IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 13, 2021, accepted September 27, 2021, date of publication October 19, 2021, date of current version November 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3121587

Review on 5G NR LDPC Code: Recommendations for DTTB System

FENGSHUANG LI[1,2], CHAO ZHANG[1,2], (Senior Member, IEEE), KEWU PENG[1,2], (Senior Member, IEEE), ALEKSEI E. KRYLOV[3], ALEKSANDR A. KATYUSHNYJ[3], ANDREY V. RASHICH[3], DMITRY A. TKACHENKO[3], (Senior Member, IEEE), SERGEY B. MAKAROV[3], AND JIAN SONG[1,2], (Fellow, IEEE)

[1]Tsinghua National Laboratory of Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Peng Cheng Laboratory, Shenzhen 518055, China
[3]Higher School of Applied Physics and Space Technologies, Peter the Great St. Petersburg Polytechnic University, 195251 St. Petersburg, Russia

Corresponding author: Chao Zhang (z_c@tsinghua.edu.cn)

This work was supported in part by the Ministry of Science and Higher Education of the Russian Federation as part of the World-Class Research Center Program: Advanced Digital Technologies under Contract 075-15-2020-934 (dated November 17, 2020), in part by the Peng Cheng Laboratory under Grant PCL2021A10, and in part by Tsinghua University–Yunnan Mobile Digital TV Company Ltd. Joint Research Center (JCICBN).

ABSTRACT The freeze of the 5th generation new radio (5G NR) Release 16 indicates that 5G development has stepped into a new stage. The application of a dedicated low-density parity-check (LDPC) code for channel coding is an important technical advance that distinguishes 5G NR from the 4th generation (4G) long-term evolution (LTE) and LTE-advanced. Although LDPC codes have been used in many different systems, the newly developed LDPC code in 5G NR integrates many cutting-edge technologies to provide better performance and attractive features. Thus, it can be a good reference for channel coding in other evolving systems headed by digital terrestrial television broadcasting (DTTB). In emerging applications, the DTTB system needs to carry information with higher density, while meeting the high requirements for real-time, coverage, and bit error rate of broadcasting. To provide a reference for DTTB channel coding that improves its performance and supports new services, a review of 5G NR LDPC code implementation is carried out from three aspects: code analysis and design, decoding algorithm, and decoder architecture. We thoroughly evaluate each solution and highlight some candidates for existing implementations or directions for further development of the DTTB system.

INDEX TERMS Low-density parity-check (LDPC) code, digital terrestrial television broadcasting (DTTB), 5th generation new radio (5G NR), code design, decoding algorithm, decoder architecture.

I. INTRODUCTION

As a new generation standard for mobile communication, the 5th generation (5G) should be designed to meet the demands of multiple application scenarios, including enhanced mobile broadband (eMBB), ultra-reliable and low latency communications (URLLC), and massive machine type communications (mMTC) [1]. On the technical level, it means a significant improvement in data rate, latency, compatibility, and many other key performance indicators. Specified by the 3rd Generation Partnership Project (3GPP), 5G New Radio (NR) is a standard developed for 5G air interfaces to meet the above technical requirements. In contrast to the Turbo code used in 4G long-term evolution (LTE) and LTE-advanced (LTE-A), low-density parity-check (LDPC) code is introduced in 5G NR as the channel coding scheme for the data channel [2], which is one of the landmark technical achievements.

After decades of research since it was first proposed in [3], the LDPC code has been widely used in many wireless communication scenarios, such as deep space communication [4], wireless local area networks (LANs) [5], and digital terrestrial television broadcasting (DTTB) [6]–[8]. As a broadcasting system, the DTTB system has special technical requirements. For instance, area coverage is a characteristic criterion in the DTTB system, and a lower bit error rate is required in the physical layer owing to the relatively less upper layer control. With the continuous expansion

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Li.

of demand for service quality and variety, higher spectrum efficiency and throughput are also desired. To improve performance, DTTB standards have been constantly evolving in recent years. Following the digital video broadcasting – terrestrial 2 (DVB-T2) standard, the American Advanced Television System Committee (ATSC) officially promulgated ATSC 3.0 in 2017 [6], and Chinese digital television terrestrial multimedia broadcasting-advanced (DTMB-A) was accepted as the second-generation DTTB standard by ITU in 2019 [8].

The newly developed 5G NR LDPC code integrates both existing and emerging technologies, leading to better performance. Thus, utilizing its characteristics and implementation methods to enhance the coding scheme of the evolving DTTB system is a subject worthy of investigation. For simplification, the 5G NR LDPC code and LDPC codes in the existing DTTB standards are represented by 5G-LDPC and DTTB-LDPC in the following, respectively.

To provide a reference, we focus on not only the specific 5G-LDPC but also its complete implementation process, including code analysis and design, encoding/decoding algorithm, and encoder/decoder architecture. It should be pointed out that owing to structural designs such as quasi-cyclic (QC), raptor-like (RL), and irregular-repeat-accumulate (IRA) in 5G-LDPC and DTTB-LDPC, encoding is relatively easy to implement [9]–[11]. On the other hand, the decoding part often determines the performance of the entire transmission system and still has a lot of room for research. Therefore, we mainly focus on the decoding algorithm and decoder implementation.

The three aforementioned parts are not independent. Some constraints among them are given in [12], but it focuses on the decoder architecture and lacks discussion of the other two parts. Based on [12], a more detailed description is provided in Fig. 1.

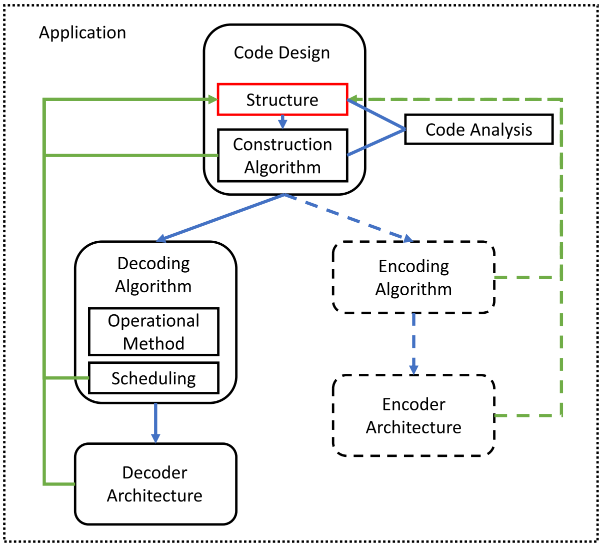


FIGURE 1. **Different parts and their relationships in the whole implementation process. This paper mainly focuses on the parts labeled by full line.**

Code analysis and design are always the primary parts of the process. Analytical tools are utilized to evaluate the average asymptotic thresholds and potential error floors for different code ensembles under the constraints of the parity-check matrix (PCM) with structural features and parameters. We define such constraints as ''structure'' which is marked in red in Fig. 1. After determining the structure, a construction algorithm is executed to find a specific code with good performance and desirable features in the ensemble.

In terms of decoding algorithms, most existing methods can be classified as message passing (MP) algorithms. In the Tanner graph, each row of the PCM corresponds to a check node (CN), each column corresponds to a variable node (VN), and each nonzero element in the matrix corresponds to an edge. In hardware implementation, VN and CN refer to two types of operation units, named VNU and CNU, respectively, which process incoming messages and pass the results along the edges to other units. Thus, the decoding algorithm is divided into the operational method (the method for node operations) and the scheduling. The former focuses on the operations in two types of units and message representation, while the latter focuses on the order of these operations.

Generally, the design flow corresponds to the blue arrows in Fig. 1, and the structure design should be carried out at the very beginning to provide a foundation for the following parts. Considering the major goal of having good performance, irregular codes involving some typical structures, such as RL and IRA, have been widely used in previous works, whose performances are validated by multi-edge-type density evolution (MET-DE) [13]. As shown by the green arrows in Fig. 1, the structure design should also consider other implementation-related requirements. Some structural designs, e.g., QC and protograph-based design [14], are preferred for narrowing the search space in code construction, facilitating code description, simplifying scheduling, and facilitating parallelism in both the encoder and decoder.

