A Reliability Profile Based Low-Complexity Dynamic Schedule LDPC Decoding

RUIJIA YUAN\textsuperscript{1}, (Member, IEEE), TIANJIAO XIE\textsuperscript{1}, (Member, IEEE), AND ZHONGFENG WANG\textsuperscript{2}, (Fellow, IEEE)

\textsuperscript{1}China Academic of Space Technology, Xi’an 710100, China
\textsuperscript{2}School of Electronic Science and Engineering, Nanjing University, Nanjing 210008, China

Corresponding author: Tianjiao Xie (xiexietianjiao@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61801377.

ABSTRACT In order to improve the Bit Error Rate (BER) performance of Low-Density Parity-Check (LDPC) codes, in this paper, we propose a Reliability Profile (RP) Based Low-Complexity Dynamic Schedule, called RPD. In terms of dynamic scheduling, the new RPD method is distinct from the existing Residual Belief Propagation (RBP) and its variations, since RPD is based on residual criterion. Reliability divides nodes message into two categories: reliable and unreliable. By using as many messages as possible from the latest available current iteration instead of the previous iteration to update unreliable messages multiple times to speed up the convergence speed and increase the total number of updates of effective nodes message, so as to improve the BER performance. By not updating the reliable message, the number of ineffective nodes message updates is reduced, thereby reducing the decoding complexity. Besides, the more iterations increase, the fewer calculations of these reliable nodes. Moreover, based on reliability profile, RPD has an advantage over RBP for dynamic scheduling in that the former only needs integer operations, while the latter requires float operations. Therefore, the RPD algorithm not only accelerates the convergence speed of LDPC decoding and but also improves the BER performance while reducing the complexity. The analysis and simulation results show that the RPD strategy not only retains the fast-convergence advantage of RBP, but also has a better convergence BER performance compared to that of node-wise residual belief propagation (NW RBP). Furthermore, the simulation results show that the RPD can significantly improve the convergence speed of protograph LDPC decoding.

INDEX TERMS LDPC decoding, reliability profile, dynamic schedule, residual, protograph LDPC codes.

I. INTRODUCTION

LDPC decoding, reliability profile, dynamic schedule, residual, protograph LDPC codes.

It is well-known that Low-Density Parity-Check (LDPC) codes \cite{1} can obtain performance close to the capacity limit performance by using the sum-product algorithm (SPA) with the Flooding decoding algorithm (FA) \cite{2}. Since FA is a fully parallel iterative soft-decision decoding algorithm, it usually requires up to 30 to 50 iterations to converge to the ideal BER performance. Hence, in order to reduce the algorithm complexity, it is necessary to speed up the convergence speed of the LDPC decoding. Among current iteration reduction schemes, the most typical one is the Serial scheduled decoding Algorithm (SA) \cite{3}, which converges almost twice as fast as the FA by incorporating row layered decoding and column layered decoding approaches. Shortly afterwards, Casado \textit{et al.} \cite{4} observed that the convergence speed can be further improved if the LDPC decoding schedules are dynamic. i.e., message updates are not predetermined by the sequence of the row or the column of the parity-check matrices. Based on the observation, the first published Informed Dynamic Scheduling (IDS) for LDPC decoding strategy is the Residual Belief Propagation (RBP) algorithm, which defines residual information by the magnitude of the difference between the current and the previous values of the message updating. Propagating the message with large differences priority makes SA converge faster. RBP is a greedy algorithm. Although the Bit Error Rate (BER) of RBP decreases very fast for the first few iterations, its convergence error correction performance is worse than that of both FA and SA. In order to achieve a superior convergence performance, Node Wise
RBP (NW-RBP) was proposed by Casado in [5]. NW-RBP does not converge as rapidly as RBP, but it does converge to the correct codeword far more often.

Followed by these various modified versions, different types of dynamic schedule strategies for LDPC decoding are proposed. Informed Variable-to-Check Residual Belief Propagation (IVC RBP) [6] effectively restrains the influence of cycles in the Tanner graph. Adaptive IDS in [7] changes decoding process adaptively according to the state of the BER performance. Quota-based RBP (Q-RBP) and Silent-Variable-Node-Free RBP (SVNF-RBP) [8] break the greedy group and static variable node phenomenon of RBP respectively, and achieve the comparable convergence speed of RBP and the decoding performance of NW-RBP. In [9], the simulated annealing method is introduced into dynamic scheduling decoding to solve the problem of greedy algorithm. In [10], a posteriori probability RBP dynamic scheduling algorithm is proposed to improve the BER performance under low channel conditions. The importance of dynamic selection strategy is analyzed and a dynamic selection strategy based on triple judgment is proposed in [11] to help locate the most unreliable message quickly and accurately. In [12], a dynamic silent-variable-node-free scheduling (D-SVNFS) algorithm is proposed, which can alleviate and the convergence performance of error correction. In [13], a knowledge-aided IDS decoding algorithm [13] exploits the reliability of the messages to refine the exchange of information in the graph. To solve the problem of poor convergence BER performance of RBP decoding based on greediness, a residual message transfer algorithm based on residual attenuation is proposed [14]. It can prevent decoding resources from being unreasonably occupied by a small number of edges in the Tanner graph. So the greedy phenomenon can be alleviated and the convergence performance of error correction is than RBP. Aiming at the slow convergence speed of NR LDPC codes in 5G, a residual based hierarchical confidence propagation algorithm is proposed in [15] by dynamically rearranging the layers between different iterations. This algorithm reduces the decoding complexity with a small BER performance loss.

These dynamic schedule decoding algorithms determine the sequence of message updates based on the residual [5], which is defined by the difference between the numerical value of a message before and after an update. RBP and its derived methods are common with high complexity for the updates sequence since the residuals are computed by the log-likelihood ratios (LLRs), in the form of floating data or quantized bits. The hardware implementation for these IDS decoders are challenging.

