A Very Fast Joint Detection for Polar-Coded SCMA

DONGJUN NA [1], **WON-GI SEO**[2], **HONGSUK SHIM**[3], **KWANWOONG RYU** [4],
AND KWONHUE CHOI [1], (Senior Member, IEEE)

[1]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan-si 38541, South Korea
[2]Nextwill, Daejeon 34155, South Korea
[3]Hanwha Systems, Seongnam-si 13524, South Korea
[4]Electronics and Telecommunication Research Institute (ETRI), Daejeon 34129, South Korea

Corresponding author: Kwonhue Choi (gonew@yu.ac.kr)

ABSTRACT We propose a very fast-convergence joint iterative detection and decoding (JIDD) scheme for channel-coded sparse code multiple access (SCMA). In the conventional JIDD, all users' channel decoding iterations are performed in parallel after all users' variable nodes in the SCMA factor graph are updated. The proposed JIDD scheme, however, slices all channel decoding iterations into per-user channel decoding, and inserts them deeply into the factor graph. By doing this, message enhancement by channel decoding immediately propagates to the connected users' messages in the factor graph, even within one message passing algorithm iteration, while maintaining the same total computational complexity per JIDD iteration. Numerical results confirm that in various link scenarios, the proposed scheme requires only two or three iterations for BER convergence, while the conventional JIDD scheme requires more than six iterations. In downlink scenarios, the proposed scheme achieves an even faster convergence rate. Moreover, in a uplink scenario with perfect power control, the converged BER of the proposed scheme is quite a bit lower than the conventional scheme, and the proposed scheme requires only two iterations to get the same BER level as the conventional scheme. Consequently, thanks to fast convergence, the proposed scheme dramatically reduces the overall computational complexity for achieving BER convergence. In addition, the fast convergence rate compensates for the multi-user detection latency issue, which is inherent in sequential algorithms, and the issue is further overcome by employing the group-wise sequential version of the proposed scheme.

INDEX TERMS Decoding, iterative detection, joint detection, polar code, SCMA.

I. INTRODUCTION

Sparse code multiple access (SCMA) is a non-orthogonal multiple access scheme that has attracted much attention in recent years. SCMA substantially improves spectral efficiency in linear sparse sequences through multi-dimensional shaping gain of codebooks [1]. In addition, a message passing algorithm (MPA) enables SCMA to achieve near-optimal multi-user detection (MUD) [2], [3]. However, there is a major challenge to be resolved in MPA-based detectors for more practical usages: poor MUD performance in low signal-to-noise ratio (SNR) regions due to high overloading factors [4]. Therefore, in order to overcome this issue, combining SCMA with channel coding is essential.

As for improving MUD performance by combining SCMA with channel coding, iterative user detection and decoding schemes for channel-coded SCMA systems have been studied [5]–[8]. In [5], the authors proposed an iterative detection and decoding (IDD) scheme for Turbo-coded SCMA systems. The IDD scheme, in which each outer iteration consists of two separate inner loop iterations for the SCMA detector and turbo decoder, can achieve considerable MUD performance improvement. However, this scheme requires a lot of iterations, causing high computational complexity. To overcome this problem, a more tight combination of SCMA detection and channel decoding (termed joint IDD (JIDD) throughout this paper to distinguish it from IDD [5]) was recently proposed [6]–[8]. This structure utilizes the output extrinsic messages of the channel decoder in the middle of one cycle of the MPA process, and thus, MUD performance is significantly

improved, and BER convergence is accelerated, compared to the IDD scheme in [5].

In particular, the JIDD scheme for Polar coded SCMA systems shows an outstanding convergence rate, compared to rates with other channel-coded SCMA schemes such as turbo-coded or LDPC-coded SCMA systems [8]. This is due to the fast convergence characteristics of Polar code [9]. In addition, Polar code is drawing attention from its lower complexity compared to other channel codes [10], [11], and it has already been adopted for downlink control channels in 5G new radio (NR) interfaces. Although the JIDD scheme improves MUD performance from a Polar-coded SCMA system, the number of required iterations is still high (more than five) [8]. Therefore, it is very timely to study and improve Polar-coded SCMA systems, and in this paper, we mainly consider the Polar-coded SCMA system.

Meanwhile, in the context of fast-convergence SCMA without channel coding, a sequential MPA called a *shuffled* MPA (S-MPA) was proposed where the two processes for message update (i.e., updating messages from function node (FN) to variable node (VN) and updating messages from VN to FN) are shuffled [12], [14]. This scheme makes updated messages immediately propagate to other message updates connected in a factor graph, even within one MPA iteration, and thus, fast convergence is achieved. However, because S-MPA-based SCMA does not improve the converged BER level, but improves the convergence rate, channel coding needs to be combined for acceptable MUD in low SNR regions. The problem is that just combining S-MPA-based SCMA with channel coding or iterative channel decoding fails to conserve the benefits frome the S-MPA, since the decoding iteration is performed exclusively with the S-MPA iteration, and thus, there is no synergy between the S-MPA and channel decoding. Moreover, the aforementioned JIDD cannot be applied to the S-MPA. This is because the JIDD scheme performs channel decoding iterations in between two one-directional parallel message updates in the factor graph (i.e., FN-to-VN-messages-only updates and VN-to-FN-messages-only updates), and this does not allow VN/FN-shuffled message updates during one joint iteration of MPA and channel decoding.

The main goal of this paper is to develop a new Polar-coded SCMA system achieving fast convergence as well as improved MUD performance. We propose a new JIDD, termed *user-sequential* JIDD (us-JIDD), which constructively combines the sequential message update idea with JIDD to maximize the synergy between sequential message update and JIDD in terms of fast convergence capabilities. To this end, we first need to fundamentally solve the aforementioned issue, i.e., the structural conflict between JIDD and sequential message updates. The proposed us-JIDD scheme resolves this issue by slicing all channel decoding iterations into per-user channel decoding and inserting them into the SCMA factor graph as deep as the message update unit. Specifically, each user's channel decoding iteration is

performed right after the corresponding user's message is updated in each MPA iteration. In addition, due to this tightly shuffled channel decoding and MPA, careful consideration should be taken in combining a message updated from the MPA process and the enhanced message from the channel decoder.

The contributions of this paper are summarized as follows:

- We propose user-sequential JIDD, which constructively combines the conventional JIDD with the idea of sequential message updates in the MPA. So far, research on applying a user-sequential manner to channel-coded SCMA has not been reported. The proposed scheme can be extended to SCMA with other iterative channel codes, such as LDPC and turbo code.

- The proposed scheme brings a dramatic performance improvement to channel-coded SCMA. In various link scenarios, the proposed scheme requires only two or three iterations for BER convergence, and achieves an even faster convergence rate in downlink scenarios. On the other hand, the conventional JIDD-based SCMA scheme requires more than six iterations. Moreover, the proposed scheme ameliorates the multi-user detection latency issue, which is typically inherent in sequential algorithms.

- In light of its low complexity from fast convergence without the latency issue, the proposed us-JIDD scheme is expected to be used as an essential component of next-generation SCMA, rather than an option.

