ISSN 2047-3338

An Efficient Decoding of Low Density Parity Check Codes Based on Variable Node Layering

LAOUINI Nassib[1], BEN HADJ SLAMA Larbi[2] and BOUALLEGUE Ammar[3]

***Abstract*—Layered decoding is known to provide efficient and high-throughput implementation of LDPC decoders. The Check-Node Layered Belief Propagation (CL-BP) algorithm is a modification of Belief Propagation algorithm (BP), where the check nodes are divided in subgroups called layers and each iteration is broken into multiple sub-iterations. Some simplifications can also be made to lower the complexity of both BP and CL-BP algorithms, and particularly the complexity of the check node update rule. In this paper, we consider The Check-Node Layered Belief Propagation (CL-BP) decoding and propose efficient Variable-Node Layering (VL-BP) for updating extrinsic information based on corrective terms. Simulation results show that good performance can be achieved, and which can even be improved by the addition of a normalization term or an offset adjustment term.**

***Index Terms*— Layered decoding, CL-BP, VL-BP, Min-Sum and VL-BP**

I. Introduction

LOW Density Parity Check (LDPC) codes, first introduced by R. Gallager [1] in the early 1960s, deliver very good performance when decoded with the belief-propagation (BP) [15] or the sum-product algorithm [2-4]. As LDPC codes are being considered for use in a wide range of applications, the search for efficient implementations of decoding algorithms is being pursued intensively.

The BP algorithm can be simplified using the so-called Min-Sum (BP-based) approximation [10]. But this simplification is made at the expense of a substantial loss in performance. In [11], an improvement is made to the Min-Sum (BP-based) algorithm by using a correction factor in the check node update rule. It is denoted by offset BP-based algorithm when the correction factor is subtracted to the minimum value, or normalized BP-based algorithm when it is multiplied by the correcting factor.

Recently, several papers have investigated different types of scheduling strategies in BP LDPC decoding. With sequential scheduling, the messages are generated sequentially using the latest available information. Sequential scheduling was introduced as a sequence of check-node updates in [5, 6] and a sequence of variable-node updates in [7, 8]. It is also presented in [20] under the name of Layered BP (LBP), in [9] and [12] as serial schedule, in [13] as row message passing and column message passing.

Check nodes Layered BP algorithm CL-BP is a modification of BP algorithm that divides check nodes into small subgroups called layers and breaks each iteration into multiple sub-iterations. In each sub-iteration one layer of check nodes and their neighboring variable nodes are processed [20].

In [23], an improvement is made to the Check nodes Layered BP (Cl-BP) algorithm by using an efficient variable node layering strategy that significantly increases decoding convergence of LDPC codes as compared to CL-BP. Results show that the decoding convergence of the proposed variable nodes layering outperforms CL-BP and BP decoding.

In this paper, we consider Variable Node Layered BP (VL-BP) algorithm and propose the VL-BP algorithm for LDPC code which has better performance not only from BP algorithm but also from CL-BP algorithm.

The rest of the paper is organized as follows. Section II introduces the LDPC representation with bi-partite graph and describe the principal of LDPC optimal decoding algorithm. Section III presents the Check-Node Layered BP (CL-BP) algorithm. In section IV, a new decoding strategy for updating extrinsic information is proposed based on variable node layering. Sign-magnitude expression of the check node update rule is presented in section V. The section VI focuses on simplifying the check-node update rule to obtain reduced-complexity Variable-Node Layered BP (VL-BP) derivatives that achieve near-optimum performance. The simulation results are discussed in section VII, and finally, conclusions are drawn in section VIII.

II. LDPC Codes and Optimal Decoding

A. LDPC Codes

A binary (N, j, k) LDPC code is a linear block code of length N having a small fixed number 'j' of ones in each column of the parity check matrix H, and a small fixed

[1]SYSCOM Laboratory, ENIT, Tunisia, (Email: nassib_laouini@yahoo.fr)
[2]INNOVCOM Laboratory, SUPCOM, Tunisia, (Email: larbi.bhslama@supcom.rnu.tn)
[3]SYSCOM Laboratory, ENIT, Tunisia, (Email: ammar.bouallegue@enit.rnu.tn)

number 'k' of ones in each rows of H. A sparse $M \times N$ parity-check matrix H can be viewed as a Tanner graph.A Tanner graph is a bipartite graph where the elements of a first class can be connected to the elements of a second class. In a Tanner graph of an LDPC code, elements of the first class are N variable nodes denoted by v_n corresponding to the encoded symbols and the elements of the second class are M parity-check nodes denoted by c_m corresponding to the parity checks represented by the rows of the matrix H. A variable node v_n is connected to a check node c_m if and only if $H(m,n)$ has a non-zero entry. The Tanner graph representation of LDPC codes is very useful since their decoding algorithms can be explained by the exchange of information along the edges of these graphs. The notations related to the Tanner graph and an important hypothesis will be hereafter detailed.