Moreover, the application scenarios also impel the structure design. The above structures guarantee good performance and high decoding throughput requirements in both 5G NR and DTTB applications. 5G-LDPC has QC and RL features, while ATSC 3.0, as an example of DTTB standards, also uses the QC-LDPC code which has the RL or IRA structure for different code rates. In addition, to adapt to various 5G application scenarios and make it competitive with 4G LTE Turbo code, the higher flexibility of multi-length and multi-rate (multi-mode) should be supported by 5G-LDPC. In contrast, the DTTB system adopts only limited predetermined modes chosen from many available mode options, which implies a relatively low demand for flexibility. However, considering the application of layered division multiplexing [15], the compatibility of code length required by the control channel [16], and so on, the guarantee of flexibility is still important to the DTTB system.

It can be concluded that MET, QC, and multi-mode support are applied to both 5G-LDPC and DTTB-LDPC. Thus, we review the existing solutions and noteworthy recent works on 5G-LDPC implementation considering these characteristics. Some candidates are identified and recommended for the DTTB system. Sections II, III, and IV focus on code analysis and design, decoding algorithm, and decoder architecture, respectively. Finally, Section V concludes the paper.

II. CODE ANALYSIS AND DESIGN

Different from the structure design discussed above, this section talks about the process of further narrowing down the code ensemble under the certain structure (MET, QC, multi-mode) until a specific code (PCM) is obtained, which corresponds to the "construction algorithm" in Fig. 1. For QC-LDPC codes, a two-step design for the base matrix (base graph) and lifting matrix (lifting factor) is utilized.

Threshold performance is the primary criterion for code design and is guaranteed by the base graph design, especially for moderate or long code lengths. Owing to the structural design in demand, the traditional searching algorithm can be utilized for a given structure with a limited search space. Analytical tools, e.g., MET-DE, are used to evaluate the threshold performance for each searching stage.

Extrinsic information transfer (EXIT) [17] analysis can also help evaluate threshold performance by tracking the variation of mutual information with the advantage of visualization via EXIT curve matching. Based on the MET view of irregular codes, protograph-based EXIT (PEXIT) is proposed [18], which can be used for code construction but cannot be visualized. A recent study [19] simplifies the classification of edges originating from the RL structure of 5G-LDPC, in which VNs and CNs are both divided into two categories. In this way, the EXIT analysis results can be displayed in a 3-dimensional diagram called a 3D-EXIT chart. This method has relatively low complexity, and the graphical results provide a new perspective for algorithm scheduling.

Error floor performance is another concern in code design, and the existence of trapping sets is the primary cause of the undesirable error floor. A review of related works and a search algorithm for elementary trapping set in irregular code are given in [20]. Alternatively, the existence of small girths is a more intuitive indicator closely related to the trapping set [21]. The optimization of small girths can also improve the error floor performance, leading to the proposal of progressive edge growth (PEG) algorithm [22] and approximated cycle extrinsic message degree (ACE) algorithm [23]. For QC codes, cyclic PEG (cPEG) [24] is proposed to jointly design the base graph and lifting factors.

These methods can be directly utilized in single-length and single-rate code designing, which are suitable for existing DTTB standards with a relatively small number of modes. As for 5G-LDPC, it has to meet the requirement of higher flexibility, which leads to an update of the structure and the corresponding construction method. 5G-LDPC utilizes a nested structure similar to the accumulate-repeat-4-jagged-accumulate (AR4JA) code [4] and protograph-based raptor-like (PBRL) code [25] but has a two-dimensional extension on both CNs and VNs to obtain finer granularity. A review of the nested design is given in [26], with a newly proposed progressive matrix growth (PMG) algorithm for two-dimensional construction of the base graph. Note that as one of the few algorithms for nested design, PMG uses average threshold performance as the criterion in every step of extension instead of greedy searching. Therefore, it can guarantee the average performance among different modes and has a larger search space. The flexibility of the lifting matrix size is another source of multi-length. 5G-LDPC adopts modulo-lifting [27], which provides the lifting matrix size with length-scalability and simplifies the code description.

The above methods can be applied to existing or even future codes with the same characteristics as 5G-LDPC, including QC, RL, and nested structures. For 5G-LDPC itself, which has already been standardized, we can use the above indicators to further evaluate its performance and point out potential development directions.

5G-LDPC has been proved to outperform LTE Turbo codes in threshold [28], yet the error floor also exists in some modes, as pointed out in [29]. Because there are many modes with similar parameters, it is suggested to avoid these underperformed modes in practical use.

Different from [29], we tested modes with smaller code lengths. We define the lifting matrix size as $z \times z$, the number of information VNs in the base graph is k, and the number of CNs in the base graph is m. We took Base Graph 2 with $k = 6, 8$, $m = 5, 6, 8, 10$, and $z = 13, 26, 52$. We carried out ACE analysis and found in modes with $k = 6, m = 5, z = 13$ and $k = 6, m = 8, z = 13$, there were girth-4s with a small ACE value, which suggested a potential undesired error floor. Thus, cPEG and ACE algorithms were utilized to re-design the lifting factors of the base graph with $k = 8$, $m = 10$, $z = 52$, which gave a newly designed base matrix with a row number of 10, column number of 18, and all of the lifting factors were not larger than 52. For easy comparison, only part of Base Graph 2 with $k = 8$, $m = 10$ is considered. Only the first 12 columns of both the newly designed base graph and original Base Graph 2 are displayed in Fig. 2 (a) and (b), respectively, since the last 6 columns are the same in both designs with the form $\left[\mathbf{0}_{6\times 4}\ \mathbf{I}_{6\times 6}\right]^T$.

Simulations of the original and new code families were conducted using the binary-input additive white Gaussian noise (AWGN) channel. The results are shown in Fig. 3. It can be observed that in the mode with $k = 6$, $m = 5$, $z = 13$ (code rate $R = 2/3$), the new code family performs better than the original one. In the other examined modes, the two codes have similar performance, which indicates the compatibility of the new one. It is proved that in the process of designing the 5G-LDPC, the optimization of the error floor with multi-mode compatibility is still a challenge and lacks effective methods. 5G-LDPC still has room for improvement.

Although existing DTTB-LDPC does not use a nested design, it is recommended to introduce such a design in the

144	20	177	166	0	0	197	0	1	1	0	0
19	0	0	28	4	103	186	18	0	1	1	0
127	164	0	48	184	0	0	0	2	0	1	1
0	37	49	0	19	112	204	4	1	0	0	1
44	28	0	0	0	0	0	0	0	118	0	0
137	50	0	0	0	37	0	133	0	63	0	0
8	0	0	0	0	35	0	199	0	13	0	0
0	164	0	0	0	79	0	144	0	108	0	59
102	178	0	0	0	0	0	0	0	0	23	0
0	187	0	0	0	0	0	0	206	82	0	0

(a)

1	1	1	1	0	0	1	0	1	1	0	0
2	0	0	13	1	1	8	1	0	1	1	0
3	33	0	36	13	0	0	0	45	0	1	1
0	34	48	0	31	30	44	40	15	0	0	1
1	48	0	0	0	0	0	0	0	19	0	0
1	23	0	0	0	24	0	6	0	5	0	0
1	0	0	0	0	33	0	30	0	16	0	0
0	1	0	0	0	6	0	22	0	51	0	14
1	16	0	0	0	0	0	0	0	0	49	0
0	1	0	0	0	0	0	0	45	40	0	0

(b)

FIGURE 2. The first 12 columns of the examined part of Base Graph 2 (a) and the new-designed base graph (b).

