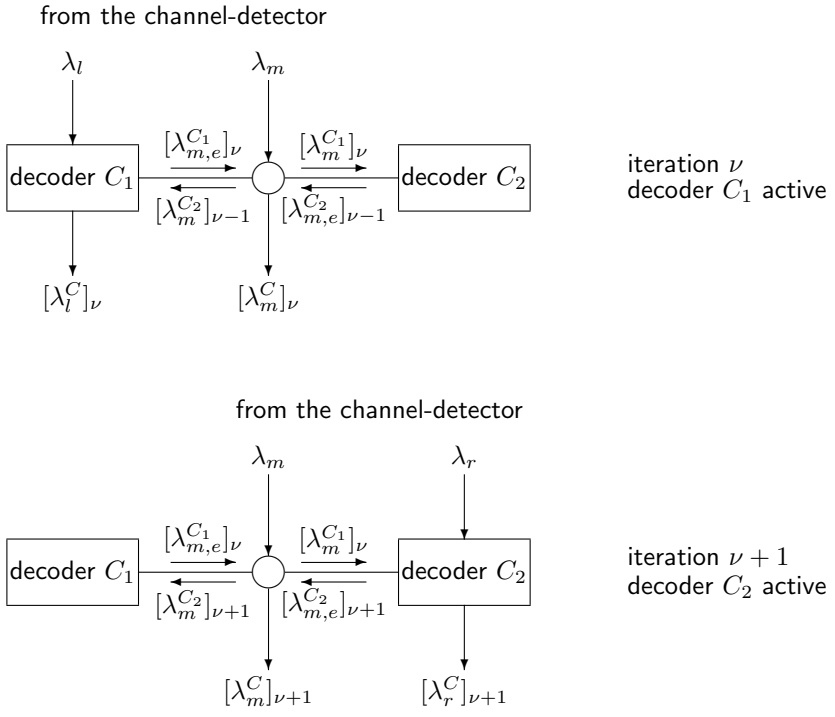


FIG. 1. *Message passing procedure and the corresponding soft-output in a serial turbo decoding procedure. Here the symbol positions checked by both component codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are denoted by m. The symbol positions involved only in the parity check equations of code $\mathcal{C}_1$, respectively $\mathcal{C}_2$, are denoted by r, respectively l.*

have the following scenario: The input information are the channel values λ_r for the bit positions r checked only by code $\mathcal{C}_1$ and channel values

λ_m together with the extrinsic information $[\lambda^{\mathcal{C}_2}_{m,e}]_{\nu-1}$ of code $\mathcal{C}_2$ from the previous iteration for bit positions m checked by both component codes. As described above for the two first iterations, the soft output of decoder $\mathcal{C}_1$ is calculated as

$$[L^{\mathcal{C}_1}_m]_\nu = \log\frac{1+\lambda_m}{1-\lambda_m} + \underbrace{\log\frac{\displaystyle\prod_{i\in\mathcal{I}_1\otimes}\Big[1+\prod_{\substack{j=0\\ j\neq m}}^{n-1}[\lambda^{\mathcal{C}_2}_j]^{h_{ij}}_{\nu-1}\Big]}{\displaystyle\prod_{i\in\mathcal{I}_1\otimes}\Big[1+(-1)^{h_{im}}\prod_{\substack{j=0\\ j\neq m}}^{n-1}[\lambda^{\mathcal{C}_2}_j]^{h_{ij}}_{\nu-1}\Big]}}_{[L^{\mathcal{C}_1}_{m,e}]_\nu}$$

giving

$$[\lambda^{\mathcal{C}_1}_m]_\nu = \frac{\lambda_m \otimes \displaystyle\prod_{i\in\mathcal{I}_1\otimes}\Big[1+\lambda^{h_{im}}_m\prod_{\substack{j=1\\ j\neq m}}^{n-1}[\lambda^{\mathcal{C}_2}_j]^{h_{ij}}_{\nu-1}\Big]}{\displaystyle\prod_{i\in\mathcal{I}_1\otimes}\Big[1+\lambda^{h_{im}}_m\prod_{\substack{j=1\\ j\neq m}}^{n-1}[\lambda^{\mathcal{C}_2}_j]^{h_{ij}}_{\nu-1}\Big]}.$$