The remainder of the paper is structured as follows. In Sec. II, the channel-coded SCMA system model is presented. In Sec. III, we give an overview of the conventional JIDD scheme for the Polar-coded SCMA. In Sec. IV, we detail the user-sequential JIDD scheme for Polar-coded SCMA. In addition, we provide an analysis of the latency in JIDD versus the proposed us-JIDD scheme, and we provide a latency reduction method. In Sec. V, we provide an analysis of operational complexities of JIDD and the proposed us-JIDD scheme. Sec. VI presents simulated channel-coded BER as a function of the number of joint MPA iterations, and compares convergence speeds from the JIDD scheme with the proposed us-JIDD. In addition, we compare computational complexity and latency for JIDD versus the proposed us-JIDD. Finally, in Section VII, concluding remarks are provided.

The simulation code for the JIDD-based SCMA and the proposed us-JIDD-based SCMA can be downloaded from [13]. The numerical results in this paper can be regenerated by using this code.

II. SYSTEM MODEL

Consider a Polar-coded SCMA system where K users are multiplexed with a data bit sequence length of m. In the transmitter, the data bit sequence of user k, denoted by $\mathbf{u}^{(k)} = \left\{ u_1^{(k)}, u_2^{(k)}, \ldots, u_m^{(k)} \right\}$, $1 \leq k \leq K$, is encoded by the Polar encoder. The Polar-coded bit sequence is denoted by
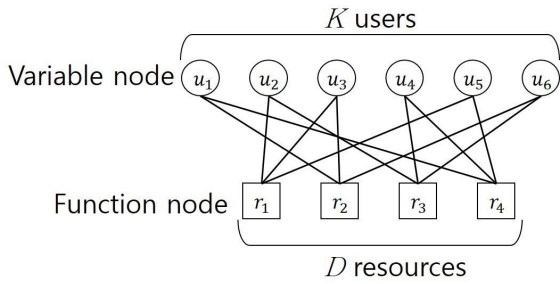
FIGURE 1. Factor graph of SCMA with six variable nodes and four function nodes.

FIGURE 2. The structures of JIDD and us-JIDD schemes.

$\mathbf{c}^{(k)} = \left\{ c_1^{(k)}, c_2^{(k)}, \ldots, c_N^{(k)} \right\}$, where N is the length of the coded bit sequence. Then, the Polar-coded bit sequence is interleaved and mapped to complex domain codewords by the SCMA encoder. A random interleaver is used between the SCMA encoder and the Polar encoder. The interleaved sequence is denoted by $\mathbf{b}^{(k)} = \left\{ b_1^{(k)}, b_2^{(k)}, \ldots, b_N^{(k)} \right\}$. Assume that the constellation size of SCMA is M, and every $Q(= \log_2 M)$ bits of $\mathbf{b}^{(k)}$ are grouped and mapped to a D-dimensional complex codeword. Thus, the total number of SCMA codewords per frame L is N/Q, and the l-th SCMA codeword of user k is denoted by $\mathbf{x}_l^{(k)} = \left\{ x_{l,1}^{(k)}, x_{l,2}^{(k)}, \ldots, x_{l,D}^{(k)} \right\}^T$.

In the SCMA system, K users' codewords are multiplexed and spread over D orthogonal resources and transmitted. SCMA codewords contain zero and nonzero elements, indicating the use of orthogonal resources. For more than K active users in a cell, the users are grouped with K members in each group, and the separate D resources are exclusively allocated to each of the multiple user groups. We consider a widely used SCMA codeword set where $K = 6$, $D = 4$, and $M = 4$ (a diagram is shown in [4], [6]–[8], [12], [15], [16], [18] and [20]). Fig. 1 shows the corresponding factor graph with variable nodes (VNs) for 6 ($= K$) users and function nodes (FNs) for 4 ($= D$) resources, where u_k denotes the k-th VN for user k, and r_d denotes the d-th FN for the d-th resource.
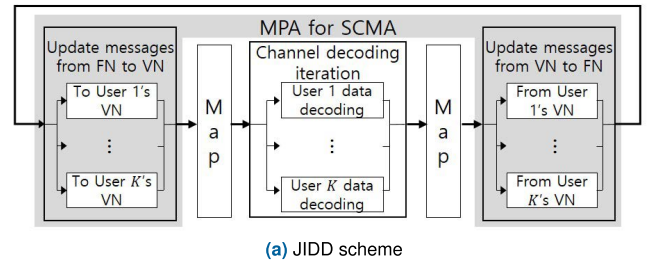
The received signal of the l-th SCMA codeword, denoted by $\mathbf{y}_l = \left\{ y_{l,1}, y_{l,2}, \ldots, y_{l,D} \right\}^T$, is expressed as

$$\mathbf{y}_l = \sum_{k=1}^{K} \mathbf{H}_l^{(k)} \mathbf{x}_l^{(k)} + \mathbf{z}_l, \ 1 \leq l \leq L, \tag{1}$$
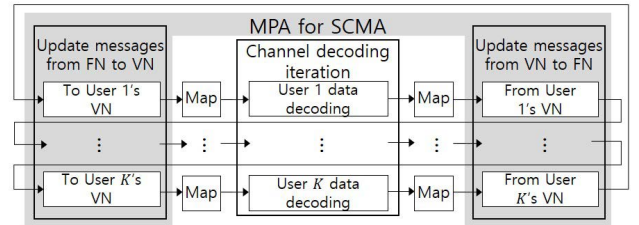
where $\mathbf{H}_l^{(k)}$ is the channel matrix between the base station and user k at the l-th SCMA codeword, and is expressed as follows [8]:

$$\mathbf{H}_l^{(k)} = diag \left(h_{l,1}^{(k)}, h_{l,2}^{(k)}, \ldots, h_{l,D}^{(k)} \right), \tag{2}$$

where $h_{l,d}^{(k)}$ represents the channel gain for the d-th resource. The term $\mathbf{z}_l = \left\{ z_{l,1}, z_{l,2}, \ldots, z_{l,D} \right\}^T$ is the AWGN vector, and $\mathbf{z}_l \sim \mathcal{CN} \left(0, \sigma^2 \mathbf{I} \right)$, in which σ^2 is the variance of $z_{l,d}$. We consider a power-normalized SCMA codebook such that

the total signal power per resource, i.e., $\sum_{k=1}^{K} \left\| \mathbf{x}_l^{(k)} \right\|^2 / D$, is equal to unity [26], and power normalized channel gain $E \left[\left| h_{l,d}^{(k)} \right|^2 \right] = 1$. This signal model corresponds to the case when the data bit energy, E_b, is $\frac{1}{C_r} \times \frac{D}{KQ}$ where C_r denotes the Polar code rate, $\frac{D}{KQ}$ is equal to the channel bit energy, and noise density $N_0/2 = \sigma^2/2$. Consequently, for a simulation with a given $\frac{E_b}{N_0}$, we set $\sigma^2 = \left(\frac{E_b}{N_0} \right)^{-1} \frac{D}{C_r KQ}$.

