IET Research Journals

Submission Template for IET Research Journal Papers

Journals
The Institution of Engineering and Technology

Knowledge-Aided Informed Dynamic Scheduling for LDPC Decoding of Short Blocks

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Cornelius Healy[1], Zhichao Shao[1], Robert M. Oliveira[1], Rodrigo C. de Lamare [1,2] and Luciano L. Mendes[3]*

[1] *Centre for Telecommunications Studies (CETUC), Pontifical Catholic University of Rio de Janeiro, Brazil*
[2] *Department of Electronic Engineering, University of York, UK*
[3] *Inatel, Brazil*
* *E-mail: delamare@cetuc.puc-rio.br*

Abstract: Low-density parity-check (LDPC) codes have excellent performance for a wide range of applications at reasonable complexity. LDPC codes with short blocks avoid the high latency of codes with large block lengths, making them potential candidates for ultra reliable low-latency applications of future wireless standards. In this work, a novel informed dynamic scheduling (IDS) strategy for decoding LDPC codes, denoted reliability-based residual belief propagation (Rel-RBP), is developed by exploiting the reliability of the message and the residuals of the possible updates to choose the messages to be used by the decoding algorithm. A different measure for each iteration of the IDS schemes is also presented, which underlies the high cost of those algorithms in terms of computational complexity and motivates the development of the proposed strategy. Simulations show that Rel-RBP speeds up the decoding at reduced complexity and results in error rate performance gains over prior work.

1 Introduction

Low-density parity-check (LDPC) codes, invented by Gallager [1], have been widely adopted in industry standards in recent years including IEEE 802.11ad (WI-FI) [2], IEEE 802.16e (WiMAX) [3], DVB-S2 [4], IEEE 802.3an (Ethernet) [5] and for enhanced mobile broadband applications of the 5th generation of wireless communications systems. In recent standards, codes with shorter blocks have been proposed, rather than codes with larger blocks as originally advocated for LDPC codes. This is key for the adoption of these codes in future wireless systems and applications such as ultra reliable low-latency (URLL) for machine-type communications [6], as the decoding time required by codes with large blocks might result in unacceptable latency. The structure of the LDPC codes are amenable to low complexity decoding, whereas the computational cost of encoding can be reduced with the help of structured graphs [7–9]. Indeed, cost-effective encoding and decoding of LDPC codes can be an important element that contributed to the energy efficiency of future networks.

The decoding of LDPC codes based on message passing performs very well for large blocks. However, the decoding may experience performance degradation in the presence of cycles found in codes with short blocks. Designs for short blocks [10–12] may be used to mitigate the effects of cycles and improve the performance of LDPC codes by modifying the graph used in the decoding. Another issue with message-passing decoding for LDPC codes in situations with strict requirements on latency and energy consumption [6], is the need for many decoding iterations of the standard belief-propagation type algorithms [1, 13–16] such as the sum-product algorithm (SPA) [1] and the minimum-sum algorithm (Min-sum) [13]. The computational burden of the decoding task can be reduced by introducing approximations to the recursions of SPA as in Min-sum at the cost of error rate degradation. The Min-sum algorithm can be enhanced by introducing correction factors in the check node update approximation, through a multiplicative normalization update factor [15] or an additive offset update factor [14]. In such case, the computational cost per iteration is smaller but the number of required decoding iterations remains the same. In order to accelerate the convergence of decoding algorithms, techniques based on reweighting [17–19] and scheduling [20–24, 27–30] have been studied in the last few years. Reweighting techniques apply scaling factors to the check node update to address the overconfidence introduced by messages exchanged in the presence of cycles. In contrast, scheduling strategies exploit the status of the messages exchanged in the graph to determine the next update of message and which message update brings the highest gain. In particular, dynamic scheduling techniques are suitable for short-block LDPC codes because such codes exhibit a larger number of short cycles and a smaller girth than large-block LDPC codes, which are known to affect the performance of message passing algorithms [12].

In the context of scheduling techniques, the introduction of the residual belief propagation (RBP) [20], the node-wise BP (NW-BP) [24] and the sequential Layered Belief Propagation (LBP) [21] algorithms have motivated a number of studies and further improvements. RBP uses the largest residual obtained from the absolute value of the difference between the messages exchanged between nodes and the message to be updated. LBP employs the most recent message in the graph by performing updates in a serial rather than parallel manner. The informed dynamic scheduling (IDS) schemes [20, 22–24] offer a substantially enhanced performance in terms of convergence speed and error rate, while requiring a significantly higher computational cost due to the extra operations needed to compute residual, many of which are discarded and recomputed when the status of the messages in the graph changes (i.e., when a message is updated, the residuals of the messages affected in the following iteration must be recomputed).

In this work, we present a dynamic schedule, named reliability-based residual belief propagation (Rel-RBP) whose preliminary results were reported in [25, 26]. The proposed Rel-RBP approach exploits the reliability of the messages and the residuals of possible updates to speed up decoding algorithms. We consider a notion of reliability that corresponds to the magnitude of the log-lihelihood

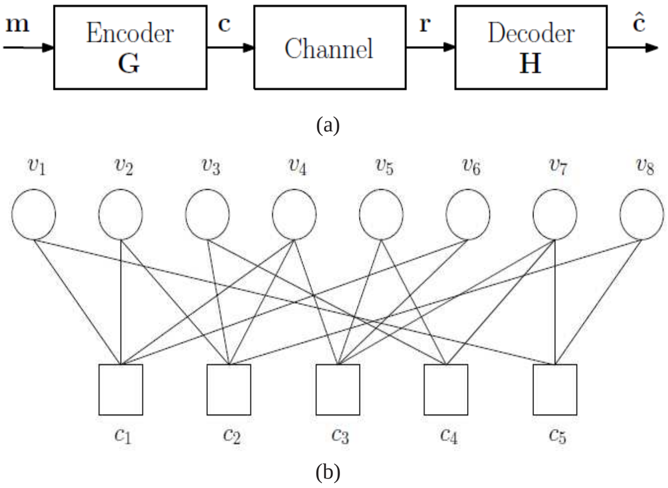


Fig. 1: a) A general LDPC coding system. b) A pictorial description of a Tanner graph.

ratios (LLRs) for log-domain BP algorithms. Using the concept of reliability with possible message updates, a message is then selected for update by computing and comparing its residuals as in the RBP and NW-BP schemes. This approach has two benefits: it limits the use of the residual, thereby reducing the complexity of the algorithm; and prioritizes the update of messages with lower incoming reliability, resulting in accelerated convergence due to the update of information in parts of the graph that did not contributed towards convergence. Rel-RBP has the key feature that message updates associated with a large change in the LLRs (i.e. have a large residual) at nodes with strong beliefs are avoided in order to promote smaller changes at nodes with unreliable beliefs. Simulations indicate that Rel-RBP results in substantial gains in the convergence speed of the decoding algorithm when considering standard decoding iterations. An analysis of the update rules of Rel-RBP shows that the complexity reduction of the proposed approach makes it more practical than existing IDS schemes.