The overall soft-output of the decoding scheme after iteration ν is computed by

$$\begin{aligned}[L^{\mathcal{C}}_m]_\nu &= \log\frac{1+\lambda_m}{1-\lambda_m} + [L^{\mathcal{C}_2}_{m,e}]_{\nu-1} + [L^{\mathcal{C}_1}_{m,e}]_\nu\\ &= [L^{\mathcal{C}_2}_m]_{\nu-1} + [L^{\mathcal{C}_1}_{m,e}]_\nu\\ &= [L^{\mathcal{C}_2}_{m,e}]_{\nu-1} + [L^{\mathcal{C}_1}_m]_\nu.\end{aligned}$$

Therefore the corresponding $[\lambda^{\mathcal{C}}_m]_\nu$ can be expressed as

$$[\lambda^{\mathcal{C}}_m]_\nu = \frac{[\lambda^{\mathcal{C}_2}_m]_{\nu-1}\otimes\displaystyle\prod_{i\in\mathcal{I}_1\otimes}\Big[1+\prod_{j=0}^{n-1}[\lambda^{\mathcal{C}_2}_j]^{h_{ij}}_{\nu-1}\Big]}{\displaystyle\prod_{i\in\mathcal{I}_1\otimes}\Big[1+\prod_{j=0}^{n-1}[\lambda^{\mathcal{C}_2}_j]^{h_{ij}}_{\nu-1}\Big]}$$

and, similarly,

$$[\lambda^{\mathcal{C}}_m]_{\nu+1} = \frac{[\lambda^{\mathcal{C}_1}_m]_\nu\otimes\displaystyle\prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0}^{n-1}[\lambda^{\mathcal{C}_1}_j]^{h_{ij}}_{\nu}\Big]}{\displaystyle\prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0}^{n-1}[\lambda^{\mathcal{C}_1}_j]^{h_{ij}}_{\nu}\Big]}.$$

We have thus proved the following:

THEOREM 4.1. *Two-iterations turbo decoding algorithm of [2], [3] approximates the optimal by substituting the term*

$$\prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0}^{n-1}\Big(\lambda_j^{\mathcal{C}_1}\Big)^{h_{ij}}\Big] \tag{4.4}$$

for

$$\prod_{i\in\mathcal{I}_2\otimes}\Big[1+\prod_{j=0\otimes}^{n-1}\Big(\lambda_j^{\mathcal{C}_1}\Big)^{h_{ij}}\Big] \tag{4.5}$$

in the optimal message passing equation (3.5). An example is given in the Appendix.

Expression (4.5) evaluates to

$$a\sum_{i=0}^{2^{n-k}-1}\prod_{j=0}^{n-1}\lambda_j^{c'_{ij}}, \tag{4.6}$$

whereas expression (4.4) evaluates to

$$\sum_{i=0}^{2^{n-k}-1}a_i\prod_{j=0}^{n-1}\lambda_j^{c'_{ij}},\quad a_i\geq 0. \tag{4.7}$$

Both expressions are sums over all dual codewords c' of the turbo code $\mathcal{C}$ of the same weighted terms. The weights in (4.5) are constant; in (4.4) they are in general not and each is equal to a_i. These factors a_i originate from the difference between the special multiplication $\otimes$ and the real number multiplication. The convergence of the suboptimal decoder is thus related to the asymptotic properties of the factors a_i.

5. Low Density Parity-Check Codes. Equation (2.7) can be expressed as

$$L_m^{\mathcal{C}}=L_m+\log\frac{\prod\limits_{\substack{i=0\\h_{im}=0}}^{n-k-1\otimes}\Big[1+\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]\otimes\prod\limits_{\substack{i=0\\h_{im}=1}}^{n-k-1\otimes}\Big[1+\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}{\prod\limits_{\substack{i=0\\h_{im}=0}}^{n-k-1\otimes}\Big[1+\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]\otimes\prod\limits_{\substack{i=0\\h_{im}=1}}^{n-k-1\otimes}\Big[1-\prod\limits_{j=0}^{n-1}\lambda_j^{h_{ij}}\Big]}. \tag{5.1}$$

Let $\mathcal{C}$ be an LDPC code such that, for all pairs of bits m and l, there is at most one index $i\in\mathcal{I}$ for which $h_{im}=1$ and $h_{il}=1$. In the bipartite,

variable nodes 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. We first illustrate Gallager's belief propagation algorithm in our terminology on a special one-step case, and then consider a general case in terms of the results presented earlier in the paper.

