Knowledge-Aided Reweighted Belief Propagation Decoding for Regular and Irregular LDPC Codes with Short Blocks

Jingjing Liu
Department of Electronics
The University of York
Heslington, York, YO10 5DD, UK
Email: jl622@ohm.york.ac.uk

Rodrigo C. de Lamare
Department of Electronics
The University of York
Heslington, York, YO10 5DD, UK
Email: rcdl500@ohm.york.ac.uk

***Abstract*—In this paper a new message passing algorithm, which takes advantage of both tree-based re-parameterization and the knowledge of short cycles, is introduced for the purpose of decoding LDPC codes with short block lengths. The proposed algorithm is called variable factor appearance probability belief propagation (VFAP-BP) algorithm and is suitable for wireless communications applications, where both good decoding performance and low-latency are expected. Our simulation results show that the VFAP-BP algorithm outperforms the standard BP algorithm and requires a significantly smaller number of iterations than existing algorithms when decoding both regular and irregular LDPC codes.**

***Index Terms*—LDPC codes, belief propagation, tree-based re-parameterization, message passing, low-latency.**

I. Introduction

Low-density parity-check (LDPC) codes were first introduced by Robert Gallager in his doctoral dissertation [1] and re-discovered by MacKay, Luby, and others in the 1990s [2], [3]. It has been widely recognized that LDPC codes are able to closely approach the channel capacity by using iterative decoding algorithms, which are parallelizable in hardware and have much lower per-iteration complexity than turbo codes [4]. Unlike turbo codes, it is very easy to implement LDPC codes with any block length and flexible code rate due to the convenience of adjusting the size of the parity-check matrix. Moreover, most of decoding errors are detectable since the decoded codeword is validated by a simple set of parity-check equations. Equipped with efficient decoders, LDPC codes have found applications in a number of communication standards, such as DVB-S2, IEEE 802.16 and Wi-Fi 802.11. Nevertheless, the decoding algorithms of LDPC codes normally require a significantly higher number of iterations than that of turbo codes, which results in severe decoding latency [4].

The belief propagation (BP) algorithm is an efficient message passing algorithm which has been employed to solve a variety of inference problems in wireless communications, among which its applications in decoding powerful error-correcting codes are the most noticeable. Various versions of BP-based algorithms [5]-[8] were reported for decoding turbo codes and LDPC codes. All relevant decoding strategies, either mitigating the error floor or improving waterfall behavior, can be classified into two categories: 1) removing the short cycles in the code graph to avoid “near-codeword” or “trapping sets”; 2) enhancing the suboptimal BP decoding algorithm, when using maximum-likelihood (ML) decoding is intractable [9]. However, in wireless communications where a large amount of data transmission and data storage are required, those decoding algorithms fail to guarantee convergence and still suffer from high-latency due to the fact that too many iterations are often required. In [10] and [11], the authors state that the BP algorithm is capable of producing the exact inference solutions when the graphical model is a spinning tree, while it is not guaranteed to converge if the graph possesses cycles which significantly deteriorate the overall performance. Inspired by the tree-reweighted BP (TRW-BP) algorithm [10], Wymeersch and others [15] recently proposed the uniformly reweighted BP (URW-BP) algorithm which takes advantage of BP’s distributed nature and defines the factor appearance probability (FAP) in [10] as a constant value. In [16], the URW-BP has been shown to consistently outperform the standard BP in terms of LDPC decoding among other applications.

In this paper, we explore the re-parameterization of a certain part of a factorized representation of the graphic model while also statistically taking the effect of short cycles into account. By combining the re-parameterization framework with the knowledge about the structure of cycles of a graph obtained by the cycle counting algorithm [12], which has been successfully employed in our previous works on rate-compatible LDPC codes [13], [14], we present the variable FAP BP (VFAP-BP) algorithm which aims to decode regular and irregular LDPC codes with short block lengths more effectively. The main contributions are:

- A knowledge-aided BP algorithm is devised such that the reweighting factors (FAPs) are chosen by a simple criterion. Moreover, the proposed algorithm can be applied to both symmetrical and asymmetrical graphs.
- We conduct a study on the most recent reweighted BP algorithms [15], [16], and compare the proposed VFAP-BP algorithm to the standard BP and URW-BP algorithms, in terms of convergent behavior as well as decoding performance.