The main contributions of this work can be summarized as:

- A novel knowledge-based message passing approach that exploits the reliability of the messages in the graph for the decoding of LDPC codes.
- A low-complexity message passing algorithm that exploits the reliability of the messages to produce the decoding schedule at a lower cost.
- An alternative approach for measuring the number of effective iterations which provides insight into the computational cost and effectiveness of the algorithms considered.
- Analyses of the computational complexity and fundamental advantages of the proposed and existing algorithms.

The remainder of this paper is structured as follows: Section 2 introduces the notation and the model of the LDPC system. Section 3 reviews the recursions employed by the SPA and the RBP algorithms, introduces the alternative measurement of the iterations performed in the IDS schemes and explains the rationale for the work described in this paper. In Section 4, the proposed Rel-RBP algorithm is detailed. Section 5 presents the simulation study and the discussion of the results. Section 6 concludes the paper.

2 LDPC System Model

A general LDPC coding system is considered in this work, as shown in Fig. 1 a), where a message represented by the $1 \times k$ vector $\mathbf{m}$ is encoded to the length $1 \times n$ code word vector $\mathbf{c}$, subjected to the channel such that the decoder operates on the vector $\mathbf{r}$ to produce an estimate of the code word $\hat{\mathbf{c}}$.

In the decoder, a message passing algorithm works with a bipartite graph that corresponds to the parity-check matrix of the code. Such graph, known as a Tanner graph, is shown in Fig. 1 b), whereas the corresponding parity-check matrix is given by

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \quad (1)$$

The number of rows in the matrix that corresponds to the check nodes in the graph is defined as $n-k$, whereas the number of columns that refers to the variable nodes is defined as n. The average number of entries in a row and column are labeled d_c and d_v, respectively.

3 BP Decoding, RBP Scheduling and Effective Iterations

BP decoding is the most prominent approach for decoding LDPC codes. And the most popular BP decoding techniques are the SPA and the Min-sum. Moreover, scheduling can be used in conjunction with these two techniques to accelerate their convergence. In what follows, we make a review of the SPA with the RBP techniques.

3.1 BP Decoding and RBP Scheduling

In SPA, the messages are exchanged between the variable and the check nodes of the Tanner graph and are updated by the following recursions:

$$\mu^{(k)}_{c_i \to v_j} = 2\tanh^{-1}\Big(\prod_{j' \in \mathcal{N}(c_i)\backslash j} \tanh\big(\frac{\mu^{(k)}_{v_{j'} \to c_i}}{2}\big)\Big), \quad (2)$$

$$\mu^{(k+1)}_{v_j \to c_i} = L_j + \sum_{i' \in \mathcal{N}(v_j)\backslash i} \mu^{(k)}_{c_{i'} \to v_j}, \quad (3)$$

where the message $\mu^{(k)}_{c_i \to v_j}$ is calculated at the check nodes. This message computation is based on the parity constraints of the code and $\mu^{(k+1)}_{v_j \to c_i}$ denotes the messages calculated at the variable nodes and exchanged with the check nodes. The notation $\mathcal{N}(n_a)$ refers to the neighbourhood of node n_a, i.e., the set of nodes connected to n_a by an edge. The quantity $\mathcal{A}\backslash b$ corresponds to the set $\mathcal{A}$ excluding the element b.

The log-likelihood ratio (LLR) for the a posteriori probabilities corresponds to the input to the decoder and is based on the received codeword bit from the channel, L_j , and the estimate for the node v_j is taken from the sign of the final LLR for that node as described by

$$M^{(k+1)}_j = L_j + \sum_{i \in \mathcal{N}(v_j)} \mu^{(k)}_{c_i \to v_j}, \quad (4)$$

In the RBP decoding approach, the message passing algorithm iterates in individual check-to-variable message updates instead of updating all nodes of one or both types, as occurs with the LBP or the standard BP algorithms, respectively. In the binary version of the RBP algorithm, the next message to be updated is selected by the message residuals according to:

$$r(\mu^{(k)}_{c_i \to v_j}) = |\mu^{(k)}_{c_i \to v_j} - \mu^{(k-1)}_{c_i \to v_j}|. \quad (5)$$

The RBP algorithm assigns only the message update with the largest residual and stores all other message updates for possible use in future iterations. With this approach, messages leaving the variable