We take the same notation as it was done by Fossorier in [15], let $\mathrm{M(n)}$ denotes the set of check nodes connected to symbol node v_n (i.e. the positions of ones in the nth column of the parity-check matrix H) and let $\mathrm{N(m)}$ denotes the set of symbol nodes that participate in the mth parity-check equation (i.e. the positions of ones in the mth row of H). Furthermore, $\mathrm{N(m)} \backslash n$ represents the set $\mathrm{N(m)}$ excluding the nth symbol node and similarly, $\mathrm{M(n)}$ represents the set $\mathrm{M(n)}$ excluding the mth check node.

Let also $\Phi_{n,k}$ the kth parity check constraint of $\mathrm{M(n)}$ with bit v_n excluded, $k \in \{1,\ldots,|\mathrm{M(n)}|\}$.

To calculate the decoding algorithms complexity, we can define $|\mathrm{M(n)}|$ and $|\mathrm{N(m)}|$ as follows:

- $|\mathrm{M(n)}|$ is the number of parity-check equation by bit.
- $|\mathrm{N(m)}|$ is the weight of the parity-check equation, i.e. the number of terms implied in the parity-check equation.

In order to have independant equations, we consider the *cycle free* hypothesis. A graph is *cycle free* if it contains no path which begins and ends at the same check node without going backward.

B. *Optimal Decoding*

The aim is to find the codeword $\hat{v} = (\hat{v}_1, \ldots \hat{v}_N)$ which is the most probable to have been sent over the channel, based on the received word $y = (y_1, \ldots, y_N)$, and on the knowledge of the code [16]. Using Bayes rule, the posterior probabilities for binary block codes are expressed by these formulas:

$$P(v_n = 0 \backslash y) = \frac{P(y \backslash v_n = 0) P(v_n = 0)}{P(y)}$$

$$P(v_n = 1 \backslash y) = \frac{P(y \backslash v_n = 1) P(v_n = 1)}{P(y)}$$

The decision on binary symbols is defined as follows:

$$\hat{v}_n = \begin{cases} 0 & if P(v_n = 0 \backslash y) > P(v_n = 1 \backslash y) \\ 1 & else \end{cases}$$

The received word $y = (y_1, \ldots, y_N)$ can be split into two sets : y_n and $y_{n' \neq n}$ [17]. Under the hypothesis of a free Inter Symbol Interference channel, y_n depends only on v_n, and is independent of $y_{n' \neq n}$. So the posterior probabilities are expressed by the following equation :

$$P(v_n \backslash y) = P(v_n \backslash y_n, y_{n' \neq n}) = P(y_n \backslash v_n) \times \frac{P(v_n \backslash y_{n' \neq n})}{P(y_n \backslash y_{n' \neq n})} \qquad (4)$$

Using equations (1) and (2), the estimated symbol can be defined as follows :

$$\hat{v}_n = 0 \Rightarrow \frac{P(\hat{v}_n = 0 \backslash y)}{P(\hat{v}_n = 1 \backslash y)} > 1 \Rightarrow log(\frac{P(\hat{v}_n = 0 \backslash y)}{P(\hat{v}_n = 1 \backslash y)}) > 0 \quad (5)$$

$$\hat{v}_n = 1 \Rightarrow \frac{P(\hat{v}_n = 0 \backslash y)}{P(\hat{v}_n = 1 \backslash y)} < 1 \Rightarrow log(\frac{P(\hat{v}_n = 0 \backslash y)}{P(\hat{v}_n = 1 \backslash y)}) < 0 \quad (6)$$

When we use the log-likelihood ratio T_n (LLR) of v_n, defined by :

$$T_n = log(\frac{P(v_n = 0 \backslash y)}{P(v_n = 1 \backslash y)}) \qquad (7)$$

For each received bit v_n; $n = 1, 2, \ldots, N$, in an N-bit block, a decoder uses its log-likelihood ratio T_n which can be expressed by:

$$T_n = I_n + E_n \qquad (8)$$

- T_n is the overall information of the bit v_n.
- $I_n = \log(\frac{P(y_n \backslash v_n = 0)}{P(y_n \backslash v_n = 1)})$ is the intrinsic information. It is related to the received value y_n and to the channel parameters.