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

$$L_m + \log \frac{\prod\limits_{\substack{i=0\\h_{im}=1}}^{n-k-1}{}_{\otimes}\Big(1+\prod\limits_{\substack{j=0\\j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big)}{\prod\limits_{\substack{i=0\\h_{im}=1}}^{n-k-1}{}_{\otimes}\Big(1-\prod\limits_{\substack{j=0\\j\neq m}}^{n-1}\lambda_j^{h_{ij}}\Big)}. \tag{5.2}$$

Because of the low-density assumption the special multiplication above is equal to the real multiplication. 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)

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

It can be easily shown (in the manner of proof of Theorem 2.1) by using the low-density assumption that

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

Upon receiving this message, the update for bit m is computed by (5.2)

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

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

This brings us a step *closer* to the expression for optimal decoding of bit m given by (5.1).

We now consider a general case. Let $\mathcal{N}(m)$ denote the set of *neighbors* of m, namely

$$\mathcal{N}(m) = \{l | 0 \leq l \leq n-1 \text{ and } \exists i \in \mathcal{I} \text{ such that } h_{im} = 1 \text{ and } h_{il} = 1\}.$$

For our low-density assumption, we shell also require that no two members of $\mathcal{N}(m)$ are neighbors themselves, *i.e.,* there are no loops of length 8.

In connection with decoding of bit m, we look at the set of subcodes of $\mathcal{C}$ whose index sets are defined as follows:

$$\begin{aligned}
&\mathcal{I}_m = \{i \in \mathcal{I} | h_{im} = 1\} \\
&\mathcal{I}_{ml} = \{i \in \mathcal{I} | h_{il} = 1 \text{ and } h_{im} = 0\} \\
&\mathcal{I}_{\mathcal{N}(m),m} = \big(\cup_{l \in \mathcal{N}(m)} \mathcal{I}_{ml}\big) \cup \mathcal{I}_m
\end{aligned}$$

Note that this index sets are disjoint.

THEOREM 5.1. *Let $\mathcal{C}$, λ_m, $\mathcal{I}_m$, $\mathcal{I}_{ml}$, and $\mathcal{I}_{\mathcal{N}(m),m}$ be as previously defined. Then Gallager's belief propagation algorithm [4] gives optimal decoding of bit m over the code $\mathcal{C}_{\mathcal{N}(m),m}$.*

Proof. By Theorem 2.1, the optimal decoder of $\mathcal{C}_{ml}$ outputs the following soft information for bit l:

$$\lambda_l^{\mathcal{C}_{ml}} = \frac{\lambda_l \otimes \prod\limits_{i \in \mathcal{I}_{ml}}{}_{\otimes} \Big[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\Big]}{\prod\limits_{i \in \mathcal{I}_{ml}}{}_{\otimes} \Big[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\Big]}.$$

Gallager's algorithm replaces the special multiplication product $\prod_{\otimes}$ by the real product but, because of the low-density assumption (no loops of length 4), the result is the same.

By Theorem 3.1, the optimal soft output for $\mathcal{C}_{\mathcal{N}(m),m}$ can be computed

by the optimal message passing (3.2) as follows:

$$\lambda_m^{\mathcal{C}_{\mathcal{N}(m),m}} = [\lambda_m^{\mathcal{C}_m}]_2 = \frac{\frac{\lambda_m \otimes A_{\mathcal{C}_{ml}}}{A_{\mathcal{C}_{ml}}} \otimes \prod_{i\in\mathcal{I}_m \otimes} \Big[1 + \prod_{l\in\mathcal{N}(m) \otimes} \Big(\frac{\lambda_j \otimes A_{\mathcal{C}_{ml}}}{A_{\mathcal{C}_{ml}}}\Big)^{h_{ij}}\Big]}{\prod_{i\in\mathcal{I}_m \otimes} \Big[1 + \prod_{l\in\mathcal{N}(m) \otimes} \Big(\frac{\lambda_j \otimes A_{\mathcal{C}_{ml}}}{A_{\mathcal{C}_{ml}}}\Big)^{h_{ij}}\Big]},$$

Gallager's algorithm replaces both special multiplication products $\prod_{\otimes}$ by the real products but, because of the low-density assumption (no loops of length 4 for the *outer* product and no loops of length 8 for the *inner* product), the result is the same. □

Note that by the above proof, we showed that the LDPC codes are a special case of turbo codes for which the participating subcodes are also optimally decoded. Only the optimal decoding of subcodes of an LDPC code is performed by message passing (and can be because of the low-density assumption), rather than by a straight-forward optimal algorithm such as the one in [3].