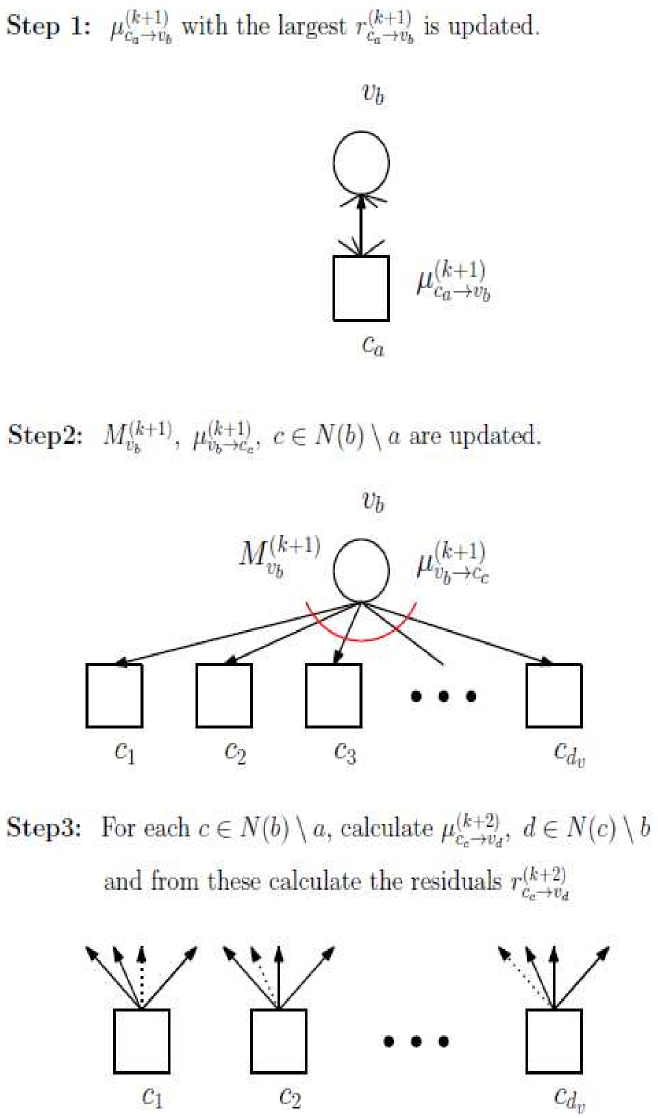


Fig. 2: The main recursions of the RBP strategy.

node v_j that receive the assigned check-to-variable message are updated. The algorithm then computes new residuals at all check nodes that receive those updates. This process is then repeated. The RBP strategy is illustrated in Fig. 2. An extension of the RBP algorithm, known as NS-BP [24], works in a similar way but updates all messages from the check node associated with the largest residual.

3.2 *Effective Decoding Iterations*

In this section, we present the concept of effective decoding iterations, which helps to establish an equivalence between the iterations of IDS schemes and standard BP algorithms that can be useful in their comparison. In our development, we first define an equivalent iteration. The iterations of standard BP algorithms are equivalent to certain points in the processing of IDS decoders. Specifically, we have adopted the approach in [24], termed classic iterations, to provide a reference to the RBP and NS-BP algorithms. Moreover, we evaluate the performance when the number of message updates in the check node assigned by IDS-based algorithms equals that of the standard schemes. This means that all message computations required for the calculation of the residual are considered as a cost of the scheduling strategy examined. However, the extra updates are often indistinguishable from those carried out for the messages exchanged. Furthermore, the message computations must be done with up-to-date information and cannot be computed in previous message exchanges. This suggests the adoption of an alternative effective decoding iteration measure. The modified decoding iteration is defined as the point in the processing of the IDS-based algorithm at which the number of check node message updates computed is equal to that of the standard BP and LBP schemes. With this modified decoding iteration measure a single iteration of each scheme (standard BP and IDS-type) requires the same number of check node updates, resulting in a comparable computational complexity.

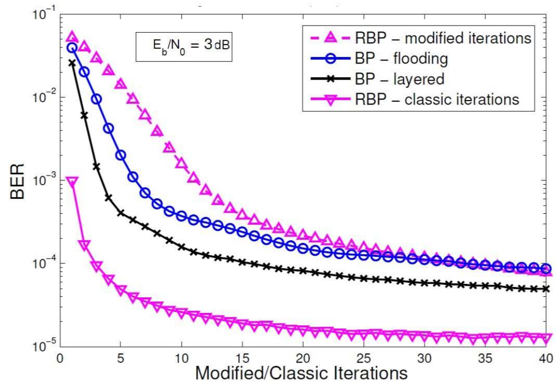


Fig. 3: Modified/classic decoding iterations for a regular LDPC code with $n = 480$, $d_c = 3$ and $d_v = 6$.

Fig. 3 illustrates the impact of this choice on the performance of IDS schemes for the simulated LDPC coding system on the AWGN channel. For an incrementing number of message updates, k, the classic decoding iteration number increments as described by

$$x = \left\lceil \frac{k}{Md_c} \right\rceil, \tag{6}$$

while the modified decoding iteration number increments as given by

$$x = \left\lceil \frac{k(d_c - 1)(d_v - 1)}{Md_c} \right\rceil, \tag{7}$$

where $\lceil a \rceil$ corresponds the smallest integer larger than a.

4 Proposed Knowledge-Aided Informed Dynamic Scheduling

In this section, we describe the proposed knowledge-aided informed dynamic scheduling technique and detail its main characteristics. In order to lower the computational complexity of IDS schemes and to enjoy the excellent decoding speed of those schemes, we have examined the nodes at which message updates result in the highest performance benefit. In addition, we have also considered that it would be beneficial if the IDS algorithm could skip the pre-computation and storage of all messages as required by RBP and NS-BP. By observing the message update behavior of the beginning of the operation, (5) reduces to the following:

$$r(\mu^{(k)}_{c_i \to v_j}) = |\mu^{(k)}_{c_i \to v_j}|. \tag{8}$$

By examining the properties of (2) we notice that this residual is associated with the edge of the graph with the LLR having the smallest

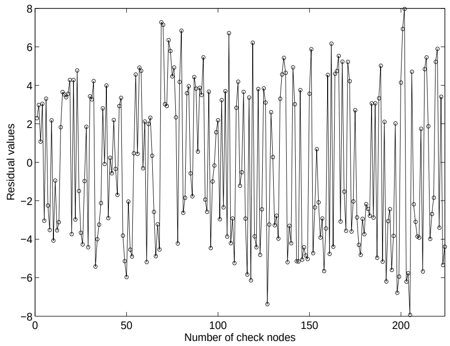


Fig. 4: Residual values for a codeword produced by a PEG design with $n = 480$, $k = 256$ and operating at $E_b/N_0 = 3$ dB.

magnitude, $|\mu^{(k)}_{v_j \to c_i}|$. The magnitude of the LLR is denoted the reliability of the message because it provides information about the confidence level of the message associated with the bit value of the node. We have observed the residuals and the reliability of the incoming messages and found that the largest residual message can be found at the edges with the smallest incoming reliabilities in the vast majority of the occurrences, as illustrated in the new Fig. 4. We have also tested the use of an arbitrary number of residual values (a range from two to ten) but found that using more than two has not resulted in further performance gains. Therefore, we have exploited this observation to reduce the number of computations of residuals required to only two for each message update.