The organization of this paper is as follows: Section II introduces the background of decoding LDPC codes using standard BP message passing rules and a tree-based re-parameterization method. In Section III, the proposed VFAP-BP algorithm is presented in detail. Section IV shows the simulation results with analysis and discussions. Finally, Section V concludes the paper.

II. Graphical Representation and BP Message Passing Rules

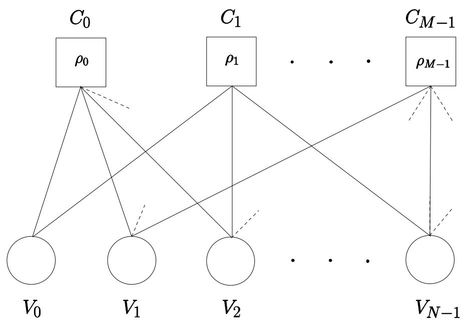

Fig. 1. The graphical model depicts BP decoding algorithms for LDPC codes, where $\rho_i(i = 0, 1, \ldots, M-1) = 1$ corresponds to the standard BP, $\rho_i(i = 0, 1, \ldots, M-1) = \rho_u$ corresponds to the URW-BP, and $\rho_i(i = 0, 1, \ldots, M-1) = \rho_v$ or 1 depending on a variable condition corresponds to the proposed VFAP-BP.

The advantages of LDPC codes arise from its sparse (low-density) parity-check matrices which can be uniquely depicted by graphical representations, referred as Tanner graphs [17]. For instance, an $M \times N$ sparse matrix $\boldsymbol{H}$ can be represented by a bipartite graph G, as in Fig. 1 where $C_0, C_1, \ldots, C_{M-1}$ denote parity check nodes corresponding to M parity check equations and $V_0, V_1, \ldots, V_{N-1}$ denote variable nodes corresponding to N encoded bits. There is an edge connecting the check node C_i and the variable node V_j in the factor graph if the entry h_{ij} of the parity-check matrix $\boldsymbol{H}$ equals 1. Suppose we have K information bits being transmitted and a set of codewords $\boldsymbol{x}$ with block length N is formed under the code rate R is K/N by an LDPC encoder. At the receiver side, the decoder strives to find an $1 \times N$ estimated codeword $\hat{\boldsymbol{x}}$ which satisfies the parity-check condition $\boldsymbol{H}\hat{\boldsymbol{x}}^{\boldsymbol{T}} = \boldsymbol{0}$. Thus, we can interpret the decoding process as finding $\hat{\boldsymbol{x}} = \arg\max p(\boldsymbol{x}|\boldsymbol{y})$. Using Bayes' rule the a posteriori distribution becomes

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}, \tag{1}$$

where the likelihood ratios $p(\boldsymbol{y}|\boldsymbol{x})$ can be obtained from the channel and $p(\boldsymbol{x})$ is the prior information. Nevertheless, directly calculating $p(\boldsymbol{x}|\boldsymbol{y})$ or $p(\boldsymbol{y})$ is computationally prohibitive because of the size of $\boldsymbol{x}$ [11]. For this reason, we resort to BP as a near-optimal message passing algorithm which can approximate either $p(\boldsymbol{x}|\boldsymbol{y})$ or $p(\boldsymbol{y})$ [18].