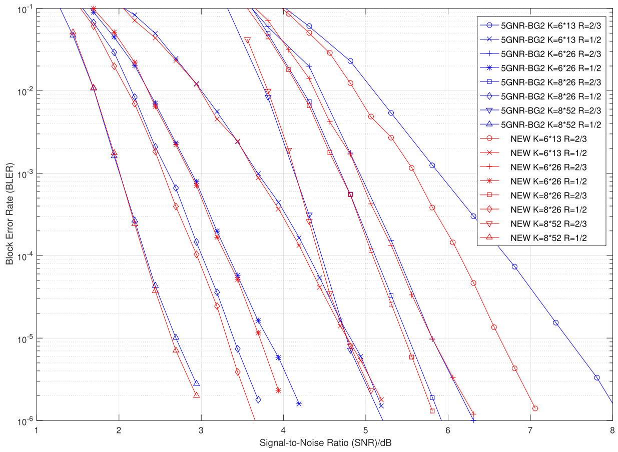


FIGURE 3. Simulation results.

future evolution standard to improve flexibility and compatibility, as pioneered in [30]. Compared to 5G NR, DTTB does not require a large number of modes but calls for better error floor performance. It also has a longer code length to ensure a lower threshold, which implies higher code construction complexity. Therefore, developing an efficient method with consideration of nested design while maintaining a good threshold and error-floor performance is still an open problem for both 5G NR and DTTB systems.

III. DECODING ALGORITHM

The sum-product algorithm (SPA) [31], also called the belief-propagation (BP) algorithm [32], is a near-optimal iterative decoding algorithm. In SPA, log-likelihood-ratio (LLR) messages are passed between VNUs and CNUs and processed by them. We define the index i for CN/CNU corresponding to the i-th row of the PCM, and the index j for VN/VNU corresponding to the j-th column. The indices of CNs that are adjacent to the VN_j form the set $M(j)$ and the indices of VNs that are adjacent to the CN_i form the set $N(i)$. The operations in VNU and CNU can be written as

$$Q_{i,j} = r_j + \sum_{i' \in M(j) \setminus i} L_{i',j} \tag{1}$$

$$L_{i,j} = 2 \tanh^{-1} \left(\prod_{j' \in N(i) \setminus j} \tanh \left(\frac{Q_{i,j'}}{2} \right) \right) \tag{2}$$

where $L_{i,j}$ is the message passed from CNU_i to VNU_j, $Q_{i,j}$ is the message passed from VNU_j to CNU_i, and r_j is the a priori message of the j-th bit from the upper module (e.g., constellation demapping) to VNU_j. $M(j)\setminus i$ denotes the set $M(j)$ excluding the element i, and $N(i)\setminus j$ is similar.

Obviously, tanh and $\tanh^{-1}$ in (2) lead to high implementation complexity. Many modified algorithms have been proposed to approximate and simplify computations. In addition, data dependence exists between VNs and CNs, which means that the processing order affects the convergence speed. Thus, we discuss two aspects of the decoding algorithm: the operational method in a single unit and the scheduling among multiple units, and then recommend decoding schemes for the DTTB system.

A. OPERATIONAL METHOD

The original operation of (2) can be implemented by looking up dedicated look-up tables (LUTs) multiple times. For further simplification, the min-sum algorithm (MSA) [33] is proposed as a suboptimal iterative algorithm, in which the operation in the CNU is

$$L_{i,j} = \prod_{j' \in N(i) \setminus j} \operatorname{sgn}\left(Q_{i,j'}\right) \cdot \min_{j' \in N(i) \setminus j} \left|Q_{i,j'}\right| \tag{3}$$

and the operation in the VNU is the same as in (1). Give the superscript SPA or MSA to $Q_{i,j}$ and $L_{i,j}$ to distinguish the messages of these two algorithms. Note that operations (1) and (3) in MSA are independent of the channel estimation error [34], leading to good robustness. However, this approximation inevitably causes performance degradation. Some improved methods have been proposed by introducing an acceptable increase in complexity. We classify them into the following categories based on different ways to get an improvement:

a) MSA-Based Approximation. A correction can be applied to $L_{i,j}^{MSA}$ to provide a numerical approximation to $L_{i,j}^{SPA}$. The simplest correction is attained by multiplying a normalization factor or subtracting an offset factor, which leads to the well-known normalized-MSA (NMSA) or offset-MSA (OMSA) [35], respectively. Note that the former preserves the independence between the algorithm and the channel estimation in MSA, indicating better robustness [34].

TABLE 1. Summary of characteristics in some noteworthy modified algorithms. "-" for not mentioned.

	[49]	[34]	[50]	[51]	[52]	[53]	[54]	[55]	[56]	[57]
View (a/c/d)	a	a	a	a/d	a/c	a	a	a	a/c	a
MET-based (Y/N)	Y	N	N	Y	N	N	Y	Y	Y	Y
Adaptive (Y/N)	Y	N	Y	N	Y	Y	Y	N	N	N
Parameter Obtaining (T/S)	S/T	-	S	-	-	-	S	T	S	-
Scheduling (F/L)	F	F	F	F/L	F	L	L	F	F	L
Standard	DVB-T2	ATSC 3.0	5G NR	5G NR	CCSDS / IEEE 802.3an	802.11n/ac	5G NR	ATSC 3.0	5G NR	5G NR

b) Optimal Post-Processing. It is pointed out in [36] that $L_{i,j}^{MSA}$ is not an LLR and mismatches to (1), which is oriented to LLR inputs, and it is the source of performance degradation. Thus, an optimal post-processing function is proposed to turn $L_{i,j}^{MSA}$ into the LLR form. The function has two other variables besides $L_{i,j}^{MSA}$ itself, which are CN_i's degree d_i^c and its a priori average mutual information I. To avoid the high complexity of implementing the function directly, a linear function dependent only on $L_{i,j}^{MSA}$ is used to fit the original function in [36]. If the fitting function is $L_{i,j}^{pp} = \alpha L_{i,j}^{MSA}$, NMSA is obtained. Similarly, if the fitting function is $L_{i,j}^{pp} = L_{i,j}^{MSA} - \beta$, OMSA is obtained. Although the implementation in b) is the same as a), it has a different point of view and reveals that the design of correction factors requires additional consideration of d_i^c and I, so it is necessary to list it separately.

c) Reliability-Based Correction. It is suggested in [37] and [38] that a stronger correction should be applied to more unreliable messages in NMSA or OMSA. In [37], two correction factors are employed, and the failure of the parity check in the current row is considered as the criterion for unreliability, giving adaptive-NMSA or adaptive-OMSA. In [38], it performs in a more extreme way. It sets unreliable $Q_{i,j}^{MSA}$ results to zero and proposes self-corrected MSA (SC-MSA). The criterion is that the sign of $Q_{i,j}^{MSA}$ with the same index changes in two adjacent iterations. It is proved that this erasure process can make the distributions of the messages approach a Gaussian distribution, which guarantees its performance improvement and provides another interpretation of SC-MSA.

d) SPA-Based Simplification. Some algorithms give a numerical approximation of the SPA directly without considering the MSA. They reserve the operation in (2) to some extent but simplify it by discarding unimportant inputs [39] or reducing the output diversity [40].

Most of the existing decoding algorithms, except SPA and MSA, are derived from the above views, and some notable recent ones are summarized in Table 1. Since the implementation of b) is mainly NMSA or OMSA, b) is regarded as a subset of a) and is not listed in Table 1.

Because of the application of irregular codes and the emergence of MET-based analysis tools, improvements based on MET are conducted on the original algorithms, indicating different parameters or operations for different types of CNUs or VNUs. In NMSA and OMSA, it means the introduction of multiple correction factors, whose necessity is proved in [36] from the optimal post-processing view. In [41], the classification is simplified based on row and column weights directly, whereas in [42], rows with different weights are merged for further simplification.

It has been pointed out in [36] that I is also a variable of the optimal post-processing function and is changing during the decoding process. Thus, adaptive correction factors are preferred to enhance the performance of NMSA or OMSA. Since I is hard to attain, some substitute criteria are utilized to guide the adaptation of factors. An intuitive one is the index of the iterations. In [43], the value of the sub-minimal input of a CNU is used as the criterion, and in [54], it is the failed parity-check equation proportion. Note that the algorithms with view c) are naturally adaptive.

To summarize, Table 1 is extended to show whether the algorithms are MET-based and adaptive.