In this paper, we propose a Reliability Profile (RP) Based Low-Complexity Dynamic Schedule LDPC Decoding. The RP is employed to classify reliable nodes and unreliable nodes. It should be noted that the concept of RP is first introduced in the context of bit-flipping algorithm to decide which variable needs to be modified [16]. In this paper, RP is introduced in a novel scenario, where the message sequence for updates is determined by RP, rather than residuals adopted in the exiting methods. RP can output the sequence of variable-to-check update. By using as many messages as possible from the latest available current iteration instead of the previous iteration, the RDP updates unreliable message multiple times to speed up the convergence speed and increase the total number of updates of effective node message, so as to improve the bit error rate (BER) performance. By not updating the reliable message, the number of ineffective nodes message updates is reduced, thereby reducing the decoding complexity. Besides, the more iterations increase, the fewer calculations of these reliable nodes. Moreover, based on the RP, RPD has an advantage over RBP for dynamic scheduling in that the former only needs integer operations, while the latter requires float operations. Therefore, the RPD algorithm not only accelerates the convergence speed of LDPC decoding but also improves the BER performance.

II. PRELIMINARIES

In this section, RP and its related definitions are explained in detail to further introduce the proposed algorithm. A binary LDPC code is defined by an $M \times N$ binary parity-check matrix $H, N, M$, and $K = N - M$ denote code length, parity-check length, and the number of information bits, respectively. Set $\hat{x} = [\hat{x}_1, \cdots, \hat{x}_n, \cdots, \hat{x}_N]$ as a received codeword vector under hard-decision.

**Definition 1:** Syndrome $S$ is an $M$-tuple over GF(2) defined by

$$S = [s_1, \cdots, s_m, \cdots, s_M] = \hat{x}^T.$$  \hspace{1cm} (1)

The syndrome is used to check whether a received $\hat{x}$ contains transmission errors. If any element of $S$ is nonzero, a corresponding parity-check failure (PF) is detected. The number of PFs of $\hat{x}$ equals the number of nonzero elements in the $S$.

**Definition 2:** Syndrome sum of $\hat{x}$ is defined by

$$s_m = \hat{x}_n h_m = \sum_{k=1}^{N} s_k h_{m,k}, \hspace{0.5cm} 0 < m \leq M.$$  \hspace{1cm} (2)

From Definition 1, if $s_m$ contains a received $\hat{x}_n$, $s_m$ is called to check on $\hat{x}_n$.

**Definition 3:** The reliability of $\hat{x}_n$:

$$f_n = \sum_{m \in M(n)} s_m.$$  \hspace{1cm} (3)

where $M(n)$ denotes the set of check nodes connected to variable node $n$.

The number of PFs on a received bit $\hat{x}_n$, i.e., $f_n$ gives a measure of the reliability of $\hat{x}_n$. The larger $f_n$, the less reliable $\hat{x}_n$ is. Conversely, the smaller $f_n$, the more reliable $\hat{x}_n$ is.

**Definition 4:** The reliability profile (RP) of $\hat{x}$ is an $N$-tuple defined by

$$f = [f_1, \cdots, f_n, \cdots, f_N].$$  \hspace{1cm} (4)
By sorting elements of $f$ in descending order, a new criterion is presented. Let $\text{order}_1, \cdots, \text{order}_N$, which is an integer $N$-tuple, denotes the sequence of messages update for dynamic scheduling. Then, this integer $N$-tuple satisfies the constraint that $f_{\text{order}_1} \geq f_{\text{order}_2} \geq \cdots \geq f_{\text{order}_N}$.

From Definition 3, it is clear that the message corresponding to variable node $\text{order}_1$ is the least reliable, and $\text{order}_N$ is the most reliable among the sequence of messages. From previous works, we know the messages tend to have high reliability when an when an iterative algorithm converges. By contrast, if a message has less reliability, the message located in a certain node has not converged. Therefore, firstly propagating the message with the least reliable should speed up the convergence.

The criteria for both reliability and unreliability are defined to further reduce the decoding complexity. We consider a check node as an unreliable one if the syndrome of the check node is not equal to zero. Likewise, unreliable variable nodes are defined as connecting one or more unreliable check nodes. In this paper, we obtained through residual messages as the numbers of unreliable check nodes linked by variable nodes. If the variable node is connected by more unreliability check nodes, its residual message is larger. Moreover, the value of residual message can be further utilized in our algorithm if its value is zero, i.e., the corresponding variable node satisfies all check equations. Therefore, it has very high reliability. By ignoring the calculations of these highly reliable nodes, the computation in each iteration can be dramatically reduced. Besides, as the number of iterations increases, the number of check nodes with zero syndromes sharply increases. Based on the developed techniques, significant decoding computation reductions can be achieved.

III. PROPOSED RPD ALGORITHM

Let $m$ and $n$ denote a check and variable node, respectively. $M(n)$ denote the set of check nodes connected to variable node $n$, and let $N(m)$ denote the set of variable nodes connected to check node $m$. The exclusion of an element $n$ from $N(m)$ or $m$ from $M(n)$ is denoted by $N(m) \setminus n$ or $M(n) \setminus m$, respectively. Let $Q_{m,n}$ represent the message from variable node $n$ to check node $m$, called V2C and let $R_{m,n}$ represent the message from check node $m$ to variable node $n$, called C2V. V2C and C2V are collectively referred to as extrinsic messages. Let $Q_n$ be the channel message of variable node $n$. It is assumed that all the messages are measured using log-likelihood ratio (LLR). Let $i_{\text{max}}$ be the maximum number of iterations. The proposed Reliability Profile Based Low-complexity dynamic schedule LDPC decoding can be described as follows.

Initialization:

Each variable node $n$ is assigned a posteriori LLR,

$$Q_n^{(0)} = \log(P(x_n = 0|y_n)/P(x_n = 1|y_n)).$$

for every position $(m, n)$ such that $H_{m,n} = 1, R_{m,n}^{(0)} = 0$.