A. Standard BP Algorithm for Decoding LDPC Codes

The BP algorithm is a powerful algorithm to approximately solve inference problems in decoding LDPC codes. This message passing algorithm computes accurate marginal distributions of variables corresponding to each node of a graphical model, and is exceptionally useful when optimal inference decoding is computationally prohibitive due to the large size of a graph [10]. Additionally, the BP algorithm is capable of producing the exact inference solutions if the graphical model is acyclic (i.e., a tree), but the convergence is no longer guaranteed when the graph possesses short cycles. The BP algorithm for computing $p(x_j|\boldsymbol{y})$ for $(j = 0, 1, \ldots, N-1)$ is a distributed algorithm. As shown in Fig. 1, all the check nodes and the variable nodes work cooperatively and iteratively so as to estimate $p(x_j|\boldsymbol{y})$ for $(j = 0, 1, \ldots, N-1)$ [19]. In addition, for achieving a numerical stability as well as saving storage space the marginal distribution $p(x_j|\boldsymbol{y})$ is replaced by log-likelihood ratios (LLRs) $L(x_j|\boldsymbol{y}) \triangleq \log \frac{p(x_j=1|\boldsymbol{y})}{p(x_j=0|\boldsymbol{y})}$.

Consequently, we derive the following message passing rules of standard BP over an additive white Gaussian noise (AWGN) channel: all variable nodes V_j for $(j = 0, 1, \ldots, N-1)$ are initialized by

$$\Psi_{ji} = L(x_j) = \log \frac{p(y_j|x_j = 1)}{p(y_j|x_j = 0)} = 2\frac{y_j}{\sigma^2}, \tag{2}$$

where σ^2 is the noise variance. Then for all check nodes C_i for $(i = 0, 1, \ldots, M-1)$, we update the message sent from C_i to V_j as:

$$\Lambda_{ij} = 2\tanh^{-1}\Big(\prod_{j' \in \mathcal{N}(i)\backslash j} \tanh\frac{\Psi_{j'i}}{2}\Big), \tag{3}$$

where 'tanh(·)' denotes the hyperbolic tangent function and $\mathcal{N}(i)\backslash j$ denotes the neighboring variable nodes' set of C_i except V_j. Next, we update the message sent from V_j to C_i for all variable nodes V_j by:

$$\Psi_{ji} = L(x_j) + \sum_{i' \in \mathcal{N}(j)\backslash i} \Lambda_{i'j}, \tag{4}$$

where $i' \in \mathcal{N}(j)\backslash i$ is the neighboring set of check nodes of V_j except C_i. Finally, at every variable node V_j we acquire the so-called beliefs

$$b(x_j) = L(x_j) + \sum_{i \in \mathcal{N}(j)} \Lambda_{ij}, \tag{5}$$

which is exactly the approximation of $L(x_j|\boldsymbol{y})$ and

$$\hat{x}_j = \begin{cases} 1, \text{ if and only if } b(x_j) > 0 \\ 0, \text{ if and only if } b(x_j) < 0 \end{cases} \tag{6}$$

While applying the above message passing rules, the variable nodes (check nodes) process the incoming message and send the extrinsic information to their neighboring check nodes (variable nodes) back and forth in an iterative fashion, until all M parity check equations are satified ($\boldsymbol{H}\hat{\boldsymbol{x}}^{\boldsymbol{T}} = \boldsymbol{0}$), or the decoder reaches the maximum number of iterations.

B. Tree-Based Re-parameterization and Bethe's Entropy

When the factor graph contains short cycles, the standard BP algorithm normally requires a larger number of iterations but still fails to converge [2], [18], [10]. To address that problem, Wainwright *et al.* developed the TRW-BP algorithm which improves the convergence of BP by reweighting certain portions of the factorized graphical representation in [10] and [11]. However, TRW-BP algorithm only considers a factorized graph with pairwise interactions, i.e., Markov field, and is not suitable for the distributed inference problem. The URW-BP algorithm [15], [16] extends the pairwise factorizations of TRW-BP to hypergraphs and reduces a series of globally optimized parameters to a simple constant.