- $E_n = \log \dfrac{P(v_n = 0 \setminus y_{n' \neq n})}{P(v_n = 1 \setminus y_{n' \neq n})}$ is the v_n extrinsic

information. It is the information improvement gained by considering the fact that the coded symbols respect the parity check constraints.

$$P(v_n = 1 \setminus y_{n' \neq n}) = P(\phi_{n,1} = 1, \ldots, \phi_{n,|\mathrm{M}(n)|} = 1 \setminus y_{n' \neq n}) \quad (9)$$

Under the assumption of cycle free hypothesis, parity check constraints equations $\phi_{n,k}$ are in disjointed trees so the events $\phi_{n,k} = 1$ for $k \in \{1, \ldots, |\mathrm{M}(n)|\}$ are conditionally independent given $y_{n' \neq n}$ [17]. As seen on Appendix A, the extrinsic information of bit v_n yield:

$$E_n = \sum_{k=1}^{|\mathrm{M}(n)|} E_{n,k} \quad (10)$$

So the extrinsic information E_n is the information given by each of the parity-check constraints $\in \mathrm{M}(n)$ on the bit v_n. Let $v_{n,k,l}$ be the first bit implied in the parity check equation $\Phi_{n,k}$ of degree $|\Phi_{n,k}|$. Then, applying equation (8) to the parity check $\Phi_{n,k}$ yields to:

$$E_{n,k} = 2 \tanh^{-1} \prod_{l=1}^{|\Phi n,k|} \tanh\left(\frac{1}{2} \ln \frac{P(v_{n,k,l} = 0 \setminus y_{n' \neq n})}{P(v_{n,k,l} = 1 \setminus y_{n' \neq n})}\right) \quad (11)$$

Hence, the total information of the bit v_n is completely expressed by:

$$T_n = I_n + \sum_{k=1}^{|\mathrm{M}(n)|} E_{n,k} \quad (12)$$

III. CHECK NODE LAYERED BP ALGORITHM (CL-BP)

LDPC decoding is based on iterative algorithms. An iteration of Belief Propagation (BP) algorithm consists of a round of message passing from each variable node to all adjacent check nodes following by another round of message passing from each check node to its adjacent variable nodes [18].

Check Node Layered BP (CL-BP) decoding is a modification of the Belief Propagation (BP) algorithm. It divides the Tanner graph of an LDPC code into smaller subgraphs, called layers, such that each subgraph consists of a set of check nodes and all their neighboring variable nodes. Each check node appears in exactly one layer, while variable nodes can appear in multiple layers. In each sub-iteration the check node and variable node updates are calculated in one layer [21].

The decoding then progresses sequentially through layers by performing message updates sub-iteration by sub-iteration.

A parity check test over the entire codeword is performed at the end of each sub-iteration. Decoding performance is achieved through repeated iterations of tow messages transmitted from nodes to nodes: $T_{(n,m)}$ and $E_{(n,m)}$. $T_{(n,m)}$ denotes the information which is sent by a variable node v_n to its connected check node c_m and $E_{(n,m)}$ denotes the information which is sent by a check node c_m to its connected variable node v_n [18].

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

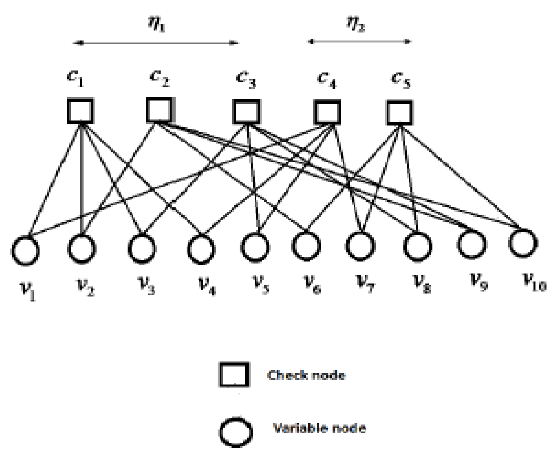

Fig. 1: A parity-check matrix and the corresponding bipartite graph. η_1 and η_2 are check node layers

IV. PROPOSED LAYERING

On this proposed layering strategy, we consider for the first layer a set of variable nodes that has a low value of the intrinsic information I_n of the bit v_n. Each variable node appears in exactly one layer, while check nodes can appear in multiple layers. In each sub-iteration the check node and variable node updates are calculated in one layer. The decoding then progresses sequentially through layers by performing message updates sub-iteration by sub-iteration.

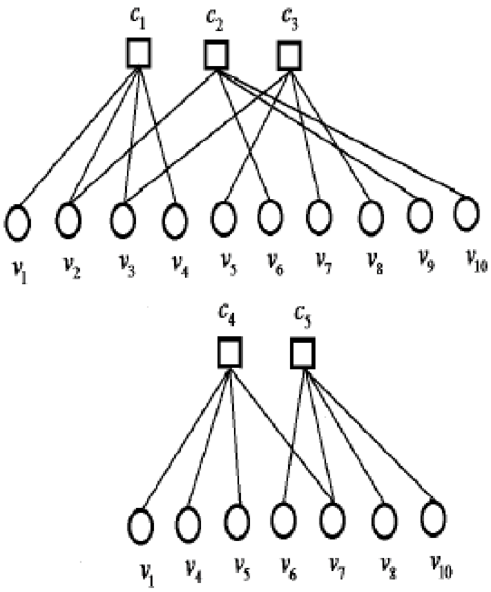

Fig. 2: Two layers (η_1 and η_2) of the Tanner graph in Fig. 1 and their corresponding subgraphs