Decision:

a) The decision code $\hat{x} = [\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_N]$ where $\hat{x}_n = 0$ if $Q_n^{(i-1)} \geq 0, \hat{x}_n = 1$ if $Q_n^{(i-1)} < 0$, where $i$ denotes the iteration index.

b) The syndrome of each check node can be obtained by multiplying $\hat{x}$ with the transpose of the parity check matrix $H$, i.e. $S = [s_1, s_2, \cdots, s_M] = \hat{x}H^T$. If $S = 0$, the algorithm is halt, and $\hat{x}$ is reported as the decoder output. Otherwise, go to step Decision c). If $i$ is a certain number of decoding iterations $i_{\text{max}}$ and $S \neq 0$, then a decoding failure is reported.

c) Compute reliability profile $f$: Let $N$ denote the number of column of a parity check matrix. The vector $f = (f_1, f_2, \cdots, f_N)$ is the reliability profile.

d) The order of nonzero elements in the reliability profile $f$ are arranged from large to small. Supposing the number of nonzero equal to $N^\ast(\leq N)$. The variable nodes $n$ corresponding to the $f$ order $\text{order}_1, \cdots, \text{order}_N^\ast$ need to be found. Iterative procedure should follow the sequence of $\text{order}_n$, $0 < n < N^\ast$ by going to Iterative Processing. Otherwise, $f_n = 0$ corresponding to the case means that variable nodes $n$ do not need to be updated.

Iterative Processing:

For the column $n = \text{order}_n$, the $i$-th decode iteration, $0 < i < i_{\text{max}}$, the executed procedure is listed:

1) Compute the posteriori of this iteration:

$$Q_n^{(i)} = Q_n^{(i-1)}$$

2) Compute the variable nodes extrinsic message of the current column:

$$Q_{mn}^{(i)} = Q_{mn}^{(i-1)} - \sum_{m'' \leq M(n)\setminus m \text{ and } m'' < m} R_{m'n}^{(i)} - \sum_{m'' \leq M(n)\setminus m \text{ and } m'' > m} R_{m'n}^{(i-1)}.$$  \hspace{1cm} (7)

3) Update the check nodes extrinsic message of the current row:

$$R_{mn}^{(i)} = \prod_{n' \in N(m)\setminus n} \tanh \frac{Q_{mn'}^{(i)}}{2}.$$  \hspace{1cm} (8)

4) Update variable nodes posterior probability message of the current column:

$$Q_n^{(i)} = Q_n^{(i)} + \sum_{m' \leq M(n)\setminus m} R_{m'n}^{(i)}.$$  \hspace{1cm} (9)

RBP or NW-RBP calculates the residual from the difference of C2V message values before and after the update and sequentially updates the variable node corresponding to the chosen edge. However, RPD calculates the RP of all the variables and sequentially updates the check nodes corresponding to the chosen edge.

The proposed RPD decoding is analyzed as follows. Initially, all the messages $Q_{nm}$ are set to the value of their corresponding channel message $C_n$. Compute all the reliability of variable nodes (Eq. (3)) and get the RP (Eq. (4)). Assuming that $n$ is the least reliable variable node, all the outgoing messages from the variable node $n$, $m \in M(n)$
Algorithm 1 The Proposed RPD Decoding for LDPC Codes

1: Initialize: $C_n = Q_{nm}^{(0)}$, $\forall n$ (Eq. (5))
2: for all $i = 1, 2, \ldots, i_{max}$ do
3: Compute $f = [f_1, f_2, \ldots, f_N]$ (Eq. (1) to (4))
4: Generate the order $order_1, \ldots, order_N$
5: for all $n = order_1, order_2, \ldots, order_N$ do
6: for all check nodes $m \in M(n)$ do
7: Generate and propagate $Q_{nm}^{(i)}$ (Eq. (7))
8: Generate and propagate $R_{nm}^{(i)}$ (Eq. (8))
9: Generate posterior messages $Q_n^{(i)}$ (Eq. (9))
10: end for
11: end for
12: if $S \neq 0$ and $i < i_{max}$ then
13: Goto Line 3
14: end if
15: end for

will be updated and propagated. After check-to-variable messages $R_{nm}$ are propagated, propagate and update the variable-to-check nodes message $Q_{nm}$. $i_{max}$ represents the number of maximum iterations. The proposed RPD Algorithm is explained exactly in Algorithm 1.

Within the RDP algorithm, syndrome $S = \hat{x} \times H^T$, where $\hat{x}$ is the sign of $Q_{nm}^{(0)}$.

IV. ANALYSIS OF THE PERFORMANCE AND COMPLEXITY

Convergence, BER performance, and complexity are important indicators reflecting the decoding performance. The proposed RDP algorithm is mainly analyzed from these three aspects.

A. ANALYSIS OF THE CONVERGENCE AND BER PERFORMANCE

For LDPC codes with iterative decoding algorithm, BER performance is an important index to reflect the decoding algorithm. For resource constrained systems, the convergence of the decoding algorithm is also significant. The faster convergence speed means the fewer iterations, and we can get better decoding performance the limited number of iterations. The BER performance of the decoding algorithm is directly proportional to the total number of effective node updates. The convergence of decoding is directly proportional to the update speed of node message. The decoding performance of the proposed RPD algorithm is analyzed from these two levels. In order to analyze the message updating process of LDPC decoding algorithm based on the idea of message propagation, several existing typical decoding algorithms are elaborated from simple to deep.

Let

$$ E = \sum_{c=1}^{M} d_c = \sum_{v=1}^{N} d_v $$

denote the number of non-zero elements of LDPC check matrix, where $d_c$ represents the number of non-zero elements in row $c$ and $d_v$ represents the number of non-zero elements in column $v$.

FA algorithm: the input of variable node update is C2V and the output is V2C. On the contrary, the input of check node update is V2C and the output is C2V. The numbers of C2V or V2C are E. For one iteration, in fact, the message propagated from the last iteration is used. Therefore E C2V and E V2C messages are modified only once in one iteration, and that is, the number of message updated in one iteration for FA is one, while the total numbers of calculations for V2C and C2V message updated are E and E respectively.