For algorithms that use predetermined parameters, selecting the best parameters is also important in the implementation. As a typical case, the design of correction factors in NMSA or OMSA has been discussed in different works. In the early literature, such as [36], theoretical analysis and numerical computation are used to get the factors. However, it becomes difficult when MET-based and adaptive views are introduced, so some search methods have been applied. To evaluate each searching stage, theoretical methods using code analysis tools and simulation methods using the Monte Carlo method are both applicable. The former has a smaller complexity and potential universality, whereas the latter tends to have better performance for a specific code. To illustrate this difference, Table 1 is extended with the classification of parameter obtaining methods, which are divided into data-based simulation methods (S) and analysis-based theoretical methods (T).

Since the goal is always to balance the performance and complexity of the decoding algorithm, the above schemes need to be carefully selected according to the specific code and application scenario.

B. SCHEDULING

The original iterative decoding algorithms, such as SPA and MSA, use *flooding-scheduling*, i.e., in every iteration, all VNUs and then all CNUs run once. This processing order can be shuffled and *layered-scheduling* is proposed in [44] (also called turbo decoding message passing, TDMP in [45]) inspired by the decoding of concatenated codes. It splits PCM into several submatrices by row grouping, called layers, in which part of the CNUs and VNUs run alternatively. The simulation results show that the average number of iterations can be reduced by 50%. If the maximum column weight in each layer is 1, parallelism can be easily performed in the decoder [44]. Thus, for QC-LDPC codes, the most direct way to define layers is to regard the submatrix corresponding to a row of the base graph as a layer. Similarly, PCM is split by column grouping in [46], which achieves a similar improvement. The key is to enable the newly calculated messages to participate in the following operations instantly. Based on this point, a more complex scheduling scheme is proposed [47], resulting in faster convergence. However, considering the complexity of implementation and the occupancy of storage resources, layered-scheduling based on row grouping is used more widely than other non-flooding methods.

Different operation units have different abilities to process messages, so the processing order of layers in layered-scheduling also influences the convergence speed. A layer ordering strategy is proposed in [48] based on the thought of processing and passing highly reliable messages preferentially. In [19], theoretical analysis and design of the layer processing order are provided using the proposed 3D-EXIT chart. Moreover, in hardware implementation, the data dependence between the VNUs and CNUs may cause conflicts in the pipeline design. Therefore, conflict avoidance should also be considered in the processing order designing, which is further discussed in Section IV.

Due to the change in schedule, the parameters or criteria that are determined under flooding-scheduling may require redesigning. The classification of scheduling has to be added to Table 1, where F means flooding and L means layered.

C. DECODING ALGORITHM SELECTION FOR 5G NR AND DTTB SYSTEMS

As for the decoding algorithm, the special consideration of nested design, which is the main difference between DTTB-LDPC and 5G-LDPC, is not involved. A comprehensive comparison can be made to determine the recommended schemes for the DTTB system.

As shown in Table 1, most of the algorithms are based on view a) even for 5G NR because of its advantages in terms of complexity and performance. Thus, NMSA/OMSA can be selected for existing or even future DTTB systems, but some improvements are needed. Based on the discussion on the structure, the typical MET characteristics in the codes of both systems should be noticed and the MET-based view is recommended. Further consideration is needed for the classification of nodes using the same parameter table, and the degree of nodes is still the most commonly used criterion. Because of the QC structure, layered-scheduling is preferred for faster convergence and utilizing parallelism in hardware, both of which improve the decoder throughput. Adaptivity can be introduced for further improvements. With respect to multi-mode support, when introducing MET-based or adaptive views, the parameters used in the algorithm should have universality to save storage resources, instead of training or calculating for every single mode.

Among the algorithms in Table 1, an enhanced NMSA with all the improvements listed above is proposed in [54], which is recommended for implementation. Note that it uses the failed parity-check equation proportion as a criterion for choosing adaptive correction factors, which has potential universality, although the factors are trained via Monte Carlo simulation. The layer processing order design proposed in [19] and [48] can be combined with this algorithm for higher throughput.

IV. DECODER ARCHITECTURE

The basic requirement of decoder design is to effectively and efficiently implement the target algorithm with good performance in various hardware indicators, such as throughput, resource occupation, and energy efficiency. Moreover, the decoder has its own concerns, including quantization, parallelization, and pipelining, which directly influence the system performance and should be well-designed.

A. FUNDAMENTAL IMPLEMENTATION

In the aforementioned algorithms, the core processing of messages is the operations in the CNUs and VNUs. Thus, the implementation of these two types of units is the basis of decoder design.

The operations in the CNU are different among the algorithms. Headed by MSA, many of them reduce the output diversity of the CNU compared to SPA, leading to a reduction in the amount of calculation and storage for intermediate results, and are preferred in decoder implementation. For VNU, the accumulation of $L_{i,j}$ in (1) is carried out in almost every algorithm. Modify (1) into the following for simplification:

$$P_j = r_j + \sum_{i' \in M(j)} L_{i',j} \tag{4}$$

$$Q_{i,j} = P_j - L_{i,j} \tag{5}$$

where P_j indicates the a posteriori message of the j-th bit. In addition, storage usage can be reduced if P_j is stored and $Q_{i,j}$ is calculated by (5) in time, instead of storing $Q_{i,j}$ directly.

Data quantization for CNU/VNU should also be optimized to achieve a trade-off between performance and resource usage. Variations in quantization can be introduced for different types of messages or along with the iteration progress [58], [59]. Note that some new algorithms have been developed for quantized messages directly [53], [57], [60], which provides a new view for algorithm design.

B. PARALLELIZATION

Parallelization is supported by the introduction of multiple operation units. Different scheduling strategies have different parallel architectures.

Consider a PCM with m rows and n columns. For flooding-scheduling, at most m CNUs and n VNUs can be used in parallel, resulting in *full-parallel* architecture [62]. Comparatively, the layered-scheduling, recommended in Section III, requires serial implementation between layers, and parallel

TABLE 2. The number of VNUs/CNUs and the processing times needed by all of the VNUs/CNUs in one iteration in different architectures.

	Full-parallel	Layered-parallel	Block-parallel	PLDA
Number of VNUs	$z_{max} \times n_{max}$	$z_{max} \times d^c_{max}$	z_{max}	n_{max}
Number of CNUs	$z_{max} \times m_{max}$	z_{max}	z_{max}	m_{max}
Processing times	1	m	E	z

calculations are applied only within a single layer, leading to *layered-parallel* or *row-parallel* [61], [62]. If each layer contains only 1 row, 1 CNU is required and is shared by all the layers. d^c_i VNUs are needed for the i-th layer (row) to support the parallel operations of VNs, so d^c_{max} VNUs should be implemented for all the layers, where d^c_{max} is the maximum among all d^c_i. The discussion above of the $m \times n$ PCM is valid for an $m \times n$ base graph. Here, d^c_i indicates the weight of the i-th row in the base graph. Thus, utilizing z times of the above units leads to full-parallel and layered-parallel architectures for a QC-LDPC code with lifting matrices of size $z \times z$. For a group of QC-LDPC codes with different lifting matrix sizes, z_{max} times are needed to support all codes, where z_{max} is the maximum z among them.