Let $v = \{v_1, v_2, ..., v_N\}$ denotes the set of all variable nodes and let $c = \{c_1, c_2, ..., c_M\}$ denotes the set of all check nodes. More specifically, for an LDPC code defined by an $(M \times N)$ parity-check matrix, the Variable Nodes Layered BP (VL-BP) algorithm is defined as follows:

• Initialization:

a) For each variable node $v_n \in v$ calculate the intrinsic information I_n.

b) Sort the variable nodes $v_n \in v$ according to the absolute values of the intrinsic information I_n in the ascending order.

c) Group the variable nodes into $K \geq 0$ subgroups S_η for $\eta = 1, \ldots, K$ such that for all $i \neq j$, $S_i \cap S_j = \varnothing$.

d) For the first layer η, consider the subset of variable nodes that has a low values of I_n.

e) All check node messages are initialized to 0 : $E_{(n,m)}^{(0,\eta)} = 0$ for variable node $v_n \in S_\eta$, $\eta = 1, \ldots, K$ and $c_m \in M(n)$. As seen in section 2, $M(n)$ denotes the set of check nodes connected to symbol node v_n.

f) $\eta = 1$ and $l = 1$

• Iterative Processing:

- Variable node update rule: For each variable node $v_n \in S_\eta$ calculate the variable node updated message to its adjacent check nodes $c_m \in M_n$.

$$T_n = T_n + \sum_{m' \in M(n)} E_{n,m'}^{(l,\eta)} - E_{n,m'}^{(l-1,\eta)}$$

- Check node update rule: For each check node $c_m \in M_n$, compute the updated message to its adjacent variable nodes $v_n \in S_\eta$

$$E_{n,m}^{(l,\eta)} = 2\tanh^{-1}\left(\prod_{n' \in N(m)\backslash n} \tanh\left(\frac{T_{n'} - E_{(n',m)}^{(l-1,\eta)}}{2}\right)\right)$$

- Decision rule:

$$\begin{cases} \hat{v}_n = 0 & if T_n > 0 \\ \hat{v}_n = 1 & if T_n < 0 \end{cases}$$

- Generate $\hat{v} = (\hat{v}_1, \ldots, \hat{v}_N)$ and do the following:

* If $H\hat{v}^T = 0$ then the decoding algorithm halts, and $\hat{v}$ is considered as a valid decoding result.

* $\eta = \eta + 1$, if $\eta \leq K$ then repeat the algorithm from variable node update of sub-layer η.

* else $l = l + 1$. A failure is declared if some maximum number of iteration stages occurs without a valid decoding.

* Otherwise, the algorithm repeats from variable node update.

V. Sign-magnitude Check node update rule for VL-BP

As seen in section (4), the check node update rule is expressed by:

$$E_{n,m}^{(l,\eta)} = 2\tanh^{-1}\left(\prod_{n' \in N(m)\backslash n} \tanh\left(\frac{T_{n'} - E_{(n',m)}^{(l-1,\eta)}}{2}\right)\right) \quad (13)$$

This equation can be separated into the sign and the magnitude, as derived hereafter. We have then from (13) :

$$\tanh\frac{E_{n,m'}^{(l,\eta)}}{2} = \prod_{n' \in N(m)\backslash n} \tanh\left(\frac{T_{n'} - E_{n',m}^{(l-1,\eta)}}{2}\right) \quad (14)$$

Replacing $[T_{n'} - E_{n',m}^{(l-1,\eta)}]$ by $[sgn(T_{n'} - E_{n',m}^{(l-1,\eta)}) \times |T_{n'} - E_{n',m}^{(l-1,\eta)}|]$ in (14) yields:

$$sgn(E_{n,m'}^{(l,\eta)}) = \prod_{n' \in N(m)\backslash n} sgn(T_{n'} - E_{n',m}^{(l-1,\eta)}) \quad (15)$$

$$\tanh\frac{|E_{n,m'}^{(l,\eta)}|}{2} = \prod_{n'\in N(m)\backslash n} \tanh\frac{|T_{n'} - E_{n',m}^{(l-1,\eta)}|}{2} \quad (16)$$