Given a factor graph, the Kullback Leibler divergence [20] between the beliefs $b(\boldsymbol{x})$ in (5) and $p(\boldsymbol{x}|\boldsymbol{y})$ is defined as

$$\mathrm{KL}(b||p) = \sum_{\boldsymbol{x}} b(\boldsymbol{x}) \log \frac{b(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{y})} \geq 0. \tag{7}$$

Combining the above equation with (1), we have the inequality

$$\log p(\boldsymbol{y}) \geq \mathcal{H}(b) + \mathcal{F}(b), \tag{8}$$

in which $\mathcal{H}(b)$ is the entropy of the distribution $b(\boldsymbol{x})$ and $\mathcal{F}(\cdot)$ is the factorization function. In addition, (8) is valid with equality if and only if $b(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ (more details can be found in [16]). Since the fixed points of the BP algorithm correspond to the stationary points of Bethe's free energy [18], the entropy term in (8) can be replaced by Bethe's approximation with a constant reweighting factor ρ_u as

$$\mathcal{H}(b|\rho_u) = \sum_{j=1}^{N} \mathcal{H}(b_j) - \sum_{i=1}^{M} \rho_u \mathcal{I}_{\mathcal{N}(i)}(b_{\mathcal{N}(i)}). \tag{9}$$

where $\mathcal{I}_{\mathcal{N}(i)}(b_{\mathcal{N}(i)})$ is the mutual information term. The work in [16] points out that maximizing Bethe's entropy is equivalent to maximizing the entropies of $b(\boldsymbol{x})$ as well as minimizing the dependence among all variables. Thus, a new message passing rule can be obtained by finding stationary points of the Lagrangian in (8). It is also important to note that $\rho_u = 1$ corresponds to the standard BP algorithm.

III. Proposed VFAP-BP Decoding Algorithm

In this section, the proposed VFAP-BP algorithm is presented which selects the reweighting parameters under a simple criterion. As mentioned before, the proposed VFAP-BP algorithm does not require a symmetrical factor graph and, for this reason, it is eligible for LDPC codes with both regular and irregular designs. In the following, we briefly explain the cycle counting algorithm then introduce our message passing rules and the VFAP-BP decoding algorithm flow.

The cycle counting algorithm [12] transforms the problem of counting cycles into that of counting the so-called lollipop walks, and has been used in our previous works on rate-compatible LDPC codes [13], [14]. Given a bipartite graph $G(V, E)$ where V denotes the set of vertices ($V = V_c \bigcup V_s$), E denotes the set of edges, and '$|\cdot|$' represents the cardinality of a set. Define $\boldsymbol{P}_{2k}^{v_c}$ as a $|V_c| \times |V_c|$ matrix in which the $(i,j)th$ element is the number of paths of length $2k$ from $v_{c_i} \in V_c$ to $v_{c_j} \in V_c$. Similarly, define $\boldsymbol{P}_{2k+1}^{v_c}$ as a $|V_c| \times |V_s|$ matrix in which the $(i,j)th$ element is the number of paths of length $2k+1$ from $v_{c_i} \in V_c$ to $v_{s_j} \in V_s$. Let us also define $\boldsymbol{L}_{2k',2k-2k'}^{v_c}$ as a $|V_c| \times |V_c|$ matrix in which the $(i,j)th$ element is the number of $(2k', 2k-2k')$-lollipop walks from $v_{c_i} \in V_c$ to $v_{c_j} \in V_c$. Similarly, define $\boldsymbol{L}_{2k'+1,2k-2k'}^{v_c}$ as a $|V_c| \times |V_s|$ matrix in which the $(i,j)th$ element is the number of $(2k'+1, 2k-2k')$-lollipop walks from $v_{c_i} \in V_c$ to $v_{s_j} \in V_s$. For counting cycles of length $2k$, the above quantities can be computed by

$$\boldsymbol{P}_{2k+1}^{v_c} = \boldsymbol{P}_{2k}^{v_c}\boldsymbol{E} - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i+1,2k-2i)}^{v_c}, \tag{10}$$