Implementations of both full-parallel and layered-parallel depend closely on the base graph and the size of the lifting matrix. If a code has parameters (m, n, d^c_{max}, or z_{max}) beyond the range specified by the decoder, it cannot work normally, which means a limitation of universality. On the contrary, if the parameters do not reach the bounds, the decoder wastes resources and has low energy efficiency. To solve this problem, parallelism can be further reduced. One solution is *block-parallel* [61], [62]. It is based on the layered-scheduling but carries out the processing of each layer in a serial manner, which is different from the layered-parallel. In the base graph view, for the i-th layer (row), d^c_i times of processing are required for the whole layer. Similarly, z_{max} degrees of parallelism should be supported by hardware implementation. This architecture decouples the hardware design from the base graph and is universally applicable to any QC-LDPC code constrained by a fixed z_{max}. All of the VNUs and CNUs should run E times to complete a decoding iteration, where E is the number of nonzero elements (edges) in the base graph. Another solution, called *parallel layered decoding architecture* (PLDA), is proposed in [63]. It is a variant of the layered-parallel and only targets QC-LDPC codes. It extracts one row from each layer defined in the layered-parallel to form a new layer. Thus, the PCM is split into z layers, each of which has m rows. In contrast to the layered-parallel, this makes rows in one layer have different weights, and the weight distribution is the same as that of the base graph. This architecture decouples the hardware design from the lifting matrix size, and all of the VNUs and CNUs should run z times in one iteration. It is compatible with codes with nested design but constrained by the largest base graph (mother code) in the nested structure whose size is $m_{max} \times n_{max}$, where m_{max} and n_{max} are the largest row number and column number of the code family, respectively [64]. It means that the universality of different base graphs is limited in PLDA.

Table. 2 gives a summary of the discussion above. Line 2/3 shows the number of VNUs/CNUs needed by each architecture, which is equal to the maximum supported degree of parallelism in VN/CN processing. It also indicates restrictions on the parameters of the code. Note that the specific implementation of VNU/CNU varies in different architectures, so the number of operation units is not equivalent to the resource occupancy. Line 4 shows the number of processing times of all of the VNUs/CNUs in one decoding iteration, indicating the unrestricted parameter supported by each architecture at the cost of decoding time.

C. MESSAGE PASSING NETWORK (MPN)

The MPN is designed to support the above parallel architectures by passing new messages at the same time. For full-parallel, CNUs and VNUs should be connected directly according to the PCM, resulting in a complicated fixed MPN, which also limits the application of this architecture. In terms of partially-parallel architectures, the connection changes in the serial process of each iteration. Thus, a simpler and reconfigurable MPN is needed. Furthermore, the structural design of the code can make the MPN structural [12]. Hence, for QC-LDPC codes, a two-level MPN design is adopted, in which the first level reflects the connection given by the base graph, and the second level implements circular shift based on the lifting factor. This makes a universal circular shift network with less resource occupancy and a shorter critical path becomes one of the key points in the MPN design.

For a group of codes with a given z_{max}, a circular shift network that can support the shift value p for z messages ($0 \leq p < z$, $1 \leq z \leq z_{max}$) is desired. Traditional multi-level barrel shifters (BSs) do not have such universality for different z. This is solved by adding an extra self-routing bit to every message and a look-up level in [65]. The Benes network, as a non-blocking switch network, is used for circular shifting in [66] with modifications. Because the original Benes network can support arbitrary switches, redundancy exists when it is used only for circular shifting. A dedicated circular shift network, QC-LDPC shift network (QSN), is proposed in [67]. It has the advantages of a shorter critical path, lower resource occupancy, and easy generation of control signals by gate circuits. In [68], a Banyan network with a bypass structure is used for the MPN, which is also an alternative with good performance. Recent works modify existing MPNs to support parallel circular shifting, which gives the decoder potential of parallel decoding when z is small, e.g., a Banyan-based shifter and a BS-based shifter (called extended barrel-shifter, EBS) are designed with this consideration in [69] and [70], respectively. Generally, there is no special restriction on z_{max} when designing MPN, but it should be noted that a dedicated designed z_{max} and optional z can be beneficial for the implementation of MPN. For example, the form $a \times 2^j$ ($a = \{2, 3, 5, 7, 9, 11, 13, 15\}$, $j = \{0, 1, 2, 3, 4, 5, 6, 7\}$, and $z = a \times 2^j$ in the range from 2 to 384) of z in 5G-LDPC makes it easier to design a Banyan network with $a \times a$ and 2×2 switches. Furthermore, when a is fixed, the degree of parallelism of the power of 2 can be easily achieved by the Banyan-based shifter proposed in [69]. This proves that the design of the MPN and lifting factor should be considered jointly and shows a further relationship between the decoder architecture and code design.

D. PIPELINING

The critical path composed of VNU, CNU, and MPN can be split into stages and pipelined for higher throughput. The main obstacle is the frequent pipeline conflict caused by data dependence.

When using flooding-scheduling, the entire process of a single iteration should be split into stages for pipelining. However, there must be data dependence between adjacent iterations, so it is impossible to make different stages process different iterations. Thus, multi-frame decoding is proposed [71], in which different stages process different codewords. It improves the throughput in multiples but has high latency and requires huge hardware resources.

In terms of layered-scheduling, the process of each layer can be divided into stages. In the base graph view, if two CNs are connected to at least one same VN, there is a risk of pipeline conflict. Similarly, multi-frame decoding is feasible naturally [72]. In addition, single-frame decoding can also be pipelined in layered-scheduling with the help of techniques that alleviate or avoid conflict, and we classify them into the following categories:

a) Stalling. It is the most direct way to totally avoid conflict but lowers the throughput because of the stall cycles.

b) PCM-Based Methods. In [73], every block of the lifting matrix is split into several submatrices without changing the code, leading to more layers but less conflict between layers. Direct avoidance of conflict between layers can also be conducted in the code design, e.g., the quasi-orthogonal structure of 5G-LDPC. Note that these methods are not perfect and may fail when the number of stages increases.

c) Decoding-Algorithm-Based Methods. Both scheduling and operations can be modified to resolve conflicts. Considering the former, a change in the layer processing order can reduce the risk of conflict. Moreover, when utilizing the block-parallel structure, the processing order of VNs provides a new degree of freedom for scheduling optimization, making the method more powerful. For the latter, a modified operation for the VNU is proposed in [74] to solve the parallel conflict, and this method is used in [75] to solve pipeline conflict in the name of residue-based. This is equivalent to conducting flooding-scheduling partially and is called hybrid scheduling in [76]. This method removes conflict completely but lowers the efficiency of message passing because partial flooding-scheduling is introduced.

d) Hardware-Based Methods. Traditional hardware solutions for pipeline conflict can be utilized directly, e.g., by introducing a forwarding module or bypass structure [77].

These four categories of methods are independent and can be combined. Since the goal is to avoid conflict completely, inserting sufficient stall cycles in a) or changing the operation in c) is necessary. Hence, the degradation of throughput caused by stall cycles or partial flooding-scheduling should be carefully evaluated.

E. DECODER ARCHITECTURE SELECTION FOR 5G NR AND DTTB SYSTEMS

A good summary of many LDPC decoders with a field-programmable gate array (FPGA) implementation is provided in [78]. However, it does not review pipelining and is unable to cover some recent implementations of 5G-LDPC. As a supplement to [78], we list several 5G-LDPC decoders in Table 3 and summarize them according to the points of the design mentioned above.

The decoder can meet the demands of universality and throughput by introducing the full-parallel architecture with sufficient operation units and pipelining for multi-frame. However, when considering resource occupation, efficiency, latency, and some other application-related criteria, partially-parallel architectures based on layered-scheduling are preferred, as shown in Table 3.

If we focus on the high flexibility of the lifting matrix size, PLDA has little waste of hardware resources for 5G-LDPC. On the contrary, if we focus on the high flexibility of the base graph, block-parallel is preferred. Since the number of edges in the base graphs of 5G-LDPC (BG1: $E = 316$, BG2: $E = 197$) has a similar magnitude as its $z_{max} = 384$ and both have a large range of variation, it is difficult to determine which architecture is better simply. However, it is worth noting that because of the parallel circular shift design, block-parallel will have higher throughput and less waste of resources even when $z < z_{max}$, which compensates for its key shortcoming compared to PLDA. If we further consider the demand for supporting two different base graphs and the potential for supporting new base graphs, block-parallel, which decouples from the base graph, is preferred. Relatively speaking, DTTB-LDPC has extremely low flexibility in parameter z (e.g., $z = 360$ in ATSC 3.0) and has different base graphs for different modes, so PLDA has no advantage, and block-parallel is recommended.