III. THE CONVENTIONAL JIDD SCHEME

In this section, we introduce the core concept of the conventional JIDD scheme, which is essential to understanding our scheme in Section IV. Fig. 2a shows the structure of the conventional JIDD, which tightly combines SCMA detection and channel decoding [6]–[8]. Hereafter, this conventional JIDD is simply called JIDD. First, each MPA iteration is split into two one-directional parallel message updates in the factor graph, i.e., FN-to-VN-messages-only updates and VN-to-FN-messages-only updates. The FN-to-VN-messages-only updates correspond to upward message passing in Fig. 1, and the VN-to-FN-messages-only updates correspond to downward message passing. The key idea in JIDD is to perform channel decoding iterations in between these two one-directional message updates, as shown in Fig. 2a, where the left side grey box indicates FN-to-VN-messages-only updates, and the right side grey box indicates VN-to-FN-messages-only updates. By doing this, the messages enhanced by channel decoding can be used in the middle of each MPA iteration.

A. JOINT MPA-BASED DETECTION IN JIDD

The joint factor graph including the MPA and channel decoding can be illustrated as shown in Fig. 3. In each joint
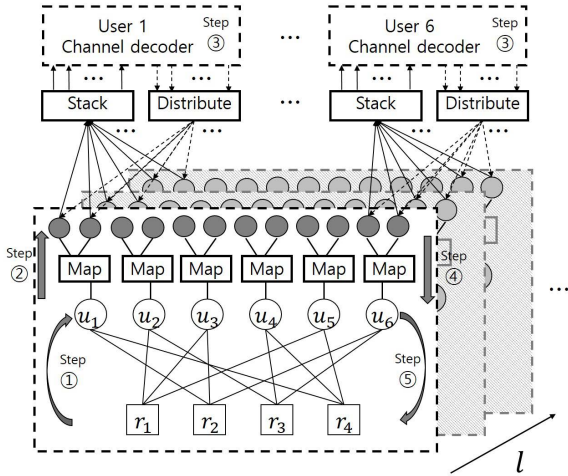
FIGURE 3. Channel decoding-embedded joint factor graph for JIDD.

iteration, steps ① to ⑤, described as follows, are performed sequentially and repeat until the BER converges.

1) STEP ① (FN-TO-VN-MESSAGES-ONLY UPDATE)

Let ξ_k and ζ_d denote the set of FNs connected to VN u_k and the set of VNs connected to FN r_d, respectively. In addition, let $\{\xi_k \setminus d\}$ and $\{\zeta_d \setminus k\}$ denote ξ_k with r_d omitted, and ζ_d with u_k omitted, respectively.

In the JIDD algorithm, the FN-to-VN-messages-only updates follow the conventional MPA [8]. Specifically, the message from FN r_d to VN u_k, for complex codeword $\mathbf{x}_l^{(k)}$ at the τ-th iteration denoted by $I_{r_d \to u_k}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right)$, is calculated as follows [12]:

$$
\begin{aligned}
&I_{r_d \to u_k}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right) \\
&= \sum_{\mathbf{x}_l^{(p)}, p \in \{\zeta_d \setminus k\}} \left\{ \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{1}{2\sigma^2}\left\| \mathbf{y}_l - \sum_{m \in \zeta_d} h_{l,d}^{(m)} x_{l,d}^{(m)} \right\|^2\right) \right. \\
&\left. \times \prod_{p \in \{\zeta_d \setminus k\}} I_{u_p \to r_d}^{(\tau-1)}\left(\mathbf{x}_l^{(p)}\right) \right\},
\end{aligned}
\tag{3}
$$

where $I_{u_p \to r_d}^{(\tau-1)}\left(\mathbf{x}_l^{(p)}\right)$ denotes the message from VN u_p to FN r_d for complex codeword $\mathbf{x}_l^{(p)}$ updated at the $(\tau - 1)$-th iteration, and the initial value of $I_{u_p \to r_d}^{(\tau-1)}\left(\mathbf{x}_l^{(p)}\right)$, i.e., $I_{u_p \to r_d}^{(0)}\left(\mathbf{x}_l^{(p)}\right)$ is set to $1/M$.

2) STEP ② TO STEP ④ (MAPPING AND CHANNEL DECODING PROCESS)

To perform a channel decoding iteration (between FN-to-VN-messages-only updates and VN-to-FN-messages-only updates, as shown in Fig. 2a), FN-to-VN-messages $I_{r_p \to u_k}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right), p \in \xi_k$, should be converted into the likelihood

of codeword $\mathbf{x}_l^{(k)}$, denoted by $P_e^C\left(\mathbf{x}_l^{(k)}\right)$, as follows [8]:

$$
P_e^C\left(\mathbf{x}_l^{(k)}\right) = \prod_{p \in \xi_k} I_{r_p \to u_k}\left(\mathbf{x}_l^{(k)}\right).
\tag{4}
$$

Then, the likelihood for each hypothesis of data bit $b_{(l-1)Q+q}^{(k)}$, denoted by $P_e^B\left(b_{(l-1)Q+q}^{(k)} = 0\right)$ and $P_e^B\left(b_{(l-1)Q+q}^{(k)} = 1\right)$, is calculated by the SCMA mapping function as follows [8]:

$$
P_e^B\left(b_{(l-1)Q+q}^{(k)} = 0\right) = \sum_{\mathbf{x} \in \left\{\mathbf{x}_l^{(k)} | b_{(l-1)Q+q}^{(k)} = 0\right\}} P_e^C\left(\mathbf{x}\right),
\tag{5}
$$

$$
P_e^B\left(b_{(l-1)Q+q}^{(k)} = 1\right) = \sum_{\mathbf{x} \in \left\{\mathbf{x}_l^{(k)} | b_{(l-1)Q+q}^{(k)} = 1\right\}} P_e^C\left(\mathbf{x}\right),
\tag{6}
$$

where $1 \leq q \leq Q$, and $\left\{\mathbf{x}_l^{(k)} \mid b_{(l-1)Q+q}^{(k)} = 0\right\}$ means a set of $\mathbf{x}_l^{(k)}$s satisfying the condition that the q-th bit of the Q data bits mapped to $\mathbf{x}_l^{(k)}$ is 0 [8]. The meaning of $\left\{\mathbf{x}_l^{(k)} \mid b_{(l-1)Q+q}^{(k)} = 1\right\}$ is similar.