$$\boldsymbol{P}_{2k}^{v_c} = \boldsymbol{P}_{2k-1}^{v_c}\boldsymbol{E}^T - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i,2k-2i)}^{v_c}, \tag{11}$$

$$\boldsymbol{P}_{2k+1}^{v_s} = \boldsymbol{P}_{2k-1}^{v_s}\boldsymbol{E}^T - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i+1,2k-2i)}^{v_s}, \tag{12}$$

$$\boldsymbol{P}_{2k}^{v_s} = \boldsymbol{P}_{2k-1}^{v_s}\boldsymbol{E} - \sum_{i=0}^{k-1} \boldsymbol{L}_{(2i,2k-2i)}^{v_s}, \tag{13}$$

$$\boldsymbol{L}_{(0,2k)}^{v_c} = (\boldsymbol{P}_{2k-1}^{v_c}\boldsymbol{E}^T) \circ \boldsymbol{I}, \tag{14}$$

$$\boldsymbol{L}_{(0,2k)}^{v_s} = (\boldsymbol{P}_{2k-1}^{v_s}\boldsymbol{E}) \circ \boldsymbol{I}, \tag{15}$$

where '$\circ$' means element-wise matrix multiplication, $\boldsymbol{E}$ is the edge matrix and $\boldsymbol{I}$ is the identity matrix. The total number of cycles of length $2k$ is

$$N_{2k} = \frac{1}{2k}\mathrm{Tr}\big(\boldsymbol{L}_{(0,2k)}^{v_c}\big) = \frac{1}{2k}\mathrm{Tr}\big(\boldsymbol{L}_{(0,2k)}^{v_s}\big), \tag{16}$$

where '$\mathrm{Tr}(\cdot)$' means the trace of a related matrix. In order to find the girth g and to count cycles of length g, $g+2$ and $g+4$ in a Tanner Graph, (10)-(15) are expanded and updated with each other such that counting short cycles is equivalent to counting the so-called lollipop recursions [12].

In this work, we only focus on the value of g, s_i for $i = 0, 1, \ldots, M-1$ is the number of length-g cycles passing a

TABLE I
The algorithm flow of the VFAP-BP algorithm

Initialization:

1: Run (10)-(15), using (16) to find the girth g and s_i the number of length-g cycles passing the check node C_i ;
2: Determine variable FAPs for each check node: if $s_i < \mu_g$ $\rho_i = 1$, otherwise $\rho_i = \rho_v$ where $\rho_v = 2/\bar{n_D}$;

VFAP-BP decoding:

Step 1: Set I_{max} the maximum number of iterations and initialize $L(\boldsymbol{x}) = 2\frac{\boldsymbol{y}}{\sigma^2}$;
Step 2: Update the message passed from variable node V_j to check node C_i using (17), where $\Lambda_{i'j}$ and Λ_{ij} are $0s$ at the first iteration;
Step 3: Update the message passed from variable node V_j to check node C_i using (3);
Step 4: Update the belief $b(x_j)$ using (18) and decide $\hat{\boldsymbol{x}}$;
Step 5: Decoding stops if $\boldsymbol{H}\hat{\boldsymbol{x}}^T = \boldsymbol{0}$ or I_{max} is reached, otherwise go back to Step 2.

check node C_i, and μ_g is the average number of length-g cycles passing a check node. In a similar way to [11] and [16], the reweighting vector $\rho_i = [\rho_0, \rho_1, \ldots, \rho_{M-1}]$ consists of variable factor appearance probabilities (FAP), which describe the probabilities of any check node appearing in a potential spinning tree. As shown in Fig. 1, every check node C_i is assigned to a FAP value such that the message from each check node is partially reweighted. Note that when $\rho_i(i = 0, 1, \ldots, M-1) = 1$ it is equivalent to the standard BP, and when $\rho_i(i = 0, 1, \ldots, M-1) = \rho_u$ it is equivalent to the URW-BP. The message passing rules of the proposed VFAP-BP algorithm can be described as follows. Firstly, the message sent from V_j to C_i is given by

$$\Psi_{ji} = L(x_j) + \rho_{i'} \sum_{i' \in \mathcal{N}(j) \setminus i} \Lambda_{i'j} - (1 - \rho_i)\Lambda_{ij}. \quad (17)$$