Row SA (R-SA) and its modified versions: the C2V updated in the first row of current iteration uses the V2C of the previous iteration, and the C2V in the next row of current iteration uses the V2C calculated in current iteration. Therefore, in one iteration, the C2V is modified $d_c$ times, and the update speed of the node is accelerated compared with FA. Therefore, the convergence speed of R-SA is twice faster than that of FA. However, the total number of calculations of V2C and C2V is still $E$.

Column SA (C-SA) and its modified versions: the V2C updated in the first column of current iteration uses the C2V of the previous iteration, and the V2C in the next column of current iteration uses the C2V calculated in current iteration. Therefore, in one iteration, the V2C is modified $d_v$ times, and the update speed of the node is accelerated compared with FA. Therefore, the convergence speed of C-SA is twice as fast as that of FA. However, the total number of calculations of V2C and C2V is still $E$.

The proposed RPD algorithm: prioritize the update and simultaneously propagate the message of all $d_v \times d_c$ check nodes connected to the $d_v$ variable nodes with the lowest reliability. Then update the message of all $d_v \times d_c$ variable nodes connected to these check nodes. In one iteration, for any variable and check node, the results of $d_v \times d_c$ are actually modified, and the update speed is twice faster than FA, $dv$ times faster than R-SA, and $dc$ times faster than C-SA. Therefore, the node message updating speed of the RPD algorithm is the fastest, which is the main reason for the fast convergence of the decoding performance.

From the above elaboration, it can be seen that in various classical decoding algorithms, the total number of node message updated in each iteration is $E$. Therefore, when the number of iterations is sufficient, FA, R-SA, and C-SA can basically converge to the same decoding performance. When the proposed RPD update strategy is adopted, for each iteration, if all nodes are updated, the total number of C2V updates will be

$$ \sum_{v=1}^{N} d_v/M = E/M $$
times of R-SA, and the total number of V2C updates will be

$$ \sum_{c=1}^{M} d_c/N = E/N $$
times of R-SA. Since the BER performance of the decoding algorithm is proportional to the total number of node updates, this RPD strategy that updates all reliable and unreliable nodes multiple times has better decoding performance. But correspondingly, the computational complexity of each iteration is also increased by several times.

In order to reduce the complexity, it is necessary to reduce the number of ineffective node message calculations as much as possible. If the syndrome of a check node is not zero, it will be regarded as an unreliable check node. If a variable node is connected to one or more unreliable check nodes, it will be regarded as an unreliable variable node. When only unreliable nodes are updated, the total number of message updated in each iteration can be reduced, but the decoding performance will not be affected. Moreover, as the number of decoding iterations per frame increases, the number of reliable nodes will increase dramatically which can greatly reduce the decoding complexity. Therefore, the RPD algorithm improves the BER performance while reducing the complexity by increasing the total number of effective node message updates and reducing the number of ineffective node message updates.

The RPD algorithm only updates the variable nodes connected to the error check nodes, but for the variable nodes connected to the none error check nodes, it does not mean that those variable nodes are correct. The algorithm uses the following method to correct them. In current iteration, there may be incorrect variable nodes with correct check node but actually unreliable. Although these incorrect variable nodes are not updated in this iteration, they will be connected to other variable nodes that are connected to the same check node gradually corrected. The syndrome of this check node will become non-zero, and such fake reliable variable node can be updated in the next iteration.

In summary, for the proposed RPD algorithm, the convergence speed increases for two reasons. On the one hand, for any variable and check node message in each iteration, \( d_v \times d_c \) results are actually modified. Speeding up the message update speed is the main reason by using as much message as possible from current iteration, which is the main reason for the rapid convergence of the RPD algorithm. On the other hand, updating the nodes by the order of reliability will also speed up the decoding convergence. The reason for the excellent BER performance is as follows. In each iteration, the unreliable nodes, which is in current iteration are updated multiple times to increase the total number of updates of effective node message. This is the main reason for the improvement of the BER performance. However, the reliable nodes in the current iteration are not updated, which reduces the number of ineffective node message updates, thereby reducing the complexity of the decoding algorithm. Therefore, the RPD algorithm not only accelerates the convergence speed of LDPC decoding but also improves the BER performance while reducing the complexity.

### B. The Computation Complexity of Total Nodes Update Message

On the one hand, for the RPD algorithm, the nodes calculation formula includes variable node update (formula 7) and check node update (formula 8), both of which have high complexity. Therefore, the total number of nodes message calculation is the main factor determining the complexity of decoding algorithm. The reduction strategy of RPD algorithm for the total number of nodes message calculations includes two aspects. One is to reduce the iterations by improving the convergence speed, so as to save the total number of node message calculations. The second is to reduce the computational complexity of message by not updating reliable nodes to offset the increased complexity caused by multiple updates of unreliable node. That is why the RPD algorithm with faster convergence speed and the existing method that all nodes are updated once, which case has lower complexity.

In terms of quantitative analysis, the block size \((N, K) = (1944, 972)\) and the coding rate \(R = 1/2\) LDPC code selected form the IEEE 802.11n standard is used to show the total number of variable and check node message updates for different LDPC decoding strategies, including FA, R-SA, C-SA, and the proposed RPD algorithm. For LDPC(1944, 972), the number of non-zero elements in the check matrix is \(E = 6966\). The proportion of non-zero elements to the total elements of the check matrix is \(6966/1944/972 = 0.37\%\).

For one frame, when \(E_b/N_0 = 2.0\) dB, FA algorithm requires 10 iterations to correct decoding and the corresponding syndromes at each iteration are 400, 286, 245, 225, 172, 104, 86, 36, 9, 2 and 0. R-SA or C-SA algorithm requires five iterations to converge and the syndrome of five iterations is 400, 200, 67, 22, 5 and 0. While the proposed RPD algorithm only needs three iterations to converge and the corresponding syndrome at each iteration is 400, 127, 7 and 0.