As for the soft-input/soft-output (SISO) decoding algorithm required for the Polar decoding process in the JIDD scheme, the soft cancellation (SCAN) algorithm is used [8]. The SCAN algorithm converges quickly and requires less memory, compared to other SISO Polar decoding algorithms [8]. The input messages of the Polar decoders are in the form of a log-likelihood ratio (LLR), and thus, the bit likelihood of the SCMA detector is transformed into LLR form, denoted by LLR_{SCMA}, as follows [8]:

$$
LLR_{SCMA}^B\left(b_{(l-1)Q+q}^{(k)}\right) = \log \frac{P_e^B\left(b_{(l-1)Q+q}^{(k)} = 0\right)}{P_e^B\left(b_{(l-1)Q+q}^{(k)} = 1\right)}.
\tag{7}
$$

Then, the bit-extrinsic information sequence in LLR form, denoted by $LLR_{SCMA}^B\left(\mathbf{b}^{(k)}\right)$, is de-interleaved to be used as a prior information for the Polar decoder. The de-interleaved bit-extrinsic information sequence denoted by $LLR_a^B\left(\mathbf{c}^{(k)}\right)$ is expressed as

$$
LLR_a^B\left(\mathbf{c}^{(k)}\right) = \Pi^{-1}\left\{LLR_{SCMA}^B\left(\mathbf{b}^{(k)}\right)\right\},
\tag{8}
$$

where $\Pi^{-1}\{\cdot\}$ is the de-interleaving function.

We consider the factor graph structure of the Polar code shown in Fig. 1 from [22] or in Fig. 4 from [8], and we use the same notations and terms used in [8]. The Polar decoder based on the SCAN algorithm has two LLR messages: left messages and right messages. Let $L_{s,t}^{(k)}$ and $R_{s,t}^{(k)}$ denote the left messages and right messages, respectively, of the Polar decoder for user k, where s and t denote the row and column index of the factor graph for the Polar decoder. Prior to the decoding iteration, $L_{0,t}^{(k)}$ is initialized as $LLR_a^B\left(\mathbf{c}^{(k)}\right)$, and $R_{\log_2(N),t}^{(k)}$ is initialized to 0 or ∞, according to the index of frozen bits. Then, the decoding iteration is performed. More details of the SCAN algorithm and frozen bit index set are provided in [8], [22], and [25], or they can be referred to in our simulation code at [13].

After the decoding iteration is performed (i.e., one iteration of Polar decoding), the output extrinsic messages of the Polar decoder, denoted by $LLR_{Polar}^B\left(c_n^{(k)}\right), 1 \leq n \leq N$, can be calculated as follows [8]:

$$LLR_{Polar}^B\left(c_n^{(k)}\right) = R_{0,n}^{(k)} + \alpha \frac{\overline{R_0^{(k)}}}{\overline{L_0^{(k)}}} L_{0,n}^{(k)}, \qquad (9)$$

where α is the weight factor for the Polar decoder in the JIDD scheme [8], and $\overline{R_0^{(k)}}$ and $\overline{L_0^{(k)}}$ are the average values of $\left\{R_{0,1}^{(k)}, R_{0,2}^{(k)}, \cdots, R_{0,N}^{(k)}\right\}$ and $\left\{L_{0,1}^{(k)}, L_{0,2}^{(k)}, \cdots, L_{0,N}^{(k)}\right\}$, respectively. Then, message sequence $LLR_{Polar}^B\left(\mathbf{c}^{(k)}\right)$ is interleaved, where $\mathbf{c}^{(k)} = \left\{c_1^{(k)}, c_2^{(k)}, \cdots, c_N^{(k)}\right\}$, and the interleaved messages, denoted by $LLR_{out}^B\left(\mathbf{b}^{(k)}\right)$, can be expressed as follows:

$$LLR_{out}^B\left(\mathbf{b}^{(k)}\right) = \Pi\left\{LLR_{Polar}^B\left(\mathbf{c}^{(k)}\right)\right\}, \qquad (10)$$

where $\mathbf{b}^{(k)} = \left\{b_1^{(k)}, b_2^{(k)}, \ldots, b_N^{(k)}\right\}$, and $\Pi\{\cdot\}$ is the interleaving function. Then, $LLR_{out}^B\left(\mathbf{b}^{(k)}\right)$ is transformed into the likelihood probabilities for the binary hypotheses, which can be expressed as follows [8]:

$$P_{out}^B\left(b_{(l-1)Q+q}^{(k)} = 0\right) = \frac{e^{LLR_{out}^B\left(b_{(l-1)Q+q}^{(k)}\right)}}{1 + e^{LLR_{out}^B\left(b_{(l-1)Q+q}^{(k)}\right)}}, \qquad (11)$$

$$P_{out}^B\left(b_{(l-1)Q+q}^{(k)} = 1\right) = \frac{1}{1 + e^{LLR_{out}^B\left(b_{(l-1)Q+q}^{(k)}\right)}}. \qquad (12)$$

Then, these likelihood probabilities are re-mapped into the symbol likelihood probability, denoted by $P\left(\mathbf{x}_l^{(k)}\right)$, which is computed as

$$P\left(\mathbf{x}_l^{(k)}\right) = \prod_{q=1}^{Q} P_{out}^B\left(b_{(l-1)Q+q}^{(k)} = b_{\mathbf{x}_l^{(k)}}(q)\right), \qquad (13)$$

where $b_{\mathbf{x}_l^{(k)}}(q)$ denotes the q-th bit of the Q data bits mapped to $\mathbf{x}_l^{(k)}$.

3) STEP ⑤ (VN-TO-FN-MESSAGES-ONLY UPDATE)
In the VN-to-FN-message update process, the re-mapped symbol likelihood probability $P\left(\mathbf{x}_l^{(k)}\right)$ is merged into the message update process. Specifically, $I_{u_k \to r_d}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right)$ is computed as

$$I_{u_k \to r_d}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right) = P\left(\mathbf{x}_l^{(k)}\right) \prod_{p \in \{\xi_k \setminus d\}} I_{r_p \to u_k}^{(\tau)}\left(\mathbf{x}_l^{(p)}\right). \qquad (14)$$

4) SCMA CODEWORDS ESTIMATION
The computation sequence from (3) to (14) is repeatedly carried out until the BER converges or the number of iterations reaches the parameter predetermined for the system, and then, estimated codewords $\widehat{\mathbf{x}}_l^{(k)}$ are computed as follows [3]:

$$\widehat{\mathbf{x}}_l^{(k)} = \underset{\mathbf{x}_l^{(k)} \in \chi}{\operatorname{argmax}} \left(\prod_{d \in \xi_k} I_{r_d \to u_k}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right)\right), \qquad (15)$$

where χ denotes a set of all possible codewords for user k.