The message sent from C_i to V_j is the same as in (3), and we have the belief $b(x_j)$ with respect to x_j described by

$$b(x_j) = L(x_j) + \rho_i \sum_{i \in \mathcal{N}(j)} \Lambda_{ij}. \quad (18)$$

Using the above message passing rules, the proposed VFAP-BP decoding algorithm is depicted in Table I. Note that $\rho_v = 2/n_D$ at the initialization is an estimate of the optimized FAP value according to [11], where n_D is the average connectivity for N variable nodes. when compared to existing re-parameterization techniques [10], [15], the proposed VFAP-BP algorithm only needs an optimization of short cycles with complexity $(\mathcal{O}(gN))$ instead of a global optimization with complexity $(\mathcal{O}(M^2N))$.

IV. Simulation Results

In this section, we compare the proposed VFAP-BP algorithm with the URW-BP algorithm [15] and the standard BP algorithm while decoding LDPC codes with a short block length. The LDPC codes are designed by the PEG [21] method, having a block length of $500(N = 500)$ and a code rate of $1/2$. Other designs and improvements over the PEG are also possible [22], [23]. The average connectivity of N variable nodes is derived as

$$n_d = \frac{1}{\int_0^1 \lambda(x)dx} = \frac{M}{N\int_0^1 \nu(x)dx}, \quad (19)$$

in which $\lambda(x)$ denotes the degree distribution of variable nodes and $\nu(x)$ denotes the degree distribution of check nodes. For the regular code tested, the degree of variable codes is $4(\lambda(x) = x^3)$, the degree of check nodes is $6(\nu(x) = x^5)$, and the average connectivity $n_{d,reg}$ is 4. For the irregular code, the degree distribution of variable nodes is $\lambda(x) = 0.21 \times x^5 + 0.25 \times x^3 + 0.25 \times x^2 + 0.29 \times x$, the degree distribution of check nodes is $\nu(x) = x^5$, and the average connectivity $n_{d,irreg}$ is 3. By using the counting cycle algorithm we found that there are 964 length-6 cycles in the regular graph while there are 1260 length-8 cycles in the irregular graph. As described in Section III, $\rho_i \in \boldsymbol{\rho} = [\rho_0, \rho_1, \ldots, \rho_{M-1}]$ equals 1 if $s_i < \mu_g$, and equals $2/n_d$ if $s_i > \mu_g$.

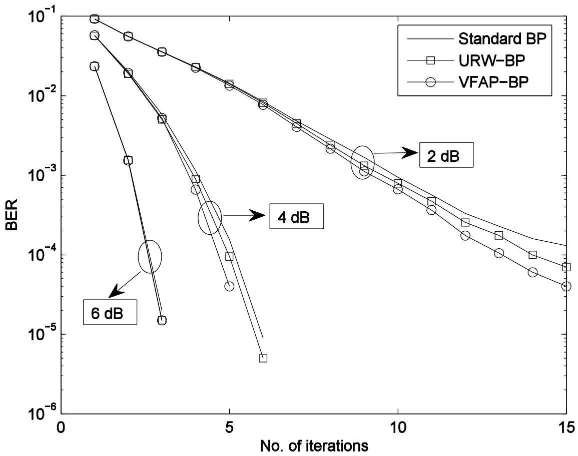

Fig. 2. Comparison of the convergent behaviors of the URW-BP, VFAP-BP and standard BP algorithms for decoding regular LDPC codes, where SNR equal to 2 dB, 4 dB and 6 dB.