The message updates of the proposed Rel-RBP algorithm are carried out according to the following procedure: for the check node c_m, we identify the two incoming messages associated with the smallest absolute value, i.e., the two incoming messages with the smallest reliabilities as given by

$$\mu^{(k+1)}_{v_{n_1} \to c_m} : |\mu^{(k+1)}_{v_{n_1} \to c_m}| = \min_{n \in \mathcal{N}(c_m)} |\mu^{(k+1)}_{v_n \to c_m}|, \tag{9}$$

and

$$\mu^{(k+1)}_{v_{n_2} \to c_m} : |\mu^{(k+1)}_{v_{n_2} \to c_m}| = \min_{n \in \mathcal{N}(c_m) \backslash n_1} |\mu^{(k+1)}_{v_n \to c_m}|. \tag{10}$$

For the variable nodes v_{n_1} and v_{n_2}, we calculate the variable-node residual described by

$$r^{(k+1)}_{c_m \to v_n} = |\mu^{(k+1)}_{c_m \to v_n} - \mu^{(k)}_{c_m \to v_n}|, n \in \{n_1, n_2\} \tag{11}$$

Then, we compute the check-node residual as given by

$$\mathcal{R}^{(k+1)}_{c_m} = r^{(k+1)}_{c_m \to v_x} : r^{(k+1)}_{c_m \to v_n} = \min_{n \in \{n_1, n_2\}} r^{(k+1)}_{c_m \to v_n}, \tag{12}$$

where the variable node v_x and its related message $\mu^{(k+1)}_{c_m \to v_x}$ are stored.

This procedure is conducted again for each $m = 1, \ldots, k$. Then, we assign the message update that corresponds to the largest check-node-residual, which is selected according to:

$$\mathcal{R}^{(k+1)}_{c_a} = \max_{m \in \{1, \ldots, k\}} \mathcal{R}^{(k+1)}_{c_m}. \tag{13}$$

Every time a check node c_a is identified, the associated message $\mu^{(k+1)}_{c_a \to v_n}$ is assigned, where both n and $\mu^{(k+1)}_{c_a \to v_n}$ are the values stored for c_a that are calculated according to (11) and (12).

The main steps required in the calculation of the residual of the check node are shown in Fig. 5, where Fig. 5a) depicts the identification of the two messages with smallest reliabilities in blue. Fig. 5b) illustrates how the outgoing messages on those two identified edges are computed and Fig. 5c) shows the stored variables for that check node, namely, $\{\mathcal{R}^{(k+1)}_{c_m}, v_{n_2}, \mu^{(k+1)}_{c_m \to v_{n_2}}\}$.

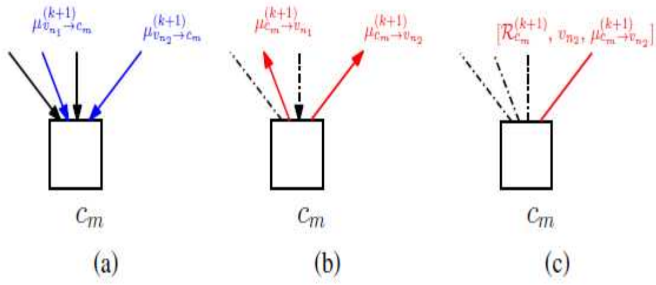


Fig. 5: Main steps for computing the reliability-based residual.

Algorithm 1 Proposed Rel-RBP Algorithm

Initialization
$\mu^{0}_{c_m} \to v_n = 0$, $\mu^{0}_{v_n} \to c_m = L_n$
$\mathbf{i}_b = \mathbf{1}$
For each check node, identify and calculate the maximum residual

$$\mathcal{R}^{(1)}_{c_m} = r^{(1)}_{c_m \to v_a} = \max_n r^{(1)}_{c_m \to v_n}$$

and store the message $\mu_{c_m \to v_a}$
while stoppping rule is not satisfied **do**
    Obtain c_b and compute the following:

$$\mathcal{R}^{(k+1)}_{c_b} = \max_m \mathcal{R}^{(k+1)}_{c_m}$$

    Propagate $\mu_{c_b \to v_c}$, and set the appropriate entries of the vector $\mathbf{i}_b$ to zero.
    Perform a verification of the indicator vector $\mathbf{i}_b$:
    if sum($\boldsymbol{i}_b$) $== 0$ **then**
        $\mathcal{R}^{(k+1)}_{c_b} = 0$
    else if sum($\boldsymbol{i}_b$) $== 1$ **then**
        Apply (9) and (10) with the sets $\mathcal{N}(c_b) \backslash v_q$ and $\mathcal{N}(c_b) \backslash (v_q \cap n_1)$ in the minimization operations, where v_q is the node with nonzero entry of $\boldsymbol{i}_b$.
    else
        Apply (9) and (10) unchanged.
    end if
    For the selected check node, c_b, identify and calculate the next-largest residual $\mathcal{R}^{(k+1)}_{c_b}$ and store the message $\mu_{c_b \to v_p}$
    Update each $\mu_{v_c \to c_d}, c_d \in \mathcal{N}(v_c) \backslash c_b$ and M_{v_c} according to (2) and (3) respectively, and set $\boldsymbol{i}_d$ to one in the appropriate position.
    for each c_d **do**
        Perform verification of the indicator vector $\boldsymbol{i}_b$ as above
        Identify and compute the check node residual

$$\mathcal{R}^{(k+2)}_{c_d}$$

        and save the message $\mu_{c_d \to v_r}, v_r \in \mathcal{N}(c_d) \backslash v_c$, keeping the value r.
    end for
    if the iteration count increments **then**
        Stopping rule: perform parity-checks and stop if all checks are satisfied or if the maximum iteration count has been reached.
    end if
end while

In order to save computations of residuals when there is no new incoming information at a check node, we have adopted a simple binary indicator vector which is represented by $\boldsymbol{i}_m$ at each check node c_m. The entries of $\boldsymbol{i}_b$ correspond to the edges incident on c_m. The binary indicator vector $\boldsymbol{i}_m$ is initialized with ones in all entries. An entry in $\boldsymbol{i}_m$ is set to zero when its corresponding edge has an outgoing message assigned and is set to one when an incoming message update is assigned. Then, prior to the use of (9)-(13), the following verification is performed for each check node. If $\boldsymbol{i}_m$ contains all zeros, then $\mathcal{R}_{c_m}^{(k+1)}$ is set to zero. If $\boldsymbol{i}_m$ contains exactly one nonzero entry, then the recursions (9)-(13) are applied to the set of edges excluding the edge related with the nonzero entry. Otherwise equations (9)-(13) are applied to all edges at c_m.

5 Analysis

In this section we analyze the performance and the complexity of the proposed Rel-RBP and existing decoding algorithms.