B. PARALLEL MESSAGE PASSING SCHEDULE OF JIDD
Note that in each step of the flow described in the previous subsection, all users' messages can be updated or decoded together and in parallel. By enclosing the operations that can be performed in parallel in each step of Fig. 3, denoted with a pair of braces, { }, the list of message updates and decoding iterations performed in each JIDD iteration, denoted by L_{JIDD}, is expressed as follows:
L_{JIDD}:

$\{r_1 \to u_2, r_1 \to u_3, r_1 \to u_5, r_2 \to u_1, r_2 \to u_3, r_2 \to u_6,$
$r_3 \to u_2, r_3 \to u_4, r_3 \to u_6, r_4 \to u_1, r_4 \to u_4, r_4 \to u_5$
$\}$
$\Rightarrow \{ \mathbf{CD}_1, \mathbf{CD}_2, \mathbf{CD}_3, \mathbf{CD}_4, \mathbf{CD}_5, \mathbf{CD}_6 \}$
$\Rightarrow \{ u_1 \to r_2, u_1 \to r_4, u_2 \to r_1, u_2 \to r_3, u_3 \to r_1,$
$u_3 \to r_2, u_4 \to r_3, u_4 \to r_4, u_5 \to r_1, u_5 \to r_4, u_6 \to r_2,$
$u_6 \to r_3 \},$

where the first and the last (the third) pairs of braces correspond to step ① and step ⑤, respectively, in Fig. 3. The expression $r_d \to u_k$ denotes the message update from FN r_d to VN u_k, i.e., the calculation of $I_{r_d \to u_k}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right)$, and the expression $u_k \to r_d$ denotes the message update from VN u_k to FN r_d, i.e., the calculation of $I_{u_k \to r_d}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right)$. Note that each of these expressions ($r_d \to u_k$ or $u_k \to r_d$) implies multiple separate updates for all SCMA codewords belonging to the current channel codeword. The second pair of braces, $\{ \mathbf{CD}_1, \mathbf{CD}_2, \cdots, \mathbf{CD}_6 \}$, corresponds to step ③ in Fig. 3, and $\mathbf{CD}_k$ denotes the channel decoding iteration for user k with deinterleaving pre-performed. The two big arrows ($\Rightarrow$) in L_{JIDD} represent the coded bit mapping and de-mapping process from or to the messages in the variable nodes, and thus, they correspond to step ② and step ④, respectively, in Fig. 3.

IV. THE PROPOSED US-JIDD SCHEME
A. USER-SEQUENTIAL MESSAGE PASSING SCHEDULE FOR US-JIDD
In this subsection, we present the user-sequential message passing schedule in the proposed us-JIDD scheme for channel-coded SCMA systems. The main difference from JIDD is that the proposed us-JIDD scheme slices all channel decoding iterations into per-user channel decoding, inserting them into the SCMA factor graph as deep as the message update unit, as shown in Fig. 2b. To do this, we break the three processes in braces in L_{JIDD} shown in Section III-B, and then fully rearrange the elements into a user-sequential schedule. In updating messages sequentially from user u_1 to user u_6, without loss of generality, the list of message updates and decoding iterations in each us-JIDD iteration, denoted by $L_{us-JIDD}$, is ordered as follows:
$L_{us-JIDD}$:

$\{ r_2 \to u_1, r_4 \to u_1 \text{ for all } l\text{'s} \} \Rightarrow \{\mathbf{CD}_1\} \Rightarrow$
$\{ u_1 \to r_2, u_1 \to r_4 \text{ for all } l\text{'s} \} \Rightarrow$
$\{ r_1 \to u_2, r_3 \to u_2 \text{ for all } l\text{'s} \} \Rightarrow \{\mathbf{CD}_2\} \Rightarrow$
$\{ u_2 \to r_1, u_2 \to r_3 \text{ for all } l\text{'s} \} \Rightarrow$
$\{ r_1 \to u_3, r_2 \to u_3 \text{ for all } l\text{'s} \} \Rightarrow \{\mathbf{CD}_3\} \Rightarrow$
$\{ u_3 \to r_1, u_3 \to r_2 \text{ for all } l\text{'s} \} \Rightarrow$

$\{ r_3 \to u_4, r_4 \to u_4 \text{ for all } l\text{'s} \} \Rightarrow \{ \mathbf{CD}_4 \} \Rightarrow$
$\{ u_4 \to r_3, u_4 \to r_2 \text{ for all } l\text{'s} \} \Rightarrow$
$\{ r_1 \to u_5, r_4 \to u_5 \text{ for all } l\text{'s} \} \Rightarrow \{ \mathbf{CD}_5 \} \Rightarrow$
$\{ u_5 \to r_1, u_5 \to r_4 \text{ for all } l\text{'s} \} \Rightarrow$
$\{ r_2 \to u_6, r_3 \to u_6 \text{ for all } l\text{'s} \} \Rightarrow \{ \mathbf{CD}_6 \} \Rightarrow$
$\{ u_6 \to r_2, u_6 \to r_3 \text{ for all } l\text{'s} \}$

from which it can be noted that in order to maximally accelerate message propagation within one joint iteration of the MPA and channel decoding, we employ three important rules in $L_{us-JIDD}$ as follows.

- Rule 1: Each VN update[1] is followed by a channel decoding iteration only for the user corresponding to the just-updated VN. For instance, if VN u_k is updated from all FNs connected to u_k and for all l's (for all SCMA codewords in the current channel codeword), then $\mathbf{CD}_k$ is directly performed.
- Rule 2: Channel decoding iteration is followed by updating the FN nodes that are connected to the VN node just updated by channel decoding. For instance, if $\mathbf{CD}_k$ is performed, then all FN nodes that are connected to u_k are updated.
- Rule 3: Each FN update is followed by updating the VN of the next user to be channel-decoded. For instance, if the FN nodes that are connected to u_k are updated for all l's, then the next user's VN node, i.e., u_{k+1} is updated from all FN nodes connected to VN u_{k+1} for all l's.

Note that $L_{us-JIDD}$ and L_{JIDD} have the same operational elements, but the elements are listed in a different order. Therefore, JIDD and the proposed us-JIDD have identical computational complexity per joint MPA iteration. However, they have quite different message propagation acceleration. In JIDD, all users' channel decoding iteration results are used at the same time in between two one-directional parallel message updates in the MPA, i.e., FN-to-VN-messages-only updates and VN-to-FN-messages-only updates as shown in L_{JIDD}. Whereas, in us-JIDD, each user's channel decoding iteration is inserted into a different timing position in the sequential MPA, as shown in $L_{us-JIDD}$, and is sequentially and sparsely performed in the middle of the sequential MPA. Thus, even with a single joint MPA iteration, user-sequential channel decoding gradually enhances message reliability, and then, $L_{us-JIDD}$ effectively achieves the number of channel decoding iterations equal to K, which significantly accelerates message propagation. The power of fast convergence with us-JIDD comes from this user-sequential joint MPA mechanism.

B. JOINT MPA-BASED DETECTION IN US-JIDD

To accommodate the rearranged processing order in us-JIDD, as shown in $L_{us-JIDD}$, the joint MPA needs to be modified accordingly. In this subsection, the mathematical operations

---

[1]We define the term *VN update* as updating messages from all FNs connected to a single VN such that the phrase VN update for u_k includes $(r_j \to u_k)$ for all r_j's connected to u_k and for all SCMA codewords in the current channel codeword, i.e., for all l's. The term *FN update* is defined likewise.

of the us-JIDD scheme are described based on those for JIDD shown in Section III-A. Thus, only the modified parts from the JIDD scheme are described.