In terms of pipelining, the PLDA implementation in [80] is not pipelined. In fact, non-conflict pipelining can be introduced into PLDA using a strictly designed layer-splitting method [63], [64]. Block-parallel implementations in [79] and [76] both introduce pipelining. The latter adapts both scheduling and operation modification in c) and has been proved to have higher throughput than direct stalling, which makes it a better one. Compared to the 5G-LDPC code, DTTB-LDPC does not have a quasi-orthogonal structure,

TABLE 3. Recent implementations for 5G-LDPC and their features. "-" for not mentioned or with no special design.

	[57]	[76]	[79]	[80]	[81]
Parallel Architecture	layered-parallel	block-parallel	block-parallel	PLDA	layered-parallel
MPN	-	QSN	EBS	-	-
Pipelining (Y/N)	N	Y	Y	N	Y
Conflict Solution (a/b/c/d)	-	c	a	-	-

which means greater challenges in pipeline conflict avoidance. The implementation in [76] is worthy to be carried out and tested for the DTTB system. Additional structural constraints, such as quasi-orthogonal, can also be imposed for future code design.

V. CONCLUSION

This paper provided a review of the technical points in 5G-LDPC implementation based on recent notable works. Starting from the code structure, it discussed the code analysis and design, decoding algorithm, and decoder architecture within the framework for final implementation and application. The development roadmap was clarified, and different schemes were classified and compared. By comparing the characteristics of 5G-LDPC and DTTB-LDPC, this paper recommended some universal solutions with good performance for both 5G NR and DTTB systems and also proposed some directions for future development of DTTB code design. All key observations can be summarized as follows:

a) For code design, PMG or similar algorithms can be used for the base graph design of 5G-LDPC or codes with the same structure. Further development is needed for lifting factor design considering the average performance guarantee of different lifting matrix sizes. DTTB-LDPC can refer to 5G-LDPC with the aim of longer code length, better threshold performance, and lower error floor. Thus, it is necessary to improve the existing methods and pay more attention to the average error floor performance under different modes.

b) For decoding algorithm, a layered-scheduling, MET-based, and adaptive NMSA/OMSA provides a good balance between performance and complexity. It is suggested that indicators for choosing adaptive parameters should have potential universality, e.g., the failed parity-check equation proportion. Layer processing order design is also worthy of introduction.

c) For decoder architecture, recent block-parallel implementation toward 5G-LDPC has advantages in terms of universality and hardware efficiency. By introducing the optimized CNU/VNU/MPN design, pipelining, and conflict-handling mechanisms, higher throughput can be achieved. These techniques can be applied to the DTTB systems.

All of the recommended solutions for DTTB are based on 5G-LDPC with full consideration of its rate-compatibility and length-scalability features, making them applicable to other systems, e.g., wireless LANs and deep space communication. These solutions may also be used in future 3GPP Releases, e.g., for flexible and effective delivery of 5G broadcast and multicast services to mobile devices [82]–[84]. Moreover, the continuous improvement of these techniques can make them suitable for future codes with similar structures. New structural requirements can be put forward for the better use of existing techniques, which emphasizes constraints between the code structure and implementation.

ACKNOWLEDGMENT

The authors would like to acknowledge Dr. Eugene Popov, an Associate Professor of Peter the Great St. Petersburg Polytechnic University, for his valuable suggestions on this paper.

REFERENCES

[1] *Detailed Specifications of the Terrestrial Radio Interfaces of International Mobile Telecommunications-2020 (IMT-2020)*, document Rec. ITU-R M.2150-0, Geneva, Switzerland, Feb. 2021.

[2] *Technical Specification Group Radio Access Network; NR; Multiplexing and Channel Coding (Release 16)*, document 3GPP TS 38.212 v16.4.0, Dec. 2020.

[3] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[4] *Low Density Parity Check Codes for Use in Near-Earth Deep Space Applications*, document CCSDS 131.1-O-2, Washington, DC, USA, Sep. 2007.

[5] *Local and Metropolitan Area Networks–Specific Requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2020, Dec. 2020.

[6] *ATSC Standard: Physical Layer Protocol (A/322)*, document A/322:2017, Jun. 2017.

[7] *Digital Video Broadcasting (DVB); Frame Structure Channel Coding and Modulation for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2)*, document ETSI EN 302 755 v1.4.1, Jul. 2015.

[8] J. Song, C. Zhang, K. Peng, J. Wang, C. Pan, F. Yang, J. Wang, H. Yang, Y. Xue, Y. Zhang, and Z. Yang, "Key technologies and measurements for DTMB—A system," *IEEE Trans. Broadcast.*, vol. 65, no. 1, pp. 53–64, Mar. 2019.

[9] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[10] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[11] H. Jin, A. Khandekar, and R. J. McEliece, "Irregular repeat accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes*, Brest, France, Sep. 2000, pp. 1–8.

[12] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 976–996, Dec. 2003.

[13] T. Richardson, "Multi-edge type LDPC codes," in *Proc. Workshop Honoring Prof. Bob McEliece 60th Birthday*. Pasadena, CA, USA: California Institute of Technology, May 2002, pp. 24–25.

[14] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *JPL INP Prog. Rep.*, vol. 42, pp. 42–154, Aug. 2003.

[15] W. Li, Y. Wu, S. Lafleche, K. Salehian, L. Zhang, A. Florea, S.-I. Park, J.-Y. Lee, H.-M. Kim, N. Hur, C. Regueiro, J. Montalban, and P. Angueira, "Coverage study of ATSC 3.0 under strong co-channel interference environments," *IEEE Trans. Broadcast.*, vol. 65, no. 1, pp. 73–82, Mar. 2019.

[16] H. Jeong, K. Kim, S. Myung, J. Shin, J. Kim, S. Park, S. Kwon, Y. Shi, and S. Kim, "Flexible and robust transmission for physical layer signaling of ATSC 3.0," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 204–215, Mar. 2016.

[17] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[18] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Washington, DC, USA, Nov. 2007, pp. 3250–3254.

[19] S. Shao, Y. Zhang, R. G. Maunder, and L. Hanzo, "3D EXIT charts for analyzing the 5G 3GPP new radio LDPC decoder," *IEEE Access*, vol. 8, pp. 188797–188812, 2020.

[20] Y. Hashemi and A. H. Banihashemi, "Characterization of elementary trapping sets in irregular LDPC codes and the corresponding efficient exhaustive search algorithms," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3411–3430, May 2018.

[21] M. Karimi and A. H. Banihashemi, "Efficient algorithm for finding dominant trapping sets of LDPC codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6942–6958, Nov. 2012.

[22] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[23] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, Aug. 2004.

[24] Z. Li and B. V. K. V. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *Proc. Conf. Rec. 38th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, vol. 2, Nov. 2004, pp. 1990–1994.

[25] T. Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1522–1532, May 2015.

[26] Y. Zhang, K. Peng, Z. Chen, and J. Song, "Progressive matrix growth algorithm for constructing rate-compatible length-scalable raptor-like quasi-cyclic LDPC codes," *IEEE Trans. Broadcast.*, vol. 64, no. 4, pp. 816–829, Dec. 2018.

[27] S. Myung, K. Yang, and Y. Kim, "Lifting methods for quasi-cyclic LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 6, pp. 489–491, Jun. 2006.

[28] *R1-166388, LDPC Rate Compatible Design*, 3GPP document TSG RAN WG1 86, Qualcomm, Gothenburg, Sweden, Aug. 2016.

[29] *R1-1801468, Evaluation of Channel Coding Schemes*, 3GPP document TSG RAN WG1 92, Huawei, HiSilicon, Athens, Greece, Feb. 2018.

[30] S. I. Park, Y. Wu, H. M. Kim, N. Hur, and J. Kim, "Raptor-like rate compatible LDPC codes and their puncturing performance for the cloud transmission system," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 239–245, Jun. 2014.

[31] 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.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[33] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

[34] S. Myung, S.-I. Park, K.-J. Kim, J.-Y. Lee, S. Kwon, and J. Kim, "Offset and normalized min-sum algorithms for ATSC 3.0 LDPC decoder," *IEEE Trans. Broadcast.*, vol. 63, no. 4, pp. 734–739, Dec. 2017.