The total number of node updates for each iteration of the FA algorithm is 6,966, and 69,660 node information needs to be calculated for 10 iterations. While R-SA or C-SA only needs 5 iterations to obtain the same BER performance as the FA algorithm needs 10 iterations. The total node message calculation amount for the FA, R-SA, and C-SA three types of algorithms are the same, which is 34830. For the proposed RPD algorithm, the total number of C2V updates are 6966/972 = 7.17 times of R-SA or C-SA, and the total number of V2C updates is 6966/1944 = 3.58 times of R-SA or C-SA. In the first iteration, there are 400 unreliable check nodes, and 1481 unreliable variable nodes connected to them. The total number of C2V and V2C updates is 6966 × 7.17 × 400/972 = 20554, and 6966 × 3.58 × 1481/1944 = 19000, respectively. In the second iteration, there are 127 unreliable check nodes, and 657 unreliable variable nodes connected to them. The total number of C2V and V2C updates is 6966 × 7.17 × 127/972 = 6526, and 6966 × 3.58 × 657/1944 = 8428, respectively. In the third iteration, there are 7 unreliable check nodes, and 46 unreliable...
variable nodes connected to them. The total number of C2V and V2C updates is 6966 × 7.17 × 7/972 = 360, and 6966 × 3.58 × 46/1944 = 590, respectively. It can be seen that for the frame with Eb/N0=2.0dB, the proposed RPD algorithm needs three iterations. The total number of C2V updates is 20554+6526+360=27440, accounting for 78.8% of R-SA or C-SA, and the total number of V2C updates is 19000+8428+590=28018, accounting for 80.4% of R-SA or C-SA. Therefore, the proposed RPD algorithm has lower computational complexity for the numbers of updating message.

For decoding one frame data, compared with R-SA or C-SA, the proposed RPD algorithm saves the updates calculation of C2V and V2C up to 1-78.8% = 21.1% and 1-80.4% = 19.6%, respectively. Compared with FA, the proposed RPD algorithm saves the updates calculation of C2V and V2C up to 2 × 21.1% = 42.2% and 2 × 19.6% = 39.2%, respectively. Table 1 shows the number of required node updates of three types of algorithms for one frame data at Eb/N0 = 2.0dB.

To get more accurate comparison of average decoding complexity from a statistical point of view, we evaluate the complexity of (1944, 972) LDPC decoded with three algorithms by simulations, and the statistics average computation complexity from a statistical point of view, we evaluate the complexity of (1944, 972) LDPC codes with sparse matrix. From Eq. (2), Eq. (3), and Eq. (4), N integer adders and comparisons with \([\log_2 g]\) bits precision are used for the computation of \(f \) and finding order of their elements, respectively. Obviously, they are all integer operations.

By contrast, the existing residual is the absolute (ABS) value of the difference between the numerical values of a message before and after an update. Therefore, \(M \times w = N \times g \) float (or \(Q \) bits quantized precision) additions and ABS operations are used to calculate the residuals, where \(w \) denotes the row weight of parity-check matrix. The same number of comparisons is needed to determine the sequence of check node update. NW-RBP performs comparisons many times in one iteration.

As shown in Table 1 and Table 2, the proposed RPD algorithm gives up updating the message of variable nodes and check nodes that are considered reliable during each iteration. As the iterations increases, the number of syndromes becomes zero sharply. Increase, reduce the number of unreliable variable nodes and check nodes, which will greatly reduce the complexity required for decoding. Therefore, the total number of nodes message calculations are less complex than the existing decoding algorithm as all nodes are updated once. As shown in Table 2, In the case of low SNR, the more obvious the iterations decreases, so it has lower complexity. Compared with FA, R-SA, and C-SA algorithm, the proposed RPD algorithm saves the ratio of the number of variable nodes and checks node update calculations will increase as the SNR decreases.

### C. The Computation Complexity of the Reliability Profile

For the residual-based IDS algorithm, residual needs to be recalculated for each check node update in each iteration, \(M \) (the number of rows of the check matrix) check nodes need to be recalculated and compared \(M \) times, and these pre-calculated residuals need to be abandoned in the actual update. The RPD algorithm proposed only needs to calculate the reliability of each variable node once in an iteration, and number of calculations is only \(1/M \) of residue-based IDS algorithm. The complexity can be reduced when the floating-point computation for the residual are replaced by the integer computation for the syndrome, i.e., the reliable in the proposed RDP algorithm. And the complexity comparison is shown in Table 3.

The RP needs to calculate \( f = [f_1, f_2, \cdots, f_N] \) and find their order. From Eq. (1) to Eq. (4), \( N \times g \) binary adders are used to calculate syndrome sum \( s_m \), the maximum of \( s_m \) is \( g \), where \( g \) denotes the column weight of parity-check matrix (The average of \( g \) is commonly less than 32 for LDPC codes with sparse matrix). From Eq. (2), Eq. (3), and Eq. (4), \( N \) integer adders and comparisons with \([\log_2 g]\) bits precision are used for the computation of \( f \) and finding order of their elements, respectively. Obviously, they are all integer operations.

As shown in Table 1 and Table 2, the proposed RPD algorithm gives up updating the message of variable nodes and check nodes that are considered reliable during each iteration. As the iterations increases, the number of syndromes becomes zero sharply. Increase, reduce the number of unreliable variable nodes and check nodes, which will greatly reduce the complexity required for decoding. Therefore, the total number of nodes message calculations are less complex than the existing decoding algorithm as all nodes are updated once. As shown in Table 2, In the case of low SNR, the more obvious the iterations decreases, so it has lower complexity. Compared with FA, R-SA, and C-SA algorithm, the proposed RPD algorithm saves the ratio of the number of variable nodes and checks node update calculations will increase as the SNR decreases.