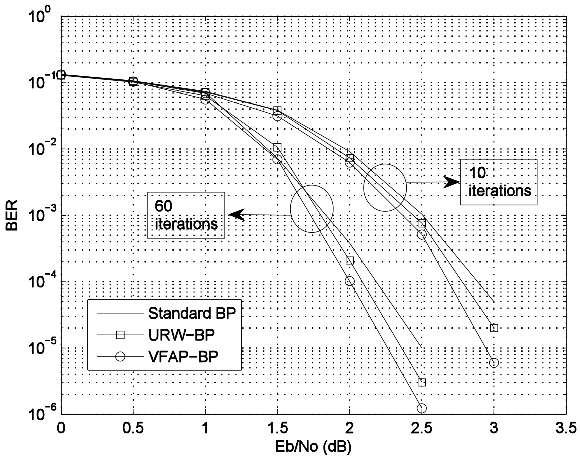

Fig. 3. Comparison of the BER performances of the VFAP-BP, URW-BP and standard BP algorithms while decoding regular LDPC codes with 10 and 60 maximum decoding iterations.

In Fig. 2 the convergent behaviors of the URW-BP, VFAP-BP and standard BP algorithms are compared, in order to illustrate that the proposed algorithm converges faster particularly at lower SNR region. Furthermore, Fig. 3 reveals the decoding performances of three algorithms in which the VFAP-BP outperforms others regardless of the number of maximum decoding iterations. When decoding irregular codes with asymmetrical graphs, as shown in Fig. 4 and in Fig. 5, the proposed VFAP-BP algorithm still shows a better convergence behavior and consistently outperforms the standard BP, but the URW-BP fails to converge at 2 dB as well as no longer outperforms the standard BP with the maximum number iterations equal to 10 and 60, respectively.

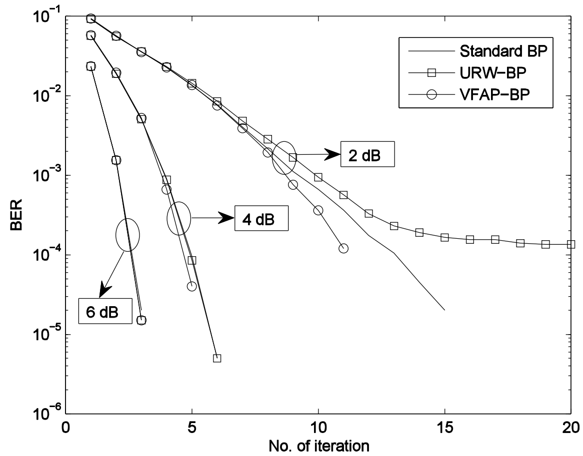

Fig. 4. Comparison of the convergent behaviors of the URW-BP, VFAP-BP and standard BP algorithms for decoding irregular LDPC codes, where SNR equal to 2 dB, 4 dB and 6 dB.

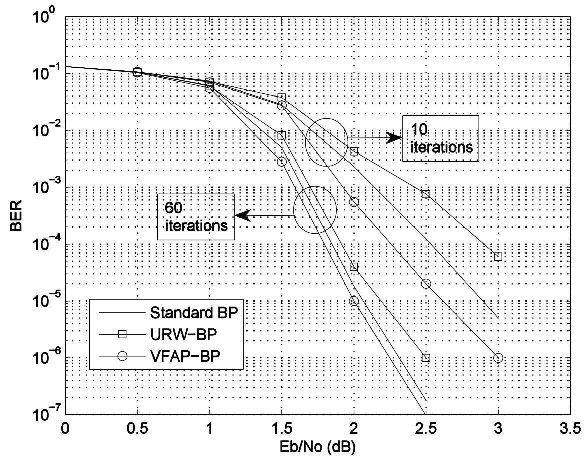

Fig. 5. Comparison of the BER performances of the VFAP-BP, URW-BP and standard BP algorithms while decoding irregular LDPC codes with 10 and 60 maximum decoding iterations.