[35] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208–210, May 2002.

[36] G. Lechner and J. Sayer, "Improved sum-min decoding of LDPC codes," in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, Parma, Italy, Oct. 2004, pp. 1–4.

[37] X. Wu, Y. Song, M. Jiang, and C. Zhao, "Adaptive-normalized/offset min-sum algorithm," *IEEE Commun. Lett.*, vol. 14, no. 7, pp. 667–669, Jul. 2010.

[38] V. Savin, "Self-corrected min-sum decoding of LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Toronto, ON, Canada, Jul. 2008, pp. 146–150.

[39] F. Guilloud, E. Boutillon, and J.-L. Danger, "Lambda-min decoding algorithm of regular and irregular LDPC codes," in *Proc. Int. Symp. Topics Coding*, Brest, France, 2003, pp. 451–454.

[40] C. Jones, E. Valles, M. Smith, and J. Villasenor, "Approximate-MIN constraint node updating for LDPC code decoding," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Boston, MA, USA, Oct. 2003, pp. 157–162 vol. 1.

[41] J. Zhang, M. Fossorier, D. Gu, and J. Zhang, "Improved min-sum decoding of LDPC codes using 2-dimensional normalization," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, St. Louis, MO, USA, Nov. 2005, pp. 1187–1192.

[42] Y. Liu, K. Peng, C. Pan, and L. Fan, "Genie-aided adaptive normalized min-sum algorithm for LDPC decoding," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Dubrovnik, Croatia, Aug. 2015, pp. 221–225.

[43] R. Chen and Y.-M. Wang, "Modified normalized min-sum decoding of LDPC codes," in *Proc. 9th Int. Conf. Signal Process.*, Beijing, China, Oct. 2008, pp. 1811–1814.

[44] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop Signal Process. Syst. (SIPS)*, Austin, TX, USA, Oct. 2004, pp. 107–112.

[45] M. M. Mansour, "A turbo-decoding message-passing algorithm for sparse parity-check matrix codes," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4376–4392, Nov. 2006.

[46] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," in *Proc. Conf. Rec. 36th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, vol. 1, Nov. 2002, pp. 8–15.

[47] S. Usman, M. M. Mansour, and A. Chehab, "Interlaced column-row message-passing schedule for decoding LDPC codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[48] C.-Y. Liang, M.-R. Li, H.-C. Lee, H.-Y. Lee, and Y.-L. Ueng, "Hardware-friendly LDPC decoding scheduling for 5G HARQ applications," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 1418–1422.

[49] Y. Xu, L. Szczecinski, B. Rong, F. Labeau, D. He, Y. Wu, and W. Zhang, "Variable LLR scaling in min-sum decoding for irregular LDPC codes," *IEEE Trans. Broadcast.*, vol. 60, no. 4, pp. 606–613, Dec. 2014.

[50] X. Wu, M. Jiang, and C. Zhao, "Decoding optimization for 5G LDPC codes by machine learning," *IEEE Access*, vol. 6, pp. 50179–50186, 2018.

[51] K. Sun and M. Jiang, "A hybrid decoding algorithm for low-rate LDPC codes in 5G," in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Hangzhou, China, Oct. 2018, pp. 1–5.

[52] A. Kharin, I. Volkov, A. Ovinnikov, E. Likhobabin, and V. Vityazev, "Performance of self-corrected min-sum decoding algorithm for decoders with quantized input," in *Proc. 18th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, Mar. 2019, pp. 1–4.

[53] I. Tsatsaragkos and V. Paliouras, "A reconfigurable LDPC decoder optimized for 802.11n/ac applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 1, pp. 182–195, Jan. 2018.

[54] Z. Zhou, K. Peng, A. Krylov, A. Rashich, D. Tkachenko, F. Li, C. Zhang, and J. Song, "Enhanced adaptive normalized min-sum algorithm for layered scheduling of 5G-NR LDPC codes," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Paris, France, Oct. 2020, pp. 1–5.

[55] N. Gao, Y. Xu, D. He, S.-I. Park, H. Hong, H. Ju, C. Chen, and W. Zhang, "Min-sum algorithm using multi-edge-type normalized scheme for ATSC 3.0 LDPC decoders," *IEEE Trans. Broadcast.*, vol. 66, no. 3, pp. 729–736, Sep. 2020.

[56] P. Kang, Y. Xie, L. Yang, and J. Yuan, "Enhanced quasi-maximum likelihood decoding based on 2D modified min-sum algorithm for 5G LDPC codes," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6669–6682, Nov. 2020.

[57] H. Cui, F. Ghaffari, K. Le, D. Declercq, J. Lin, and Z. Wang, "Design of high-performance and area-efficient decoder for 5G LDPC codes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 879–891, Feb. 2021.

[58] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. J. Wainwright, "Design of LDPC decoders for improved low error rate performance: Quantization and algorithm choices," *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3258–3268, Nov. 2009.

[59] S. Kim, G. E. Sobelman, and H. Lee, "Adaptive quantization in min-sum based irregular LDPC decoder," in *Proc. IEEE Int. Symp. Circuits Syst.*, Seattle, WA, USA, May 2008, pp. 536–539.

[60] M. Stark, L. Wang, G. Bauch, and R. D. Wesel, "Decoding rate-compatible 5G-LDPC codes with coarse quantization using the information bottleneck method," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 646–660, 2020.

[61] *R1-1706177, QC-LDPC Code Throughput Comparison*, 3GPP document TSG RAN WG1 NR, Mediatek, Spokane, WA, USA, Apr. 2017.

[62] C. Roth, A. Cevrero, C. Studer, Y. Leblebici, and A. Burg, "Area, throughput, and energy-efficiency trade-offs in the VLSI implementation of LDPC decoders," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Rio de Janeiro, Brazil, May 2011, pp. 1772–1775.

[63] K. Zhang, X. Huang, and Z. Wang, "High-throughput layered decoder implementation for quasi-cyclic LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 985–994, Aug. 2009.

[64] K. Zhang, X. Huang, and Z. Wang, "A high-throughput LDPC decoder architecture with rate compatibility," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 4, pp. 839–847, Apr. 2011.

[65] C.-H. Liu, S.-W. Yen, C.-L. Chen, H.-C. Chang, C.-Y. Lee, Y.-S. Hsu, and S.-J. Jou, "An LDPC decoder chip based on self-routing network for IEEE 802.16e applications," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, Mar. 2008.

[66] D. Oh and K. K. Parhi, "Low-complexity switch network for reconfigurable LDPC decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 1, pp. 85–94, Jan. 2010.

[67] X. Chen, S. Lin, and V. Akella, "QSN—A simple circular-shift network for reconfigurable quasi-cyclic LDPC decoders," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 10, pp. 782–786, Oct. 2010.

[68] X. Peng, Z. Chen, X. Zhao, F. Maehara, and S. Goto, "High parallel variation banyan network based permutation network for reconfigurable LDPC decoder," in *Proc. 21st IEEE Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Rennes, France, Jul. 2010, pp. 233–238.

[69] Z. Zhong, Y. Huang, Z. Zhang, X. You, and C. Zhang, "A flexible and high parallel permutation network for 5G LDPC decoders," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 12, pp. 3018–3022, Dec. 2020.

[70] E. Boutillon and H. Harb, "Extended barrel-shifter for versatile QC-LDPC decoders," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 643–647, May 2020.

[71] M. Weiner, B. Nikolic, and Z. Zhang, "LDPC decoder architecture for high-data rate personal-area networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Rio de Janeiro, Brazil, May 2011, pp. 1784–1787.

[72] S. Kumawat, R. Shrestha, N. Daga, and R. Paily, "High-throughput LDPC-decoder architecture using efficient comparison techniques & dynamic multi-frame processing schedule," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 5, pp. 1421–1430, May 2015.

[73] C. Marchand, J.-B. Dore, L. Conde-Canencia, and E. Boutillon, "Conflict resolution for pipelined layered LDPC decoders," in *Proc. IEEE Workshop Signal Process. Syst.*, Tampere, Finland, Oct. 2009, pp. 220–225.