Let $f(x)$ be defined by:

$$f(x) = -\ln(\tanh(\frac{x}{2})) = \ln\frac{e^x+1}{e^x-1} \quad (17)$$

Then, taking the logarithm of the inverse of both side of (16) yields:

$$-\ln(\tanh\frac{|E_{n,m'}^{(l,\eta)}|}{2}) = -\ln(\prod_{n'\in N(m)\backslash n} \tanh\frac{|T_{n'} - E_{n',m}^{(l-1,\eta)}|}{2}) \quad (18)$$

$$f(E_{n,m'}^{(l,\eta)}|) = -\sum_{n'\in N(m)\backslash n} \ln(\tanh\frac{|T_{n'} - E_{n',m}^{(l-1,\eta)}|}{2})$$

$$= \sum_{n'\in N(m)\backslash n} f(|T_{n'} - E_{n',m}^{(l-1,\eta)}|) \quad (19)$$

Using the propriety $f(f(x)) = x$, the magnitude of the extrinsic information can be expressed as follows:

$$|E_{n,m}^{(l,\eta)}| = f(\sum_{n'\in N(m)\backslash n} f(|T_{n'} - E_{n',m}^{(l-1,\eta)}|)) \quad (20)$$

So the check node update rule of VL-BP algorithm can be written with separate sign and magnitude, yielding the following equation:

$$E_{n,m}^{(l,\eta)} = \prod_{n'\in N(m)\backslash n} sign(T_{n'} - E_{n',m}^{(l-1,\eta)})$$
$$\times f(\sum_{n'\in N(m)\backslash n} f(|T_{n'} - E_{n',m}^{(l-1,\eta)}|)) \quad (21)$$

VI. APPROXIMATED GENERAL REPRESENTATIONS OF THE VARIABLE NODE LAYERING BELIEF PROPAGATION ALGORITHM

This section focuses on simplifying the check-node update rules to obtain reduced-complexity VL-BP derivatives that achieve near-optimum performance.

A. *VL-BP Based Decoding*

There is an important simplification for the BP algorithm in the literature [10] : the BP-based algorithm. The same approximation can also be made for the proposed VL-BP algorithm since the check node update is replaced by a selection of the minimum input value. The check node update rule of VL-BP algorithm can be expressed by the following equation :

$$E_{n,m}^{(l,\eta)} = \prod_{n'\in N(m)\backslash n} sign(T_{n'} - E_{n',m}^{(l-1,\eta)})$$
$$\times min_{n'\in N(m)\backslash n} |T_{n'} - E_{n',m}^{(l-1,\eta)}| \quad (22)$$

B. *Offset VL-BP Based Decoding*

We propose, similarly to [11], another approximation for the VL-BP Based algorithm by using a correction factor in the check node update rule. We denote by offset VL-BP-based algorithm when the correction factor is subtracted to the minimum value. The check node update rule for the offset VL-BP-based algorithm is expressed by the following equation:

$$E_{n,m}^{(l,\eta)} \leftarrow sgn(E_{n,m}^{(l,\eta)}).max(|E_{(n,m)}^{(l,\eta)}| - \beta, 0), \beta > 0 \quad (23)$$

C. *Normalized VL-BP Based Decoding*

On this section, we propose a modified VL-BP based algorithm based on normalization term. We denote by normalized VL-BP-based algorithm when the check node update is multiplied by the correcting factor. Equation (22) yields to:

$$E_{n,m}^{(l,\eta)} \leftarrow \frac{E_{n,m}^{(l,\eta)}}{\alpha}, \alpha > 1 \quad (24)$$

VII. SIMULATION RESULTS

A. *Simulation conditions*

In this section, we compare Bit Error Rate (BER) performance of BP algorithm, Check-Node Layered BP (CL-BP) algorithm, and Variable-Node Layering and their reduced-complexity derivatives. For all the simulations, there is no encoding program so the decoder generates noisy samples from the all-zero codeword. The received bit are decided by comparing the total information to zero.

Let N_0 denotes the one sided noise power spectrum density and E_b denotes the average energy per information bit.

All the simulations are specified with:

- The maximum number of iterations $iter_{max}$. The syndrome is computed at each iteration. If the syndrome is equal to zero, the iterations are stopped.

- The maximum number of errors (bit or word) to be reached before increasing the $\frac{E_b}{N_0}_{dB}$ value.

- The maximum number of words that are to be generated for each $\frac{E_b}{N_0}_{dB}$.

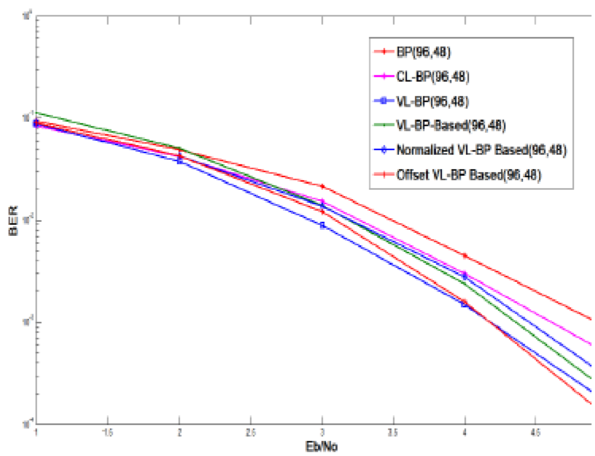

Fig. 3: Comparison between BP, CL-BP VL-BP algorithms and their derivatives (VL-BP based , offset VL-BP based, Normalized VL-BP based) for LDPC code C_1 as a function of the $\frac{E_b}{N_0}$ for itermax=2. The BER obtained with the CL-BP algorithm and VL-BP algorithm is computed by considering tow layers of check nodes and variable nodes, respectively.