1) FN-TO-VN-MESSAGES UPDATE IN THE US-JIDD SCHEME

In the us-JIDD scheme, the update process $r_j \to u_k$ follows the S-MPA [12] (where the message update is performed in a sequential manner), rather than the original MPA (where the message update is performed in a parallel manner). Thus, Eq. (3) is modified as follows:

$$
I_{r_j \to u_k}^{(\tau)} \left(\mathbf{x}_l^{(k)} \right)
$$

$$
= \sum_{\mathbf{x}_l^{(p)}, p \in \{\zeta_j \backslash k\}} \left\{ \frac{1}{\sqrt{2\pi}\sigma} exp \left(-\frac{1}{2\sigma^2} \left\| \mathbf{y}_l - \sum_{m \in \zeta_j} h_{l,j}^{(m)} x_{l,j}^{(m)} \right\|^2 \right) \right.
$$

$$
\left. \times \prod_{\substack{p \in \{\zeta_j \backslash k\} \\ p < k}} I_{u_p \to r_j}^{(\tau)} \left(\mathbf{x}_l^{(p)} \right) \prod_{\substack{p \in \{\zeta_j \backslash k\} \\ p > k}} I_{u_p \to r_j}^{(\tau-1)} \left(\mathbf{x}_l^{(p)} \right) \right\}, \quad (16)
$$

Note that, unlike (3), the message $I_{u_p \to r_j}^{(\tau)} \left(\mathbf{x}_l^{(p)} \right)$ in (16) is the sequentially updated version in the middle of the current MPA iteration, i.e., the τ-th MPA iteration. Moreover, distinguished from the S-MPA, message $I_{u_p \to r_j}^{(\tau)} \left(\mathbf{x}_l^{(p)} \right)$ in (16) is enhanced further by sequential channel decoding in the middle of the current MPA iteration. This sequential message update scheme can utilize more-reliable and fresh messages, i.e., messages updated in the current iteration, and thus, it can achieve accelerated convergence.

2) CHANNEL DECODING PROCESS IN THE US-JIDD SCHEME

From (9), recall that in the JIDD scheme, the output extrinsic messages of Polar decoder $LLR_{Polar}^B \left(c_n^{(k)} \right)$ are calculated by combining the left message $L_{0,c_n^{(k)}}^{(k)}$ (which corresponds to the message from the SCMA detector) and the right message $R_{0,c_n^{(k)}}^{(k)}$, and the weight between the two messages is adjusted by weight factor α. This is because, in JIDD, all users' messages are updated in parallel, and thus, the weighted combination of the Polar decoding message and the SCMA detector message is performed inside the SCAN algorithm.

Whereas, in us-JIDD, the users' Polar decoding messages are sequentially updated in the joint MPA, and thus, they need to be combined with the SCMA detector message right after they are updated. Hence, $LLR_{Polar}^B \left(c_n^{(k)} \right)$ in (9) is modified as follows:

$$
LLR_{Polar}^B \left(c_n^{(k)} \right) = R_{0,c_n^{(k)}}, \quad (17)
$$

and instead, the weighted combination of the Polar decoding message with the SCMA detector message is performed in the *VN-to-FN-messages update* stage, which is explained next.

3) VN-TO-FN-MESSAGES UPDATE IN THE US-JIDD SCHEME

In the us-JIDD scheme, update process $u_k \rightarrow r_j$ in (14) is modified to

$$I_{u_k \rightarrow r_d}^{(\tau)}\left(\mathbf{x}_l^{(k)}\right) = P\left(\mathbf{x}_l^{(k)}\right)^{\beta} \prod_{p \in \{\xi_k \setminus d\}} I_{r_p \rightarrow u_k}^{(\tau)}\left(\mathbf{x}_l^{(p)}\right), \quad (18)$$

where β is the weight factor in log scale for the symbol likelihood probability from the Polar decoder, i.e., $P\left(\mathbf{x}_l^{(k)}\right)$. The role of β in us-JIDD is the same as that of α in (10) for JIDD in the sense that it adjusts the portions of the two messages, i.e., one from the Polar decoding iteration and the other from the MPA iteration in SCMA.

C. LATENCY ISSUE AND ITS SOLUTION

1) LATENCY ISSUE

The overall latency to get the final decoding result, denoted by T_D, is given as

$$T_D = S T_{itr} \quad (19)$$

where T_{itr} denotes the execution time required for one joint MPA iteration, and S denotes the number of joint MPA iterations required for BER convergence.

Recall that in L_{JIDD} and $L_{us-JIDD}$, all operational elements in each pair of braces can be executed in parallel. Whereas, the braces should be executed one after another, because the results of the preceding braces should be used for the operations in the next braces. Consequently, there exists an inevitable computation time equal to the sum of the processing times for each brace pair ({ }). Let T_{FN}, T_{CD}, and T_{VN} denote the processing times for the three pair of braces in L_{JIDD}, each of which corresponds to the FN-to-VN-messages update, the channel decoding iteration, and VN-to-FN-messages update, respectively. Then, T_{itr} for JIDD, denoted by $T_{itr}^{(JIDD)}$, is expressed as

$$T_{itr}^{(JIDD)} = T_{FN} + T_{VN} + T_{PD}. \quad (20)$$

On the other hand, the number of brace pairs in $L_{us-JIDD}$ is K times more than L_{JIDD}, because the operations are grouped and reordered in user-sequential order. Instead, the operations in those three kinds of brace pairs in $L_{us-JIDD}$ are K times fewer than in L_{JIDD}. Nonetheless, in order to consider the most pessimistic latency in us-JIDD compared to JIDD, let us assume that the computation times for the three kinds of brace pairs in $L_{us-JIDD}$ are identical to those for L_{JIDD} in (20). Then, the worst case (i.e., the upper bound of) T_{itr} for us-JIDD, denoted by $T_{itr}^{(us-JIDD)}$, is K times more than JIDD, as follows:

$$\begin{aligned} T_{itr}^{(us-JIDD)} &= K \times (T_{FN} + T_{VN} + T_{PD}) \\ &= K T_{itr}^{(JIDD)} \end{aligned} \quad (21)$$

2) SOLUTION TO LATENCY ISSUE

We alleviate this latency issue in two ways. First, the fast convergence rate of us-JIDD inherently reduces latency, because overall latency T_D is simply proportional to S, as shown in (19). The numerical results in the following section show that S for us-JIDD is substantially smaller than S for JIDD. Despite the reduced S in us-JIDD, T_D of us-JIDD is still longer than T_D in JIDD because T_{itr} of us-JIDD is K times more than JIDD, as shown in (21). Hence, in order to fundamentally decrease scaling factor K for T_{itr} in (21), we additionally apply a group method to us-JIDD. If we set the number of groups G to 3, then six ($=K$) users can be grouped into three sets, i.e., $\{u_1, u_2\}$, $\{u_3, u_4\}$, $\{u_5, u_6\}$, and the list of message updates and decoding iterations, denoted by $L_{us-JIDD}^{G=3}$, is structured as follows:

$L_{us-JIDD}^{G=3}$:
  $\{ r_2 \rightarrow u_1, r_4 \rightarrow u_1, r_1 \rightarrow u_2, r_3 \rightarrow u_2 \text{ for all } l\text{'s} \} \rightarrow$
  $\{ \mathbf{CD}_1, \mathbf{CD}_2 \} \rightarrow$
  $\{ u_1 \rightarrow r_2, u_1 \rightarrow r_4, u_2 \rightarrow r_1, u_2 \rightarrow r_3 \text{ for all } l\text{'s} \} \rightarrow$
  $\{ r_1 \rightarrow u_3, r_2 \rightarrow u_3, r_3 \rightarrow u_4, r_4 \rightarrow u_4 \text{ for all } l\text{'s} \} \rightarrow$
  $\{ \mathbf{CD}_3, \mathbf{CD}_4 \} \rightarrow$
  $\{ u_3 \rightarrow r_1, u_3 \rightarrow r_2, u_4 \rightarrow r_3, u_4 \rightarrow r_2 \text{ for all } l\text{'s} \} \rightarrow$
  $\{ r_1 \rightarrow u_5, r_4 \rightarrow u_5, r_2 \rightarrow u_6, r_3 \rightarrow u_6 \text{ for all } l\text{'s} \} \rightarrow$
  $\{ \mathbf{CD}_5, \mathbf{CD}_6 \} \rightarrow$
  $\{ u_5 \rightarrow r_1, u_5 \rightarrow r_4, u_6 \rightarrow r_2, u_6 \rightarrow r_3 \text{ for all } l\text{'s} \}$

where the number of brace pairs decreases to half ($= G/K$) of that for $L_{us-JIDD}$. Therefore, K in (21) is replaced by G, and this is a T_{itr} only 3 ($= G$) times larger than for JIDD. Moreover, $L_{us-JIDD}^{G=3}$ obtains exactly the same message updates as $L_{us-JIDD}$. This is because the users are paired (grouped) to satisfy a condition that their VNs do not share resources in the SCMA code structure. For instance, Fig. 1 shows that VNs u_1 and u_2 are connected to disjointed resource sets, i.e., (r_2, r_4) and (r_1, r_3), respectively. This is the case for the other user pairs, (u_3, u_4) and (u_5, u_6), as well. Therefore, the message updates from or to the paired VNs (users) do not affect each other, and thus, even if they are performed in parallel, the message update results are the same as when done sequentially. Thus, $L_{us-JIDD}$ and $L_{us-JIDD}^{G=3}$ have identical detection performance, which will be confirmed in the next section. This also implies that even with a fully user-sequential MPA by $L_{us-JIDD}$ for the considered factor graph structure in Fig. 1, the number of effective user-sequential message propagations within a single MPA iteration is 3, i.e., half the number of users.

Summing up, owing to the two factors of us-JIDD, i.e., 1) the substantially smaller S compared to JIDD, and 2) the reduced T_{itr} in the grouping method with $L_{us-JIDD}^{G=3}$ instead of $L_{us-JIDD}$, the overall latency of us-JIDD becomes equal to that of JIDD. This will be confirmed in the next section, too.

3) FURTHER ACCELERATION IN DOWNLINK SCENARIOS

Due to the sequential message propagation of us-JIDD, the user's message to be updated later becomes more reliable. Hence, in downlink scenarios where the user detection order in $L_{JIDD}^{G=3}$ can be set differently, user by user, we can further accelerate the convergence rate simply by setting the detection order so that the desired user's message is updated last. For instance, $L_{JIDD}^{G=3}$ illustrated in the previous section is the optimal detection order for the receivers of user 5 and

TABLE 1. Number of operations for the three major processes.

Major process	Number of operations
Updating messages from FN to VN at each FN	$O_{\text{FN}} = d_f M^{d_f - 1}(M + d_f - 2)$ $+ d_f(M^{d_f} - M)$
Updating messages from VN to FN at each VN	$O_{\text{VN}} = M d_v \min(3, d_v) + (M - 1)d_v$
Channel decoding by the SCAN algorithm	$O_{\text{PD}} = 10N \log(N)$

TABLE 2. System parameters.

	Parameter	Value
Common parameters	Data bits per user in frame, m	128
	Interleaving scheme	Random
Polar code	Decoding algorithm	Soft cancellation (SCAN)
	Code rate C_r	1/2
	Code length, $N(= m/C_r)$	256
	Design SNR [24]	0.5 dB
	Frozen index set $\mathcal{F}$ of the encoder	Same as in [25]
JIDD scheme	Weight factor in Polar decoder α [8]	0.7 (optimal value from [8])
us-JIDD scheme	Weight factor β in (14)	0.4 (optimal value obtained by simulations)

user 6, who are updated last. In the receivers for user 1 and user 2, we only need to move the first three lines of $L_{\text{JIDD}}^{G=3}$ in the previous section to the end of the list, and in the receivers for user 3 and user 4, we only need to move the second three lines (lines 4-6) to the end of the list. Note that this further acceleration technique is not valid for uplink scenarios because the different orderings are not allowed in a single receiver on the BS side due to computation complexity, which is the main concern of this paper.

V. COMPLEXITY ANALYSIS

The overall number of operations for BER convergence, denoted by O_C, is given as

$$O_C = S O_{itr}, \tag{22}$$

where O_{itr} denotes the number of operations for one joint MPA iteration.

Recall that L_{JIDD} and $L_{us-\text{JIDD}}$ contain the same operational elements. Therefore, the number of operations for one iteration is the same for both JIDD and us-JIDD. Let O_{FN} and O_{VN} denote the number of computational operations for the FN-to-VN-messages update at each FN, and the VN-to-FN-messages update at each VN, and let O_{PD} denote the number of computational operations for channel decoding iterations. Then, O_{itr} for the JIDD and us-JIDD schemes is commonly expressed as

$$O_{itr} = LD \times O_{FN} + LK \times O_{VN} + K \times O_{PD}. \tag{23}$$

For reference, Table 1 provides details about O_{FN}, O_{PD}, and O_{VN} [11], [23], where d_f and d_v are the number of branches connected to each FN and VN, respectively.

Because O_{itr} is the same for JIDD and us-JIDD, overall complexity O_C in (22) is determined by S, which is different for JIDD and us-JIDD. In Section VI, we compare O_C between JIDD and us-JIDD based on the simulated S values for both schemes.

VI. NUMERICAL RESULTS

A. BER AND CONVERGENCE RATE

Table 2 shows the system parameters used in our simulations. The SCMA codebook used is shown in Section II.A of [26]. Fig. 4 shows the simulated channel-coded BERs as a function of the number of joint MPA iterations for a downlink scenario on an AWGN channel, where $h_{l,d}^{(k)}$ is commonly set to 1 for all k, l, and d [27]. The S-MPA combined with IDD achieves worse BER performance than JIDD. This confirms
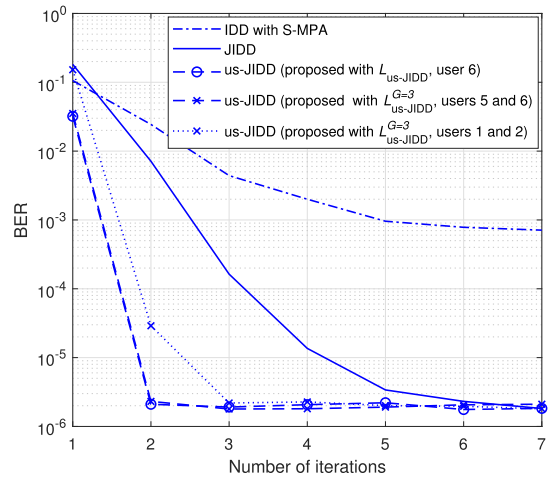