[74] M. Rovini, F. Rossi, P. Ciao, N. L'Insalata, and L. Fanucci, "Layered decoding of non-layered LDPC codes," in *Proc. 9th EUROMICRO Conf. Digit. Syst. Design (DSD)*, Dubrovnik, Croatia, 2006, pp. 537–544.

[75] O. Boncalo, G. Kolumban-Antal, A. Amaricai, V. Savin, and D. Declercq, "Layered LDPC decoders with efficient memory access scheduling and mapping and built-in support for pipeline hazards mitigation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 4, pp. 1643–1656, Apr. 2019.

[76] V. L. Petrović, M. M. Marković, D. M. El Mezeni, L. V. Saranovac, and A. RadoÅİević, "Flexible high throughput QC-LDPC decoder with perfect pipeline conflicts resolution and efficient hardware utilization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 5454–5467, Dec. 2020.

[77] X. Zhao, Z. Chen, X. Peng, D. Zhou, and S. Goto, "DVB-T2 LDPC decoder with perfect conflict resolution," in *Proc. Tech. Program VLSI Design, Autom. Test*, Hsinchu, Taiwan, Apr. 2012, pp. 1–4.

[78] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A survey of FPGA-based LDPC decoders," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1098–1122, 2nd Quart., 2016.

[79] J. Nadal and A. Baghdadi, "FPGA based design and prototyping of efficient 5G QC-LDPC channel decoding," in *Proc. Int. Workshop Rapid Syst. Prototyping (RSP)*, Hamburg, Germany, Sep. 2020, pp. 1–7.

[80] A. Katyushnyj, A. Krylov, A. Rashich, C. Zhang, and K. Peng, "FPGA implementation of LDPC decoder for 5G NR with parallel layered architecture and adaptive normalization," in *Proc. IEEE Int. Conf. Electr. Eng. Photon. (EExPolytech)*, St. Petersburg, Russia, Oct. 2020, pp. 34–37.

[81] A. Verma and R. Shrestha, "A new VLSI architecture of next-generation QC-LDPC decoder for 5G new-radio wireless-communication standard," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, Oct. 2020, pp. 1–5.

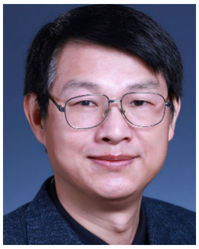
[82] J. J. Gimenez, J. L. Carcel, M. Fuentes, E. Garro, S. Elliott, D. Vargas, C. Menzel, and D. Gomez-Barquero, "5G new radio for terrestrial broadcast: A forward-looking approach for NR-MBMS," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 356–368, Jun. 2019.

[83] M. A. Taga, "Worldwide 5G broadcast/multicast trials come online," *SMPTE Motion Imag. J.*, vol. 129, no. 8, pp. 136–137, Sep. 2020.

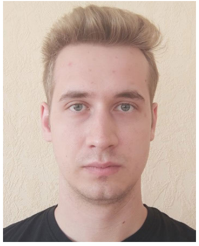
[84] Q. Zeng, "The broadcasting in the dawn of 5th generation wireless networks," presented at the IEEE Int. Symp. Broadband Multimedia Syst. Broadcast (BMSB), Paris, France, Oct. 2020. [Online]. Available: http://bmsb2020.isep.fr/pdf/keynote2.pdf

FENGSHUANG LI was born in China, in 1997. He received the B.E. degree from Tsinghua University, in 2019, where he is currently pursuing the M.E. degree with the Department of Electronic Engineering. His research interests include high performance module design and implementation of channel coding and modulation in wireless communication systems.

CHAO ZHANG (Senior Member, IEEE) received the B.S. and Ph.D. degrees from Beihang University, in 2001 and 2008, respectively. From 2008 to 2010, he was a Postdoctoral Fellow with the Department of Electronic Engineering, Tsinghua University, Beijing, China, where he is currently an Associate Professor with the Department of Electronic Engineering. He has authored over 50 journals and conference papers. He holds over 20 Chinese patents. His research interests include wireless and visible light communications. He received the IEEE Scott Helt Memorial Award (Best Paper Award in IEEE Transactions on Broadcasting), in 2016.



KEWU PENG (Senior Member, IEEE) was born in Hefei, China. He received the B.E. degree in electronics engineering from the Hefei University of Technology, in 1993, the M.E. degree in electronics engineering from Tsinghua University, in 1996, and the Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, in 2003. He was a Researcher and a Lecturer with the Department of Electronics Engineering, Tsinghua University, from August 1996 to August 1999. Since January 2005, he has been with the Digital Television Research Center, Tsinghua University, as a Research Staff, an Assistant Researcher, in 2006, and an Associate Professor, in 2009. His recent contributions focus on SC-LDPC codes and NOMA. He has published more than 90 journals and conference papers. He holds more than 80 China patents. His main research interests include mobile/wireless communications, digital terrestrial/television broadcasting, and embedded image/video transmission.



ALEKSEI E. KRYLOV was born in Russia, in 1995. He received the B.S. and M.S. degrees in telecommunication technologies and communication systems from the Peter the Great St. Petersburg Polytechnic University (SPbPU), Russia, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree. His research interests include digital signal processing, forward error correction coding, and corresponding hardware implementations.

ALEKSANDR A. KATYUSHNYJ was born in St. Petersburg, Russia, in 1997. He received the B.S. and M.S. degrees in telecommunication technologies and communication systems from the Peter the Great St. Petersburg Polytechnic University (SPbPU), Russia, in 2019 and 2021, respectively. His research interests include digital communications, FPGA programming, and 5G.

ANDREY V. RASHICH was born in USSR, in 1983. He received the B.S., M.S., and Ph.D. degrees in communication systems engineering from the Peter the Great St. Petersburg Polytechnic University (SPbPU), Russia, in 2004, 2006, and 2009, respectively. He is currently an Associate Professor with the Higher School of Applied Physics and Space Technologies, SPbPU. His current research interests include signal processing for wireless systems, demodulation, and FEC decoding algorithm architectures development and implementation in FPGAs/ASICs.

DMITRY A. TKACHENKO (Senior Member, IEEE) received the Dipl.Eng. degree in radio physics and electronics from SPbPU (currently Peter the Great St. Petersburg Polytechnic University), Russia, in 1986, and the Ph.D. degree in radio engineering and TV systems and devices, in 1993. He is currently an Associate Professor at the Higher School of Applied Physics and Space Technologies, SPbPU. His current research interests include systems for digital TV and radio broadcasting and their convergence with 5G and beyond systems as well as satellite communication systems.

SERGEY B. MAKAROV was born in Leningrad, Russia, in 1948. He received the Ph.D. degree in technical science from the Leningrad Polytechnic Institute, Russia, in 1977, and the Doctor of Science degree, in 1991. He is currently a Professor at the Higher School of Applied Physics and Space Technologies, SPbPU, Russia. He is the author of more than 300 publications in the field of wireless communications. His current research interests include spectrally efficient signaling, optimal signals for wireless communications, meteor-burst and UWB communication networks, and the Industrial Internet of Things.

JIAN SONG (Fellow, IEEE) received the B.Eng. and Ph.D. degrees in electrical engineering from Tsinghua University, in 1990 and 1995, respectively. As a Postdoctoral Researcher, he was with The Chinese University of Hong Kong and the University of Waterloo, Canada, in 1996 and 1997, respectively. Then, he worked for industry in USA for seven years. He joined the Faculty Team, Tsinghua University, as a Professor, in 2005, where he is currently the Director of the DTV Technology Research and Development Center. He has published over 300 peer-reviewed journals and conference papers. He has coauthored several books and book chapters in the aforementioned areas. He holds two U.S. and over 80 Chinese patents. His research interests include digital television terrestrial broadcasting, wireless communication, power line communication, and visible light communications. He is a fellow of IET, CIC, and CIE. He was a recipient of the IEEE Scott Helt Memorial Award, in 2015.