### TABLE 1. Comparison of required nodes updates for three types of algorithms at \( E_b/N_0 = 2.0dB \).

| Algorithm       | C2V updates | V2C updates | C2V saved | V2C saved |
|-----------------|-------------|-------------|-----------|-----------|
| RPD             | 27540       | 28018       | 21.1%     | 42.2%     |
| R-SA or C-SA    | 34830       | 34830       | 19.6%     | 39.2%     |
| FA              | 69660       | 69660       | 19.6%     | 39.2%     |

### TABLE 2. Comparison of C2V and V2C updates of three types of algorithms at diverse SNR.

| \( E_b/N_0 (dB) \) | 1.4 | 1.6 | 1.8 | 2.0 |
|---------------------|-----|-----|-----|-----|
| Average iterations  |     |     |     |     |
| FA                  | 25.9| 16.8| 12.6| 10.2|
| R-SA                | 12.1| 8.0 | 6.3 | 5.3 |
| RPD                 | 6.5 | 5.7 | 4.8 | 3.2 |
| C2V saved           |     |     |     |     |
| With R-SA           | 61.5%| 52.8%| 42.9%| 21.1%|
| With FA             | 123.1%| 105.6%| 85.8%| 42.2%|
| V2C saved           |     |     |     |     |
| With R-SA           | 38.4%| 50.3%| 40.8%| 19.6%|
| With FA             | 116.5%| 100.0%| 81.7%| 39.2%|

### TABLE 3. Dynamic scheduling operations per iteration for proposed RDP and NW-RBP algorithms.

| Algorithm (criterion) | \( |\log_2 g| \) bits | \( |\log_2 g| \) bits | \( N \times g \) bits | \( N \times g \) bits |
|-----------------------|-----------------|-----------------|-----------------|-----------------|
| Proposed RDP(RP)      | 0               | \( |\log_2 g| \) bits | \( N \times g \) bits | \( |\log_2 g| \) bits |
| NW-RBP (residual)     | \( N \times g \) | \( Q \) bits | \( N \times g \) | \( Q \) bits |
The encoded bits are QPSK modulated and transmitted over the AWGN channel. In simulations, for nodes with the same reliability, a random sorting method is used to update the nodes message.

The frame error rate (FER) performances versus the number of iterations at $\text{Eb/N}_0 = 1.75\text{dB}$ are shown in Figure 1. The figure depicts that the proposed RPD converges faster than other algorithms during the whole iterations from 0 to 50. The FER performance versus $\text{Eb/N}_0$ is demonstrated in Figure 2. It can be observed that RPD outperforms other algorithms when the number of maximum iterations approaches 50. Furthermore, the same as greedy RBP strategy, RPD updates some messages multiple times before other messages are updated once. Therefore, after a small number of message updates (10 iterations), the convergence speed of RPD and RBP is splendidly better than others. However, as the number of iterations increases, the greedy phenomenon induced by RBP begins to limit the improvement of FER. In contrast to RBP, the proposed RPD strategy maintains a better FER performance than RBP when the number of iterations is larger. RPD demonstrates a slightly better performance than the less-greedy NW-RBP strategy. The main reason is that the sequence of variable node update based on reliability profile enables all variable nodes to contribute their intrinsic messages to the decoding process. Depending on those features, RPD reduces the probability of the formation of greedy groups. Therefore, compared to the existing residual based RBP and NW-RBP algorithms, RPD exhibits not only a better convergence speed but also a better convergence performance.

In Figure 3 and Figure 4, the length-1536 rate-5/6 LDPC code from the IEEE 802.16e standard (Wimax) is selected. The maximum iteration is set to 10 iterations. The NMSA [14] is used for check-node update. Normalization constant $\alpha = 0.75$. It can be seen that similar results can be obtained in Figure 3 and Figure 4.

V. SIMULATION RESULTS AND ANALYSIS

In order to compare the proposed RPD with the current algorithms in [5] and [6], the same LDPC codes in communication standards are selected.

Figure 1 and Figure 2 show the performance of the length-1944 rate-1/2 LDPC code selected from the IEEE 802.11n standard (WiFi). The maximum number of iterations is set to 50. The precise sum-product [2] is used for check-node computing.

The performances of FA, SA, RBP, NW-RBP, and the proposed RPD algorithm are demonstrated in this section.
The bit error rate (BER) performances versus the number of iterations at Eb/N0 = 3.8dB are shown in Figure 3. The figure depicts that the proposed RPD converges obviously faster than the other four decoding algorithms during the whole iterations from 0 to 10. The BER performance versus Eb/N0 is demonstrated in Figure 4. It shows the BER performance of the five decoding schemes after 10 iterations. In agreement with the results suggested in Figure 3, the relative performances of the five schemes remain the same over the range of signal-to-noise ratios. In addition, for rate-5/6 LDPC code, row weight (the number of variable nodes corresponding to an edge) is usually larger than column weight (the number of check nodes corresponding to an edge). RBP chooses the least reliable check node based on residual and sequentially updates the variable nodes corresponding to the chosen edge, while RPD selects the least reliable variable node based on RP and sequentially updates the check nodes corresponding to the chosen edge. As a result, RDP propagates more check-to-variable messages than RBP propagates variable-to-check messages during the first iteration times. Therefore the convergence performance of RDP surpasses RBP for a small number of iterations between 0 and 6.

Through IV.A, the analysis of RPD decoding performance has been discussed in detail. The main effect of reliability is to reduce the decoding complexity. It is not the main reason to improve the decoding performance, and the column weight only affects the calculation of reliability, so it has little impact on the convergence and BER performance. From the simulation results in figures 1-4, it can be seen that the RPD algorithm can obtain good performance under different column weights.

In summary, simulation results demonstrate that by maintaining the same performance as the NW-RBP, the complexity of the proposed RPD method is much lower than the state-of-the-art IDS layered decoding algorithm.

VI. RPD FOR PROTOGRAPH LDPC CODES

ProtoGraph LDPC [17] was first described by J. Thorpe of JPL laboratory in 2003. Given the extension rules, the performance of protograph LDPC codes is completely determined by the structure of protograph. Therefore, they have the property that the minimum distance is linear to the code length with low error floor performance [18]. This unique construction determines that the original modulus LDPC codes have very low encoding and decoding complexity [19]. If quasi cyclic extension is adopted, QC-LDPC can be obtained [20], which can further reduce the complexity of encoding and decoding. Further, through deleting the variable nodes or adding and extending the row and column of protograph, we can get rate-compatible LDPC codes [21]. In terms of BER performance, when the number of decoding iterations is sufficient, the protograph LDPC codes are significantly better than the traditional LDPC code [22]. Based on the above advantages, the protograph LDPC codes were adopted in the international CCSDS standards for next-generation deep-space communications [23].