5.1 Performance Analysis

The convergence of the SPA algorithm for a cycle free graph is guaranteed. However, for graphs with cycles the convergence is not guaranteed even though message passing decoding algorithms are widely known to perform well in general. The performance of decoders is particularly good for graphs with a reduced number of short cycles and improved connectivity in the graph structure. In this section, we analyze and discuss the procedure by which messages are passed in the RBP and the proposed Rel-RBP algorithms.

Consider an arbitrary check node with incoming messages with edges labeled as a, b, c, d and e. According to the RBP strategy, at a given check node we select the message that has the largest residual. In contrast, with the Rel-RBP algorithm we select the message on the basis of the incoming message reliability as well as the message residuals. In the following analysis, we suppose that the message that arrives at edge a has the lowest reliability and that the message that arrives at edge e has the second lowest reliability. In other words, there are two possible outcomes to be examined: the first one corresponds to the situation in which the edge associated with the largest residual is a or e, whereas the second is the scenario in which the edge with the largest residual could be b, c or d. In the first scenario, RBP and Rel-RBP choose the same update message from this node according to:

$$|\mu_{\mathrm{RBP},1}| = |\mu_{\mathrm{Rel-RBP},1}|. \tag{14}$$

In the second scenario, the message is different and the value will depend on which edge, i.e., a or e, is associated with the largest residual. If the residual with the largest value corresponds to edge e that has the message with the second smallest reliability at the node, then the update of the message will consider the messages that arrive at edges a, b, c and d. Conversely, the update of messages of RBP for the edge with the largest overall residual, i.e., edge b, will involve the messages that arrive at edges a, c, d, e. Since the update of the check nodes employs a recursion with the hyperbolic tangent function, the message that is updated is influenced by the smallest absolute value among the messages that arrive at the edges of interest in this update. In this situation, whose edge is referred as $2a$, the message that arrives at edge a has the largest influence on the update and so the message updated for RBP and Rel-RBP are similar and given by

$$|\mu_{\mathrm{RBP},2a}| \approx |\mu_{\mathrm{Rel-RBP},2a}|. \tag{15}$$

Eventually, when the RBP algorithm selects a message to be updated and such message is not amongst those with the smallest reliabilities then the proposed Rel-RBP algorithm finds that the residual associated with edge a is larger than the residual associated with edge e. In this case, the updated message for RBP will be governed by the incoming messages with the smallest reliability, i.e., the message on edge a. In contrast, the update of Rel-RBP will be dictated by the message with the second smallest reliability, i.e., the message on edge e. In this case, we have

$$|\mu_{\mathrm{RBP},2b}| < |\mu_{\mathrm{Rel-RBP},2b}|. \tag{16}$$

Thus, we find that at each check node we will obtain

$$|\mu_{\mathrm{RBP}}| \leq |\mu_{\mathrm{Rel-RBP}}|, \tag{17}$$

where the equality will only hold for the first scenario previously described. Since the magnitudes of the messages correspond to the level of reliability of the bit at the variable node, the fact that Rel-RBP can generate messages with larger magnitudes that are exchanged in sectors of the graph, which might have LLRs with small magnitudes, can accelerate the convergence of the decoder. This will be corroborated by the numerical simulations in the next section. We also note that the reduced number of iterations required by Rel-RBP is highly suitable and useful for URLL scenarios and hardware implementations that cannot deal with large storage requirements for residuals and messages.

5.2 Complexity Analysis

Amongst the main motivation factors for the development of novel decoding algorithms are the decoding latency of standard (flooding) BP algorithms and the relatively high complexity of the previously reported IDS algorithms to choose in a flexible way the messages to be updated. The computational cost of the decoding techniques considered in this work is detailed in Table 1 as a function of the number of updates I, the number of variable nodes n, the number of check nodes k, the average degree of check nodes d_c, the average degree of variable nodes d_v and the total number of edges in the graph. We note that the figures included in Table I correspond to the computational effort required considering classic iterations and for each scheduling technique, as described in [30]. When we consider the modified decoding iterations, each decoding technique uses the check node update equation the same number of times.

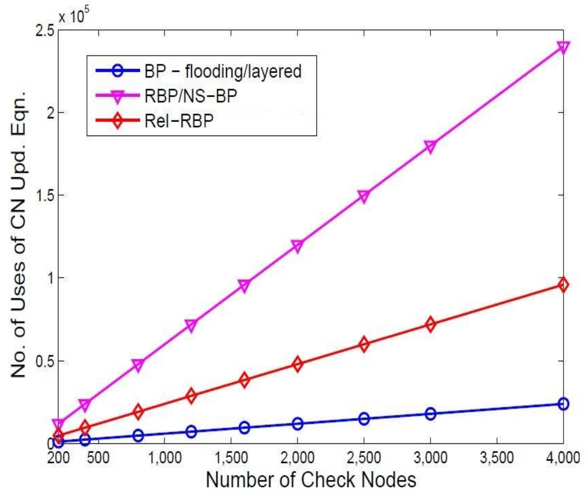
We also provide a graphical illustration of the complexity per iteration. Fig. 6 a) illustrates the variation of complexity for a fixed code rate and degree distributions and increasing block length while Fig. 6 b) shows how the complexity changes for the different degree distributions provided in Tables 1 and 2 of [31], where the average check and variable node degrees were used. The number of uses of the check node update equation (2) is taken as the measure of complexity in both plots. This is justified by the fact that, for the standard BP, the complexity of the check node update is greater than that of the variable node update (3) and thus dominates the overall complexity. Secondly, Rel-RBP and the standard IDS schemes require the same number of uses of the variable node update per classic iteration, and thus will differ in complexity only in the number of check node updates. Additionally, LBP requires approximately the same number of variable node updates as those IDS schemes, whereas standard BP requires fewer uses of the variable node update equation per iteration.