6. Threshold Decoding. Equation (2.7) can be expressed as

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

We follow the principle of threshold decoding introduced in [5]. Thus, for bit position m, the subcode over which we decode is determined by a subset of parity checks which have a 1 at position m and are *orthogonal* to each other masking out position m. With such parity checks, the special multiplication $\otimes$ in the above equation is the same as the real number multiplication, and we get for the sub-optimal threshold decoding algorithm

of [5]:

$$L_m^{\mathcal{C}} = \log \frac{1+\lambda_m}{1-\lambda_m} + \log \frac{\prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1} \Big[1 + \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}{\prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1} \Big[1 - \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}$$

$$= L_m + \sum_{\substack{i=0 \\ h_{im}=1}}^{n-k-1} \log \frac{\Big[1 + \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}{\Big[1 - \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\Big]}.$$

APPENDIX

A. Turbo Decoding of an $(8,4)$ Single-Loop Code. We consider a code $\mathcal{C}$ whose Tanner graph and the parity check matrix H are shown in Fig. 2. There are four parity check equations of code $\mathcal{C}$: h_0 and h_1 corre-

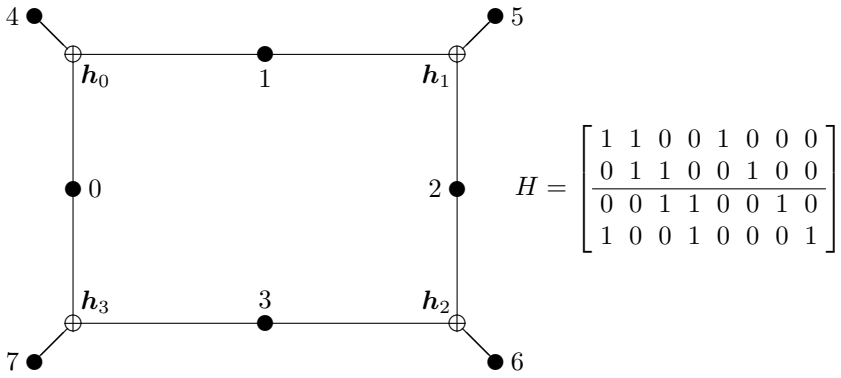


FIG. 2. *An* $(8,4)$ *single-loop code.*

sponding to code $\mathcal{C}_1$ and h_2 and h_3 corresponding to code $\mathcal{C}_2$, as indicated by the dividing line of H in the figure.

The optimal decoder of $\mathcal{C}$ computes

$$\lambda_m^{\mathcal{C}} = \frac{\lambda_m \otimes A_{\mathcal{C}}}{A_{\mathcal{C}}}, \tag{A.1}$$

where

$$\begin{aligned}A_{\mathcal{C}} =&(1+\lambda_0\lambda_1\lambda_4)\otimes(1+\lambda_1\lambda_2\lambda_5)\otimes(1+\lambda_2\lambda_3\lambda_6)\otimes(1+\lambda_0\lambda_3\lambda_7).\\ =&1+\lambda_0\lambda_1\lambda_4+\lambda_1\lambda_2\lambda_5+\lambda_0\lambda_2\lambda_4\lambda_5+\\ &\lambda_2\lambda_3\lambda_6+\lambda_0\lambda_1\lambda_2\lambda_3\lambda_4\lambda_6+\lambda_1\lambda_3\lambda_5\lambda_6+\lambda_0\lambda_3\lambda_4\lambda_5\lambda_6+\lambda_0\lambda_3\lambda_7+\\ &\lambda_0\lambda_1\lambda_2\lambda_3\lambda_5\lambda_7+\lambda_2\lambda_3\lambda_4\lambda_5\lambda_7+\lambda_0\lambda_1\lambda_5\lambda_6\lambda_7+\lambda_1\lambda_3\lambda_4\lambda_7+\\ &\lambda_1\lambda_2\lambda_4\lambda_6\lambda_7+\lambda_4\lambda_5\lambda_6\lambda_7+\lambda_0\lambda_2\lambda_6\lambda_7.\end{aligned}$$