V. Conclusion

In this paper, we have proposed a message passing decoding algorithm by exploring the tree-based re-parameterization method and the knowledge of the presence of short cycles in the graph structure. The proposed VFAP-BP algorithm has been evaluated when decoding both short-length regular and irregular LDPC codes. Simulation results have shown that the proposed VFAP-BP algorithm is capable of providing good performance while requiring less decoding iterations than the existing algorithms.

Acknowledgment

The authors would like to thank Henk Wymeersch who provided MATLAB codes of URW-BP algorithm for our comparison purpose.

References

[1] R. G. Gallager, "Low-density parity check codes," *IRE Trans. Inf. Theory.*, vol. 39, no. 1, pp. 37-45, Jan. 1962.

[2] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457-458, Mar. 1997.

[3] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *IEEE Trans. Information Theory*, vol. 42, no. 11, pp. 1732-1736, Nov. 1996.

[4] T. K. Moon, "Error Correction Coding: Mathematical Methods and Algorithms," Wiley-Blackwell, July 1, 2005.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proceedings IEEE International Conference on Communications,* Geneva, Switzerland, pp. 1064-1070, May 1993.

[6] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 619637, Feb. 2001.

[7] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498 - 519, Feb. 2001.

[8] H. Xiao and A.H. Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Trans. Communications*, vol. 12, no. 12, pp. 498 - 519, Dec. 2004.

[9] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Mitsubishi Electric Res. Labs, Cambridge, MA, Tech. Rep. TR2001-22, 2002.

[10] M. J. Wainwright, T. S. Jaakkola, and A.S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Trans. Information Theory*, vol. 49, no. 5, pp. 1120 - 1146, May 2003.

[11] M. J. Wainwright, T. S. Jaakkola, and A.S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Trans. Information Theory*, vol. 51, no. 7, pp. 2313 - 2335, July 2005.

[12] T. R. Halford, K. M. Chugg, "An algorithm for counting short cycles in bipartite graph," *IEEE Trans. on Infor. Theory*, vol. 52, no. 1, pp. 287-292, Jan. 2006.

[13] J. Liu and R. C. de Lamare, "Novel Intentional Puncturing Schemes For Finite-Length Irregular LDPC Codes," *International Conference on DSP.*, Corfu, Greece, Jul. 2011.

[14] J. Liu and R. C. de Lamare, "Finite-length rate-compatible LDPC codes based on extension techniques," *8th International Symposium on Wireless Communication Systems*, Aachen, Germany, Nov. 2011.

[15] H. Wymeersch, F. Penna and V. Savic, "Uniformly reweighted belief propagation: A factor graph approach," Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on Issue Date: July 31 2011-Aug. 5 2011 On page(s): 2000 - 2004.

[16] H. Wymeersch, F. Penna and V. Savic, "Uniformly Reweighted Belief Propagation for Estimation and Detection in Wireless Networks," *IEEE Trans. Wireless Communications*, vol. PP, No. 99, pp. 1-9, Feb. 2012.

[17] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Information Theory*, vol. 27, no. 9, pp. 533-547, Sep. 1981.

[18] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Mitsubishi Electric Res. Labs, Cambridge, MA, Tech. Rep. TR2001-22, 2002.

[19] W. Ryan and S. Lin,"Channel Codes: Classical and Modern," Cambridge University Press, 1st edition, Oct. 30, 2009.

[20] T. M. Cover and J. A. Thomas, "Elements of Information Theory," 2nd edition. Wiley-Interscience, 2006.

[21] Y. Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graph," *IEEE Transaction on Information Theory*, vol. 51, no. 1, pp. 386-398, Jan. 2005.

[22] D. Vukobratovic and V. Senk, "Generalized ACE constrained progressive edge-growth LDPC code design, *IEEE Commun. Lett.*, vol. 12, no. 1, pp. 3234, Jan. 2008.

[23] C. T. Healy and R. C. de Lamare, "Decoder-Optimised Progressive Edge Growth Algorithms for the Design of LDPC Codes with Low Error Floors, *IEEE Communications Letters*, 2012, to appear.