6 Simulations

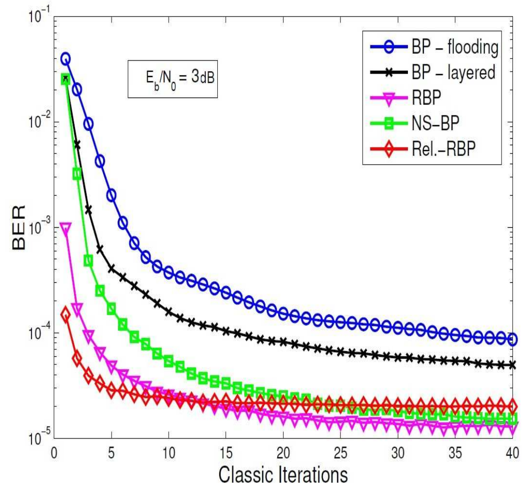
In this section, we evaluate the performance of the proposed and existing decoding algorithms using Monte Carlo simulations. In particular, we consider regular LDPC codes with rate $R = 1/2$, parameters of check and variable nodes given by $(3, 6)$ and block length $n = 480$, irregular LDPC codes with rate $R = 1/2$ that are used by the WiMAX standard [3] and block length $n = 576$ and regular LDPC codes with rate $R = 1/2$, parameters of check and

Table 1 Computational complexity required by decoding algorithms.

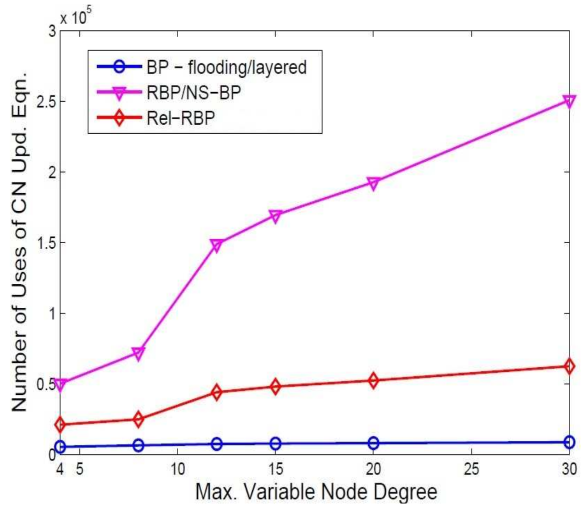
Algorithm	Variable to check node update	Check to variable node update	Message residual	Real-time comparisons
Standard BP	I	I	0	0
LBP	I	I	0	0
RBP	$(d_v-1)I$	I	$(d_v-1)(d_c-1)I$	$(E-1)I/d_v$
NW-RBP	$(d_v-1)I$	I	$(d_v-1)(d_c-1)I$	$(E-1)I/d_v$
Proposed Rel-RBP	$(d_v-1)I$	I	$(d_v-1)(d_c-1)I$	$2(d_v-1)(2(d_c-1)+1)I+(n-k)I$



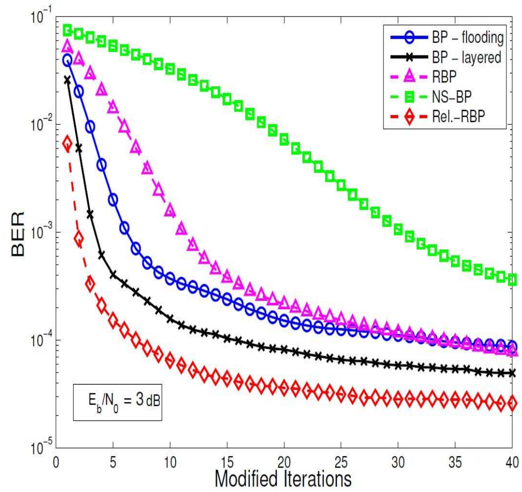
(a)



(a)



(b)

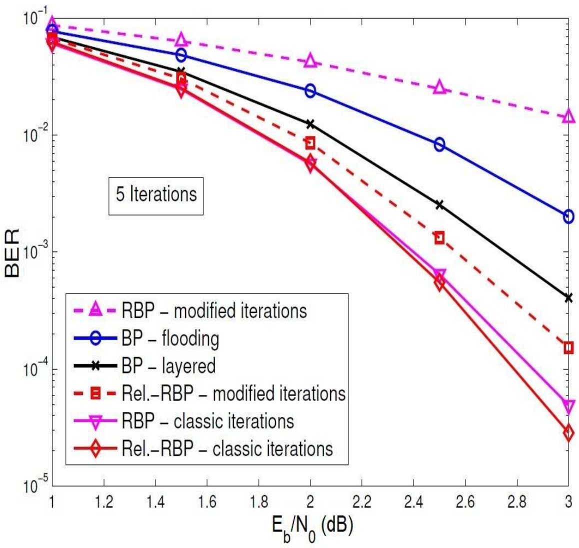


(b)

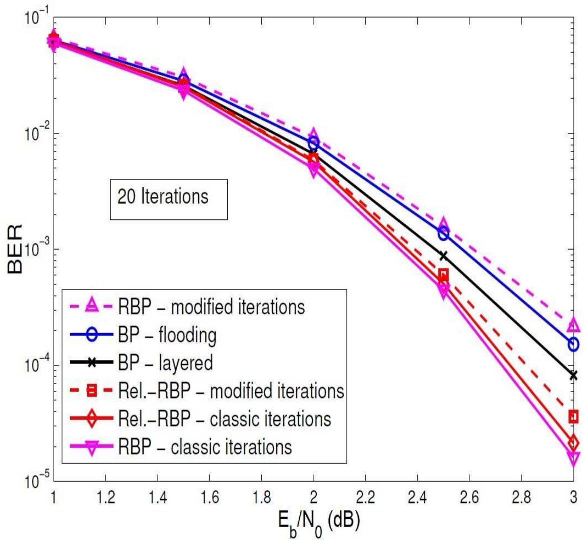
Fig. 6: Computational complexity versus a) block length with $R = 1/2$ and b) different degree distributions.

Fig. 7: a) BER performance versus classic iterations for a regular code with $n = 480$ b) BER performance versus modified iterations for a regular code with $n = 480$.

variable nodes given by $(3, 6)$ and block length $n = 100$. We consider an additive white Gaussian noise channel (AWGN) for all simulations but remark that other channels have also been considered and no significant difference in the performance hierarchy has been observed, which motivated us to focus on the AWGN channel for the sake of simplicity. We have chosen as the figure of merit to assess the proposed and existing decoding iterations the bit error rate (BER) performance. Specifically, we have obtained BER curves against the number of classic iterations, the proposed modified deocoding iterations and the signal-to-noise ratio (SNR).



(a)



(b)

Fig. 8: a) BER performance versus SNR for a regular code with $n = 480$ and a maximum of 5 iterations. b) BER performance versus SNR for a regular code with $n = 480$ and a maximum of 20 iterations.