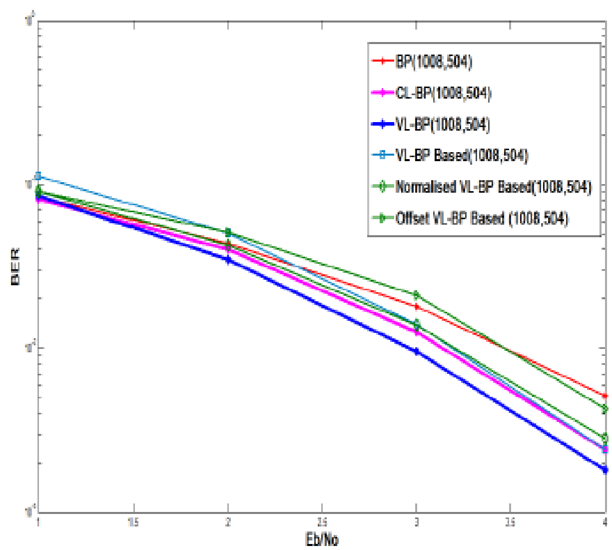

Fig.4: Comparison between BP, CL-BP VL-BP algorithms and their derivatives (VL-BP based , offset VL-BP based, Normalized VL-BP based) for LDPC code C_2 as a function of the $\frac{E_b}{N_0}$ for itermax=2. The BER obtained with the CL-BP algorithm and VL-BP algorithm is computed by considering tow layers of check nodes and variable nodes, respectively.

All the simulations ends when 200 erroneous codewords are detected. A bit is said to be wrong if the intrinsic information I_n is negative, and it is said to be right if it is positive.

B. Codes used for simulations

For all the simulations, we design tow LDPC codes of rate 0.5 taken from the MacKays online database.
The code C_1 is a regular $(5,10)$-LDPC code of length $N = 1008$. The code C_2 is a regular $(3,6)$-LDPC code of length $N = 96$.

C. Codes Algorithm Comparison

The result of the check node update equation, which is over-estimated for the BP-based algorithm, is then closer to the result obtained with the BP algorithm. Some LLR computed with different algorithms on the same channel input are given in table 1, where the input are listed in the ascending order for the code C_2. We can observe that all the approximations of the VL-BP algorithm are over-evaluated. Of course, when increases, the approximation is improved.

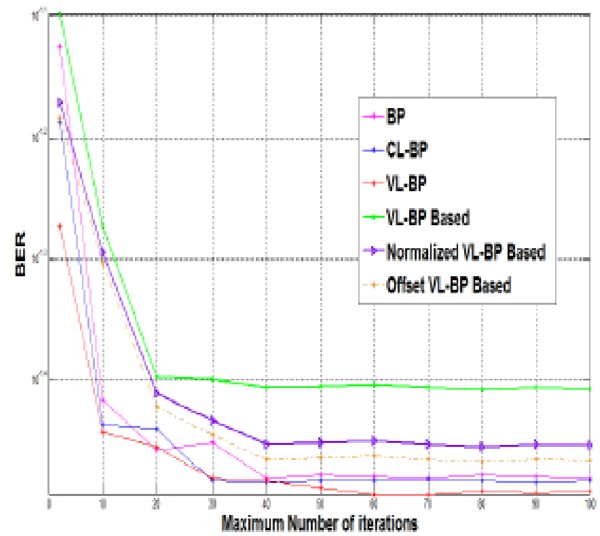

Fig .5: Comparison between BP, CL-BP VL-BP algorithms and their derivatives (VL-BP based , offset VL-BP based, Normalized VL-BP based) for LDPC code C_2 as a function of the number of iterations for a fixed $\frac{E_b}{N_0}$ of 1.5dB.

A comparison between the BP algorithm , the layered BP and the proposed layered BP algorithm for LDPC codes C_1 and C_2 as a function of the $\frac{E_b}{N_0}$ is depicted on figures (1),(2),(3),(4),(5),(6) and (7). Many conclusions can be made for this comparison:

• The performance for all the cases is increasing with the length of code. And the differences between BP, CL-BP VL-BP and their derivatives (VL-BP based , offset VL-BP based, Normalized VL-BP based) is also increasing.
• VL-BP improves the decoding convergence compared to the BP and CL-BP algorithms.