The turbo decoding process starts with decoding of $\mathcal{C}_1$ by calculating the messages to be passed to the decoder of $\mathcal{C}_2$:

$$[\lambda_0^{\mathcal{C}_1}]_1=\frac{\lambda_0\otimes A_{\mathcal{C}_1}}{A_{\mathcal{C}_1}}\quad\text{and}\quad[\lambda_2^{\mathcal{C}_1}]_1=\frac{\lambda_2\otimes A_{\mathcal{C}_1}}{A_{\mathcal{C}_1}}.$$

Note that these are the only bit positions also checked by $\mathcal{C}_2$.

In the second iteration, the decoder of $\mathcal{C}_2$ is active. It uses the information received from the first component decoder, and calculates the messages to pass back to it in the next iteration:

$$[\lambda_0^{\mathcal{C}_2}]_2=\frac{\lambda_0\otimes(1+\lambda_0\lambda_3\lambda_7)\otimes(1+[\lambda_2^{C_1}]_1\lambda_3\lambda_6)}{(1+\lambda_0\lambda_3\lambda_7)\otimes(1+[\lambda_2^{C_1}]_1\lambda_3\lambda_6)}$$

and

$$[\lambda_2^{\mathcal{C}_2}]_2=\frac{\lambda_2\otimes(1+[\lambda_0^{C_1}]_1\lambda_3\lambda_7)\otimes(1+\lambda_2\lambda_3\lambda_6)}{(1+[\lambda_0^{C_1}]_1\lambda_3\lambda_7)\otimes(1+\lambda_2\lambda_3\lambda_6)}.$$

For the soft-output of the whole decoding system after the second iteration, we combine the results from both sub decoders:

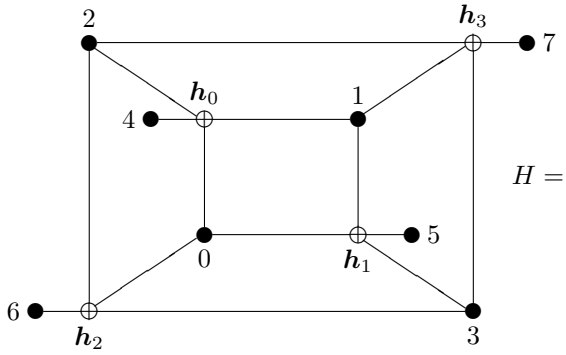
$$[\lambda_0^{C}]_2=\frac{[\lambda_0^{C_1}]_1\otimes(1+[\lambda_0^{C_1}]_1\lambda_3\lambda_7)\otimes(1+[\lambda_2^{C_1}]_1\lambda_3\lambda_6)}{(1+[\lambda_0^{C_1}]_1\lambda_3\lambda_7)\otimes(1+[\lambda_2^{C_1}]_1\lambda_3\lambda_6)}\tag{A.2}$$

and

$$[\lambda_2^{\mathcal{C}}]_2=\frac{[\lambda_2^{\mathcal{C}_1}]_1\otimes(1+[\lambda_0^{\mathcal{C}_1}]_1\lambda_3\lambda_7)\otimes(1+[\lambda_2^{\mathcal{C}_1}]_1\lambda_3\lambda_6)}{(1+[\lambda_0^{\mathcal{C}_1}]_1\lambda_3\lambda_7)\otimes(1+[\lambda_2^{\mathcal{C}_1}]_1\lambda_3\lambda_6)}$$

For bit position '0', one should compare the soft output $\lambda_0^{\mathcal{C}}$ of the optimal decoder given by (A.1) with the soft output of the turbo decoder $[\lambda_0^{C}]_2$ given by (A.2).

B. Turbo Decoding of the $(8,4)$ **Extended Hamming Code.** We consider a code whose Tanner graph and the parity check matrix are shown in Fig. 3.



$$H = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

FIG. 3. *The* $(8, 4)$ *extended Hamming code.*

REFERENCES

[1] 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.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo-codes(1)," *Proc. 1993 IEEE Int. Conf. Commun. (ICC'93)*, Geneva, Switzerland, May 1993, pp .1064–1070.

[3] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, Vol. IT-42, pp .429–445, March 1996.

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

[5] J. Massey, "Threshold Decoding," Cambridge: The M.I.T. Press, 1963.