We illustrate the BER performance versus the classic and modified decoding iterations of the proposed and existing decoding techniques in Fig. 7. The results of Fig. 7 a) indicate that with the classic measure Rel-RBP has excellent performance for a small number of iterations, outperforming the RBP decoding scheme. With a larger number of iterations, RBP and NS-BP outperform Rel-RBP by a small margin. Conversely, the results of Fig. 7 b) show that with the modified decoding iterations Rel-RBP outperforms the RBP, the NS-BP, the LBP and the BP decoding algorithms. In addition, it should be mentioned that the complexity of Rel-RBP is significantly lower than the other decoders as highlighted by the modified decoding iterations that make the decoding cost more evident.

In Fig. 8 we present the BER performance against the SNR of the proposed and existing techniques for the regular LDPC code with $n = 480$ using a fixed maximum number of iterations equal to 5 and 20, respectively. The results of Fig. 8 a) show that for 5 iterations Rel-RBP outperforms RBP using the classic iterations. In particular, the BER curves with the modified decoding iterations indicate that for an equivalent computational complexity Rel-RBP outperforms the standard BP and the RBP algorithms. It should be noted that classic and modified iterations of the standard BP and LBP are equivalent. The results of Fig. 8 b) for a maximum of 20 iterations illustrates that with the classic iterations RBP is slightly better than Rel-RBP. However, with the modified decoding iterations Rel-RBP outperforms the remaining decoding algorithms.

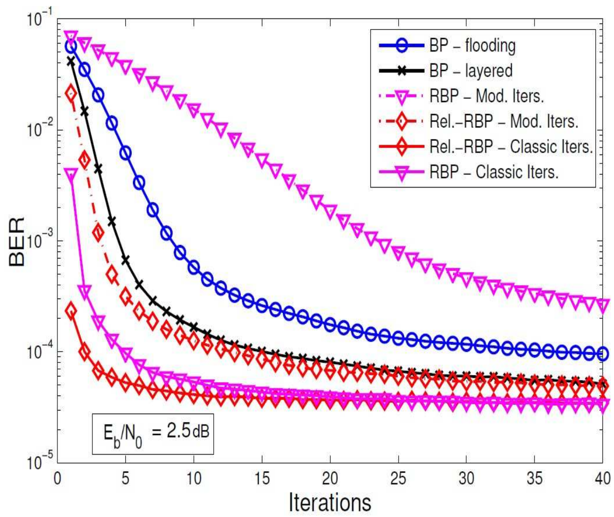
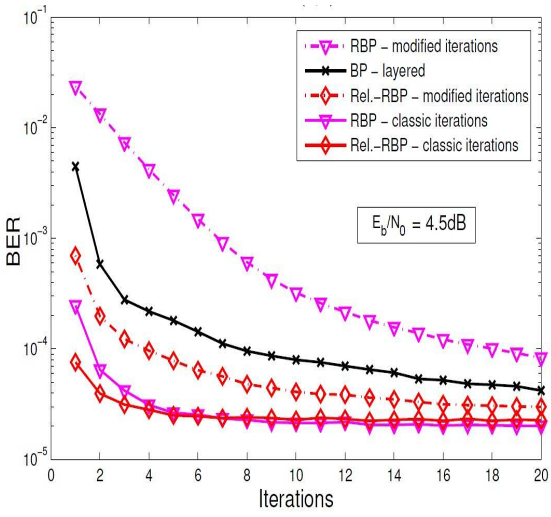


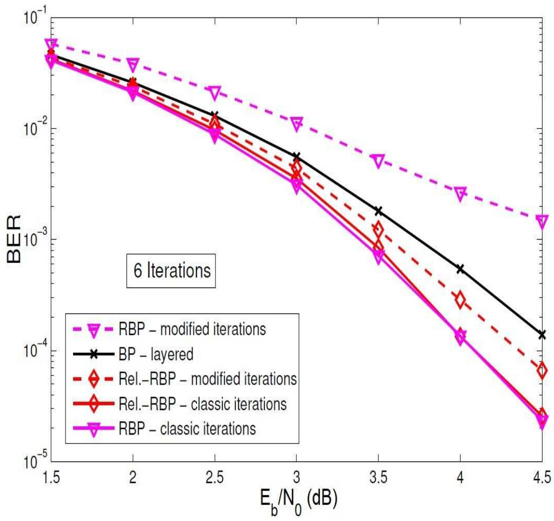
Fig. 9: BER performance versus iterations for the WIMAX irregular code with $n = 576$.

The BER performance against the number of iterations is shown in Fig. 9 a) for irregular LDPC codes of the WIMAX standard using both classic and modified decoding iterations. The proposed Rel-RBP algorithm outperforms the other decoding algorithms with the classic decoding iterations for a small number of iterations and converges to the same level of BER as the RBP algorithm. If we consider the modified decoding iterations then Rel-RBP outperforms RBP, while requiring a significantly lower computational complexity.

In the next examples, we examine a scenario that is envisaged for super dense networks and URLL applications, where the channel coding aspects are significantly constrained in terms of computational cost and latency. For this reason, the use of codes with short blocks and a small number of decoding iterations is considered. In particular, we study the performance of regular LDPC codes with parameters $(3, 6)$ and block length $n = 100$ using the proposed Rel-RBP and existing decoding algorithms. The results showing the BER performance against the number of iterations and SNR are illustrated in Fig. 10. The curves shown in 10 a) are obtained for $\text{SNR} = E_b/N_0 = 4.5$ dB and for a maximum of 20 iterations, and indicate that the BER performance of Rel-RBP is superior to RBP for a small number of iterations and much superior to standard BP for up to 20 iterations, which suggests that the proposed Rel-RBP algorithm is an excellent choice for URLL applications. The plots shown in Fig. 10 b) illustrate the BER performance against SNR using a maximum of 5 decoder iterations for the proposed and existing algorithms, and show that Rel-RBP achieves the best performance together with RBP when considering classic iterations, outperforming by a substantial margin the standard BP algorithm. When considering the modified iterations, Rel-RBP is significantly better than RBP and standard BP.



(a)



(b)

Fig. 10: a) BER performance versus iterations for the regular LDPC code with $n = 100$ and a maximum of 20 iterations. b) BER performance versus SNR for the regular LDPC code with $n = 100$ and a maximum of 6 iterations.