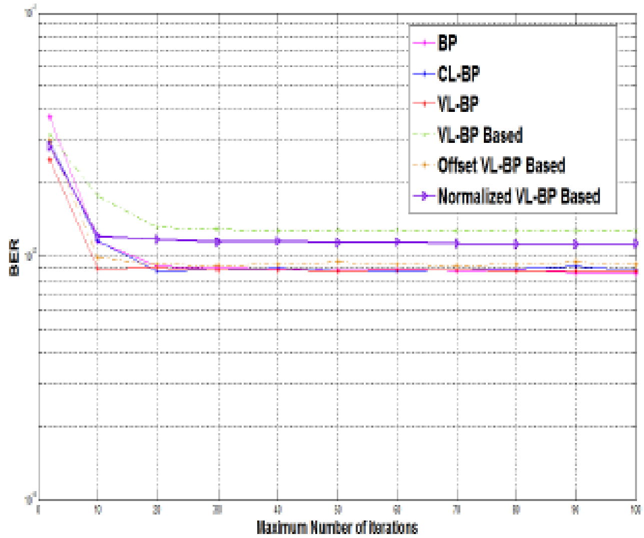

Fig.6: Comparison between BP, CL-BP VL-BP algorithms and their derivatives (VL-BP based , offset VL-BP based, Normalized VL-BP based) for LDPC code C_2 as a function of the number of iterations for a fixed $\frac{E_b}{N_0}$ of 2.5 dB.

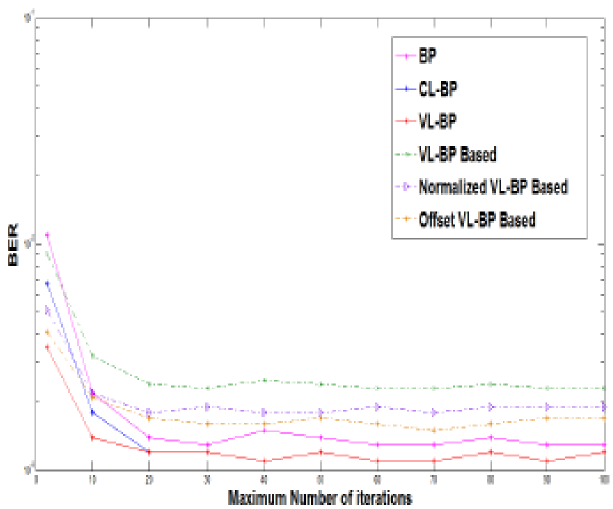

Fig.7: Comparison between BP, CL-BP VL-BP algorithms and their derivatives (VL-BP based , offset VL-BP based, Normalized VL-BP based) for LDPC code C_2 as a function of the number of iterations for a fixed $\frac{E_b}{N_0}$ of 3.5dB.

• the VL-BP based algorithm reduces the complexity of decoding but there is a degradation compared to the VL-BP algorithm.

• The performance of the VL-BP Based algorithm is improved by the addition of a correction factor in the check node update rule.The performance of offset VL-BP Based algorithm is very closed to the performance of the VL-BP algorithm.

We conclude that the variable nodes layering strategy VL-BP based on least a priori information layering can outperform both the BP and the CL-BP algorithms on terms of BER. This is explained by a faster convergence, when the number of iterations increases.

The complexity of the check node update is reduced at the expense of no significant performance loss. Moreover, the addition of an offset or a normalized factor increases the convergence speed of the VL-BP Based algorithm: for a given number of iterations, it can outperform the BP algorithm.

VIII. CONCLUSION

This paper discusses a solution to accelerate convergence of LDPC decoding algorithm. We propose an efficient simplification of Belief propagation algorithm, for updating extrinsic information that finds good variable node layering under the Layered Belief Propagation decoding. can also be made to lower the complexity of the BP algorithm, and particularly the complexity of the check node update rule. A trade-off is then to be decided between the simplifications of the algorithm, and the loss of performance. Simulation results show that good performance can be achieved and improved by the addition of a correction factor.

APPENDIX A: PROOF OF THE EXPRESSION OF THE EXTRINSIC INFORMATION

Assuming cycle free hypothesis and combining equations (8), (9) and (10) , the extrinsic information of bit v_n can be expressed as follow:

$$E_n = \ln \frac{P(v_n = 0 \backslash y_{n'\neq n})}{P(v_n = 1 \backslash y_{n'\neq n})} = \ln \frac{\prod_{k=1}^{|M(n)|} P(\Phi_{n,k} = 0 \backslash y_{n'\neq n})}{\prod_{k=1}^{|M(n)|} P(\Phi_{n,k} = 1 \backslash y_{n'\neq n})}$$

$$= \sum_{k=1}^{|M(n)|} \ln \frac{P(\Phi_{n,k} = 0 \backslash y_{n'\neq n})}{P(\Phi_{n,k} = 1 \backslash y_{n'\neq n})} = \sum_{k=1}^{|M(n)|} E_{n,k}$$

REFERENCES

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

[2] David MacKay - Good Error Correcting Codes based on Very Sparse Matrices - IEEE Transactions on Information Theory, Vol. 45, pp 399 à 431, Mars 1999.