However, for protograph LDPC codes, when the number of decoding iterations is limited, the BER performance will decline sharply. That is to say, the decoding convergence speed of protograph LDPC codes is slow. At present, [24] and [25] focus on the decoding algorithm of protograph LDPC codes. However, for these algorithms, the number of iterations to converge to the ideal performance is still large. To solve this problem, we apply the dynamic strategy column layered decoding algorithm proposed in this paper to decode the protograph LDPC code, where the CCSDS protocol LDPC (8192, 4096) code [23] is selected. The protograph of the code rate 1/2 AR4JA (accumulate-repeat-jagged-accumulate) LDPC code is shown in figure 5.

The adjacency matrix of the protograph, i.e base matrix $H_{base}$ is shown equation (10).

$$H_{base} = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 1 & 1 & 0 & 1 & 3 \\ 1 & 2 & 0 & 2 & 1 \end{bmatrix}$$

The BER performance of the CCSDS protograph LDPC (8192, 4096) code using Reliability Profile Based Dynamic Schedule Algorithm Decoding is given and compared with the current decoding algorithms.

Due to the long code length of QC-LDPC (8192, 4096) in the CCSDS standard, C language is adopted in this paper to simulate the performance of LDPC decoding with QPSK modulation. AWGN channel model is selected. In order to reduce the complexity, NMSA with a normalization factor of 0.75 is applied in the decoding algorithm of the three strategies. Among them, FA, SA, and RPD represent the flooding
decoding strategy, general serial strategy, and the proposed reliability profile-based strategy, respectively. Figure 6 shows the BER performance of FA, SA, and RPD decoding algorithms using 15 fixed iterations. Figure 7 shows the influence of the iterations of FA, SA, and RPD decoding algorithms on the BER performance when the Eb/N0 is 1.3 dB, 1.4 dB and 1.5 dB, respectively.

From these simulation results, the convergence speed of the proposed RPD decoding algorithm is faster than the LBP algorithm, and more than twice faster than the BP algorithm. Since, the RPD decoding algorithm can set a large number of nodes not to update under a high signal-to-noise ratio, it greatly reduces the complexity of decoding. It is very suitable for protograph LDPC codes with a slow convergence speed.

VII. CONCLUSION

In this paper, we propose a new dynamic schedule called RPD for LDPC codes decoding, based on the concept of RP, nodes message is divided into two categories: reliable and unreliable. By using as much message as possible from the latest current iteration instead of the previous iteration to update unreliable message multiple times to speed up the convergence process and increase the total number of updates of effective nodes message, so as to improve the BER performance. By not updating the reliable message, the number of ineffective nodes message updates is reduced, thereby reducing the decoding complexity. Besides, the more iterations, the fewer calculations are required for these reliable nodes. Moreover, based on reliability profile, RPD has an advantage over RBP for dynamic scheduling in that the former only needs integer operations, while the latter requires float operations. Therefore, the RPD algorithm not only accelerates the convergence speed of LDPC decoding but also improves the BER performance while reducing the complexity. The analysis and simulation results show that it can achieve a much better convergence BER performance compared to the RBP without sacrificing convergence speed. Furthermore, the simulation results show that the RPD algorithm can significantly improve the convergence speed of protograph LDPC codes. Therefore, by combining the low complexity of RP-based dynamic schedule with both the convergence speed in the waterfall region and the error rate in the error-floor region, the proposed RPD LDPC decoding algorithm will become a competitive candidate for wireless communication applications.

REFERENCES

[1] R. G. Gallager, “Low-density parity-check codes,” IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, Jan. 1962.
[2] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inf. Theory, vol. 45, no. 2, pp. 399–431, Mar. 1999.
[3] M. Mansour and N. Shanbhag, “High-throughput LDPC decoders,” IEEE Trans. Very Large Scale Integ. (VLSI) Syst., vol. 11, no. 6, pp. 976–996, Dec. 2003.
[4] A. I. V. Casado, M. Groot, and R. Vesel, “Overcoming ldpc trapping sets with informed scheduling,” in Proc. Inf. Theory Appl. Workshop, 2007, pp. 1–5.
[5] A. I. V. Casado, M. Groot, and R. D. Vesel, “LDPC decoders with informed dynamic scheduling,” IEEE Trans. Commun., vol. 58, no. 12, pp. 3470–3479, Dec. 2010.
[6] Y. Gong, X. Liu, W. Ye, and G. Han, “Effective informed dynamic scheduling for belief propagation decoding of LDPC codes,” IEEE Trans. Commun., vol. 59, no. 10, pp. 2683–2691, Oct. 2011.
[7] S. Kim, “Trapping set error correction through adaptive informed dynamic scheduling of LDPC codes,” IEEE Commun. Lett., vol. 16, no. 7, pp. 1103–1105, Jul. 2012.
[8] H. C. Lee, Y. L. Ueng, S. M. Yeh, and W. Y. Weng, “Two informed dynamic scheduling strategies for iterative LDPC decoders,” IEEE Trans. Commun., vol. 61, no. 3, pp. 886–896, Mar. 2013.
[9] R. Cui and X. Liu, “Dynamic decoding algorithms of LDPC codes based on simulated annealing,” in Proc. Int. Workshop High Mobility Wireless Commun., Nov. 2014, pp. 153–139.
[10] T. Xia, H.-C. Wu, and S. C.-H. Huang, “A novel fast LDPC decoder using APP-based dynamic scheduling scheme,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2014, pp. 1–6.
[11] X. Liu, Z. Zhou, R. Cui, and E. Liu, “Informed decoding algorithms of LDPC codes based on dynamic selection strategy,” IEEE Trans. Commun., vol. 64, no. 4, pp. 1357–1366, Apr. 2016.
[12] A. Aslam, Y. Guan, K. Cai, and G. Han, “Low-complexity belief-propagation decoding via dynamic silent-variable-node-free scheduling,” IEEE Commun. Lett., vol. 21, no. 1, pp. 28–31, Jan. 2017.
[13] C. Healy, Z. Shao, R. M. Oliveira, R. C. de Lame, and L. L. Mendes, “Knowledge-aided informed dynamic scheduling for LDPC decoding of short blocks,” IET Commun., vol. 12, no. 9, pp. 1094–1101, May 2018.
[14] H. Zhang and S. Chen, “Residual-decaying-based informed dynamic scheduling for belief-propagation decoding of LDPC codes,” *IEEE Access*, vol. 7, pp. 23656–23666, 2019.