7 Conclusions

In this paper, we have introduced a way of measuring decoding iterations for IDS techniques that is useful for comparison purposes and developed an improved knowledge-based dynamic scheduling scheme, denoted Rel-RBP, based on the reliability of the LLRs exchanged in the message passing. Rel-RBP has been shown to achieve improved performance with reduced complexity with respect to the RBP and NS-BP schemes, resulting in very fast convergence to low levels of BER using LDPC codes with short blocks. The proposed Rel-RBP algorithm may contribute to addressing stringent requirements of latency and energy efficiency of future wireless systems.

8 References

1 R. G. Gallager, “Low-density parity-check codes," *IRE Transactions on Information Theory*, vol.8, No.1, pp. 21-28, 1962.

2 IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band, IEEE Std 802.11ad-2012, Dec. 28 2012.

3 IEEE Standard for Local and Metropolitan Area Networks: Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Com- bined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1, IEEE 802.16e, Feb. 2006.

4 Digital Video Broadcasting (DVB), Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2), ETSI EN 302 307 V1.3.1 (2013-03) , 2013.

5 IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements, Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE Std. 802.3an, Sep. 2006.

6 C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski and A. Dekorsy, “Massive machine-type communications in 5G: physical and MAC-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59-65, September 2016.

7 H. Jin, A. Khandekar, R. J. McEliece, “Irregular repeat accumulate codes", *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, pp.1-8, 2000.

8 S. Myung, K. Yang, and J. Kim, “Quasi-Cyclic LDPC Codes for Fast Encoding", *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2894-2901, Aug. 2005.

9 Z. Li, L. Chen, L. Zeng, S.Lin, and W. H. Fong, “Efficient Encoding of Quasi-Cyclic Low-Density Parity-Check Codes", *IEEE Transactions on Communications*, vol. 54, no. 1, Jan. 2006, pp. 71-81.

10 X. Y. Hu, E. Eleftheriou and D. M. Arnold, “Regular and irregular progressive edgegrowth tanner graphs", *IEEE Transactions on Information Theory*, vol.51, No.1, pp.386-398, Jan. 2005.

11 A. G. D. Uchoa, C. Healy, R. C. de Lamare and R. D. Souza, "Design of LDPC Codes Based on Progressive Edge Growth Techniques for Block Fading Channels," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1221-1223, November 2011.

12 C. T. Healy and R. C. de Lamare, "Design of LDPC Codes Based on Multipath EMD Strategies for Progressive Edge Growth," *IEEE Transactions on Communications*, vol. 64, no. 8, pp. 3208-3219, Aug. 2016.

13 M. P. C. Fossorier, M. Mihaljevic, H. Imai, “Reduced complexity iterative decoding of low-density parity check codes based on belief propagation", *IEEE Transactions on Communications*, vol.47, No.5, pp.673-680, May 1999.

14 J. Chen, M. P. C. Fossorier, “Density evolution for two improved BP-Based decoding algorithms of LDPC codes", *IEEE Communications Letters*, vol.6, no.5, pp.208-210, May 2002.

15 J. Chen and M. P. C. Fossorier, “Near-optimum universal belief propagation based decoding of low-density parity check codes", *IEEE Transactions on Communications*, vol.50, No.3, pp.406-414, Mar. 2002.

16 J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, X.-Y. Hu, “Reduced-Complexity Decoding of LDPC Codes", *IEEE Transactions on Communications*, vol.53, No.8, pp.1288-1299, Aug. 2005.

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

18 H. Wymeersch, F. Penna and V. Savic, “Uniformly Reweighted Belief Propagation for Estimation and Detection in Wireless Networks", *IEEE Transactions on Wireless Communications*, vol.11, no.4, pp.1587-1595, Apr. 2012.

19 J. Liu and R. C. de Lamare, “Low-Latency Reweighted Belief Propagation Decoding for LDPC Codes", *IEEE Communications Letters*, vol.16, no.10, pp.1660-1663, Oct. 2012.

20 G. Elidan, I. McGraw and D. Koller, “Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing", *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI)*, July 2006.

21 D. E. Hocevar, “A reduced complexity decoder architecture via layered decoding of LDPC codes", *IEEE Workshop on Signal Processing Systems*, pp.107-112, Oct. 2004.

22 J.-H. Kim, M.-Y. Nam, H.-Y. Song, and K.-M. Lee, “Variable-to-Check Residual Belief Propagation for informed dynamic scheduling of LDPC codes", *Proc. of International Symposium on Information Theory and Its Applications ISITA 2008*, pp.1-4, Dec. 2008.

23 S. Kim, K. Ko, J. Heo, J.-H. Kim, “Two-staged informed dynamic scheduling for sequential belief propagation decoding of LDPC codes", *IEEE Communications Letters*, vol. 13, no.3, pp.193-195, March 2009.

24 A. I. Vila Casado, M. Griot and R. D. Wesel, “LDPC Decoders with Informed Dynamic Scheduling", *IEEE Transactions on Communications*, vol.58, no.12, pp.3470-3479, Dec. 2010.

25 C. Healy and R. C. de Lamare, “Knowledge-aided informed dynamic scheduling for LDPC decoding," *2015 IEEE International Conference on Communication Workshop (ICCW)*, London, 2015, pp. 2212-2217.

26 C. T. Healy, “Short-length Low-density Parity-check Codes: Construction and Decoding Algorithms", PhD thesis, University of York, 2015.

27 M. Beermann and P. Vary, “A Generalization of Residual Belief Propagation for Flexible Reduced Complexity LDPC Decoding", *Proc. IEEE Vehicular Technology Conference (VTC Fall)*, 2011, pp.1-5, Sept. 2011.

28 Y. Gong, X. Liu, W. Yecai and G. Han, “Effective Informed Dynamic Scheduling for Belief Propagation Decoding of LDPC Codes", *IEEE Transactions on Communications*, vol.59, no.10, pp. 2683-2691, October 2011.

29 H.-C. Lee, Y.-L. Ueng, S.-M. Yeh and W.-Y. Weng, “Two Informed Dynamic Scheduling Strategies for Iterative LDPC Decoders", *IEEE Transactions on Communications*, vol.61, No.3, pp.886-896, March 2013.

30 K. A. Aslam, Y. L. Guan and K. Cai, “Edge-Based Dynamic Scheduling for Belief-Propagation Decoding of LDPC and RS Codes", *IEEE Transactions on Communications*, 2017.

31 T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, “Design of capacityapproaching irregular low-density parity-check codes", *IEEE Transactions on Information Theory*, vol.47, no. 2, pp. 619-637, Feb. 2001.