[3] Sorokine Vladislav, Kschischang Franck R. et Pasupathy Subbarayan - Gallager Codes for CDMA Applications - Part I : Generalizations Constructions and Performance Bounds - IEEE Transactions on Communications, Vol. 48, n° 10, pp 1660 à 1668, Octobre 2000.

[4] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi et Daniel A. Spielman - Improved low density Parity check codes using irregular graphs and belief propagation - Technical Report TR-97-044, Digital Equipment Corporation System Research Center, Berkeley, CA, 1997.

[5] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, “High throughput low-density parity-check decoder architectures," in Proc. IEEE GLOBECOM, pp. 3019-3024, San Antonio, TX, Nov. 2001.

[6] M. M. Mansour and N. R. Shanbhag, “High-throughput LDPC decoders," IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 11, pp. 976-996, Dec. 2003.

[7] J. Zhang and M. Fossorier, “Shuffled belief propagation decoding," in Proc. 36th Annual Asilomar Conf. Signals, Syst. Comput., pp. 8-15, Nov. 2002.

[8] H. Kfir and I. Kanter, “Parallel versus sequential updating for belief propagation decoding," Physica A, vol. 330, pp. 259-270, 2003.

[9] E. Sharon, S. Litsyn, and J. Goldberger, “An efficient message-passing schedule for LDPC decoding," in Proc. 23rd IEEE Convention Electrical Electronics Engineers Israel, pp. 223-226, Sep. 2004.

[10] Fossorier, M.P.C., M. Mihaljevi c', and I. Imai. May 1999. “Reduced Complexity Iterative Decoding of Low-Density Parity-Check Codes Based on Belief Propagation.” Transactions on Communications 47:673–680.

[11] Chen, J., and M.P.C. Fossorier. May 2002. “Density Evolution for Two Improved BP-Based Decoding Algorithms of LDPC Codes.” IEEE Communication Letters 6:208–210.

[12] E. Sharon, S. Litsyn, and J. Goldberger, “Efficient message-passing schedule for LDPC decoding," IEEE Trans. Inf. Theory, vol. 53, pp. 4076-4091, Nov. 2007.

[13] P. Radosavljevic, A. de Baynast, and J. R. Cavallaro, “Optimized message passing schedules for LDPC decoding," in Proc. 39th Asilomar Conf. Signals, Syst. Comput., pp. 591-595, 2005.

[14] F. R. Kschischang, B. J. frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. IEEE Trans on Information Theory, 47(2) :498-519, Feb 2001.

[15] Jinghu Chen and Marc P.C.Fossorier. Near optimum universal belief propagation based decoding of low-density parity check codes. IEEE Transactions on Communicatons, March 2002.

[16] Frederic Lehmann. Les Systèmes de Décodage Itératif et leurs Applications aux Modems Filaires et Non-filaires. PhD thesis, Laboratoire AST Grenoble Lab dans le cadre de l’Ecole Doctorale Electronique, Electrotechnique, Automatique, élécommunications, Signal, décembre 2002.

[17] Frédéric Guilloud. Generic Architecture for LDPC Codes Decoding. PhD thesis, ENST Paris, July 2004.

[18] Sabine Leveiller. Quelques algorithmes de cryptanalyse du registre filtré. PhD thesis, L’Ecole Nationale Supérieure des Télécommunications, Janvier 2004.

[19] M.P.C.Fossorier, M.Mihaljevic, and I.Imai. Reduced complexity iterative decoding of low-density parity-check codes based on belief propagation. Transactions on Communications, May 1999.

[20] D. E. Hocevar, A reduced complexity decoder architecture via layered decoding of LDPC codes, in Proc. SIPS 2004, pp. 107112.

[21] Jeongseok Ha, Demijan Klinc, Jini Kwon, and Steven W. McLaugh- lin,Layered BP Decoding for Rate-Compatible Punctured LDPC Codes,IEEE COMMUNICATIONS LETTERS, VOL. 11, NO. 5, MAY 2007.

[22] Frederic Guilloud, Emmanuel Boutillon, and Jean-Luc Danger. Décodage des codes ldpc par l’algorithme λ-min. 19eme colloque GRETSI sur le traitement du signal et des images, Septembre 2003.

[23] Laouini Nassib, Larbi Ben Hadj Slama and BOUALLEGUE Ammar, Fast Decoding of Low Density Parity Check Codes, IEEE High Capacity Optical Networks and Enabling Technologies (HONET), 2012 9th International Conference , Page(s):052-056.