[15] B. Wang, Y. Zha, and J. Kang, “Two effective scheduling schemes for layered belief propagation of 5G LDPC codes,” *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1683–1686, Aug. 2020.

[16] W. Ryan and S. Lin, *Channel Codes: Classical Modern*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[17] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *IPN Prog. Rep.*, vol. 42, no. 154, pp. 42–154, 2003.

[18] B. Karimi and A. H. Banihashemi, “Construction of irregular protograph-based qc-ldp codes with low error floor,” *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 3–18, Jan. 2020.

[19] C. Tang, M. Jiang, C. Zhao, and H. Shen, “Design of protograph-based LDPC codes with limited decoding complexity,” *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2570–2573, Dec. 2017.

[20] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, “Quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths,” *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3758–3777, Jun. 2019.

[21] X. Zhong, K. Cai, P. Chen, and Z. Mei, “Rate-compatible protograph-based Hadamard codes,” *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 4438–4451, Jun. 2018.

[22] A. Dehghan and A. H. Banihashemi, “On the Tanner graph cycle distribution of random LDPC, random protograph-based LDPC, and random quasi-cyclic LDPC code ensembles,” *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 3758–3777, Jun. 2019.

[23] *Low Density Parity Check Codes for Use in Near-Earth and Deep Space Applications*, Clark County School District, North Las Vegas, BC, USA, 2007.

[24] F. Steiner, G. Bocherer, and G. Liva, “Protograph-based LDPC code design for bit-metric decoding,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1–6.

[25] P. W. Zhang, F. C. Lau, and C.-W. Sham, “Layered decoding for protograph-based low-density parity-check Hadamard codes,” *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1776–1780, Jun. 2021.

**TIANJIAO XIE** (Member, IEEE) received the M.S. degree in communications and information systems from Xiadian University, China, in 2008, and the Ph.D. degree in communications and information systems from Northwestern Polytechnical University, China, in 2020. Since 2008, she has been with the China Academy of Space Technology, Xi’an, China, where she is currently a Professor. Her research interests include error correcting codes and satellite networks.

**ZHONGFENG WANG** (Fellow, IEEE) received the B.E. and M.S. degrees from Tsinghua University, Beijing, China, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, in August 2000.

He recently joined Nanjing University as a Distinguished Professor after serving Broadcom Corporation as a leading VLSI architect for nearly nine years. From 2003 to 2007, he was an Assistant Professor with the School of EECS, Oregon State University, Corvallis. Even earlier, he was working with National Semiconductor Corporation. He is a World-Recognized Expert on VLSI for Forward Error Correction Codes. He has published over 100 technical papers, edited one book (“VLSI”) and filed tens of U.S. patent applications and disclosures. In the current record, since 2007, he has been having five papers ranked among top 20 most downloaded manuscripts in IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. During his tenure at Broadcom, he has contributed significantly on 10Gbps and beyond high-speed networking products. Additionally, he has made critical contributions in defining FEC coding schemes for 100Gbps and 400Gbps Ethernet standards. So far, his technical proposals have been adopted by many networking standard specs. His current research interests include VLSI for high-speed signal processing systems.

Dr. Wang was the recipient of the IEEE Circuits and Systems (CAS) Society VLSI Transactions Best Paper Award in 2007. In 2013, he served in the Best Paper Award selection committee for the IEEE Circuits and System Society. He has also served as a technical program committee member, the session chair, the track chair, and a review committee member for many IEEE and ACM conferences. Since 2004, he has served as an Associate Editor for the IEEE TRANSACTION ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS (TCAS-I), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS (TCAS-II), and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS for many terms.

---

**RUIJIA YUAN** (Member, IEEE) received the M.S. degree in software engineering from Xiadian University, in 2008, and the Ph.D. degree in communications and information systems from Xiadian University, China, in 2012. Since 2012, he has been with the China Academy of Space Technology, Xi’an, China, where he is currently a Senior Engineer. His research interests include error correcting codes and modern in satellite communications.

**ZHENJUN WANG** (Senior Member, IEEE) received the B.E. and M.S. degrees from Tsinghua University, Beijing, China, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, in August 2000.

He recently joined Nanjing University as a Distinguished Professor after serving Broadcom Corporation as a leading VLSI architect for nearly nine years. From 2003 to 2007, he was an Assistant Professor with the School of EECS, Oregon State University, Corvallis. Even earlier, he was working with National Semiconductor Corporation. He is a World-Recognized Expert on VLSI for Forward Error Correction Codes. He has published over 100 technical papers, edited one book (“VLSI”) and filed tens of U.S. patent applications and disclosures. In the current record, since 2007, he has been having five papers ranked among top 20 most downloaded manuscripts in IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. During his tenure at Broadcom, he has contributed significantly on 10Gbps and beyond high-speed networking products. Additionally, he has made critical contributions in defining FEC coding schemes for 100Gbps and 400Gbps Ethernet standards. So far, his technical proposals have been adopted by many networking standard specs. His current research interests include VLSI for high-speed signal processing systems.

Dr. Wang was the recipient of the IEEE Circuits and Systems (CAS) Society VLSI Transactions Best Paper Award in 2007. In 2013, he served in the Best Paper Award selection committee for the IEEE Circuits and System Society. He has also served as a technical program committee member, the session chair, the track chair, and a review committee member for many IEEE and ACM conferences. Since 2004, he has served as an Associate Editor for the IEEE TRANSACTION ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS (TCAS-I), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS (TCAS-II), and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS for many terms.