FIGURE 4. Simulated channel-coded BERs as a function of the number of joint MPA iterations for a downlink scenario on an AWGN channel, $E_b/N_0 = 5$dB.

that the benefit from the of S-MPA (i.e., sequential message propagation within an MPA iteration) fails to be conserved in the IDD process. Remarkably, the proposed us-JIDD scheme converges to the same BER as the converged BER of JIDD after only two iterations, i.e., $S = 2$, whereas for JIDD, $S = 6$. In accordance with our analysis in Section IV-C1, the user grouping method ($G = 3$) for latency reduction achieves BER performance identical to that of the fully user-sequential method ($G = 6 (= K)$). Also shown is that with message update order $L_{us-\text{JIDD}}^{G=3}$, the users updated first, i.e., user 1 and user 2, require $S = 3$, whereas the users updated last, i.e., user 5 and user 6, require $S = 2$. This confirms the property of the proposed us-JIDD, mentioned in Section IV-C3, that BER convergence can be further accelerated by updating the desired user last in the user-sequential message update order.

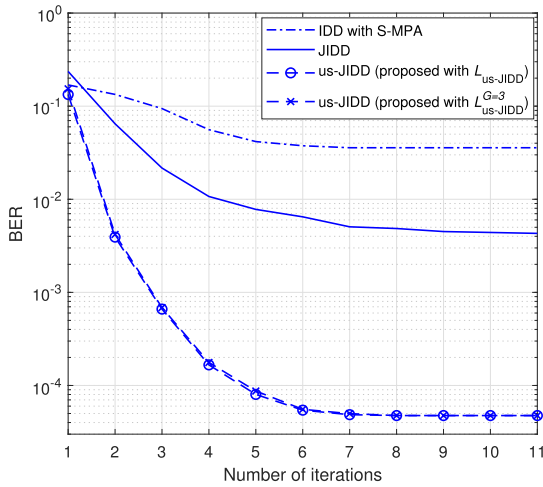Fig. 5 shows the simulated channel-coded BERs as a function of the number of joint MPA iterations for an uplink scenario on a flat channel with perfect power control, where $h_{l,d}^{(k)}$ is set to $e^{j\theta}$ and random phase θ is uniformly distributed over $[0 \ 2\pi]$ and is identical for a different l and d, but independently generated for a different k. In the uplink scenarios, the designed coding gain from SCMA with the

FIGURE 5. Simulated channel-coded BERs as a function of the number of joint MPA iterations for an uplink scenario on a flat channel with perfect power control, $E_b/N_0 = 6$dB.

FIGURE 6. Simulated channel-coded BERs as a function of the number of joint MPA iterations for a downlink scenario on a Rayleigh fading channel.

FIGURE 7. Simulated channel-coded BERs as a function of the number of joint MPA iterations for an uplink scenario on a Rayleigh fading channel.

multidimensional complex codebook is not preserved due to phase mismatch among the signals received from the users. This is confirmed in Fig. 5, which shows the increased BERs compared to the downlink scenario in Fig. 4, despite the increased SNR. Note that the converged BER level of JIDD on a flat channel with perfect power control is unacceptably high. Meanwhile, the converged BER level with us-JIDD is significantly lower than with JIDD. This result is meaningful in that, for the flat channel with perfect power control, the proposed us-JIDD outperforms JIDD in terms of converged BER level rather than in terms of convergence speed. From a practical point of view, this out-performance can improve the convergence speed as well. For instance, if the target BER is the same as the converged BER with JIDD, the effective S of the proposed us-JIDD scheme is reduced to 2 because the proposed us-JIDD scheme requires only two iterations to get the same BER level as the converged BER with JIDD.

Fig. 6 shows the simulated channel-coded BERs as a function of the number of joint MPA iterations for a downlink scenario on a Rayleigh fading channel, where $h_{l,d}^{(k)}$ is a complex Gaussian random variable with unit variance, and is identical for a different k, but is independently generated for a different l and d [27]. The proposed us-JIDD has the same trend as that in the AWGN channel in Fig. 4, such as very fast convergence compared to JIDD, identical performance based on grouping method and further acceleration from the proper detection order in the downlink scenario. Most of all, the proposed us-JIDD scheme achieves the same BER as the converged BER with JIDD after only two iterations (i.e., $S = 2$), whereas with JIDD, $S = 6$. Also shown is that all these features hold for different SNRs.

Fig. 7 shows the simulated channel-coded BERs as a function of the number of joint MPA iterations for uplink scenario in a Rayleigh fading channel, where $h_{l,d}^{(k)}$ is a complex Gaussian random variable with unit variance, and is independently generated for a different k, l, and d [28].
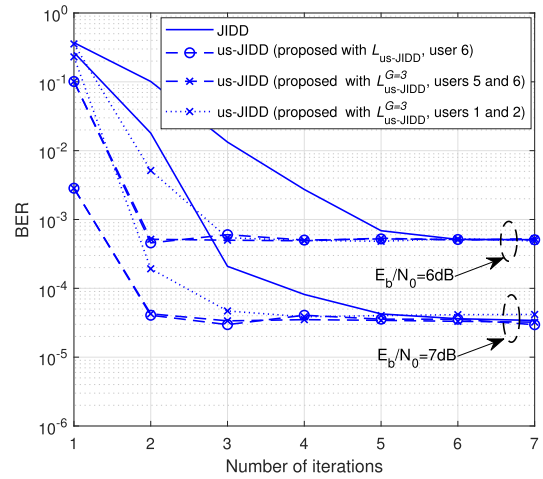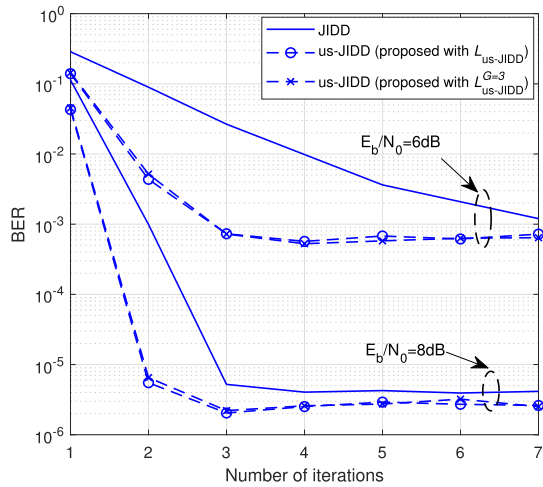
Unlike the results for the uplink scenario on a flat channel in Fig. 5, there is no improvement in the converged BER performance, but very fast BER convergence is achieved. For $E_b/N_0 = 6$ dB, the proposed us-JIDD converges when $S = 3$, whereas JIDD converges when $S = 7$. For $E_b/N_0 = 8$ dB, both the proposed us-JIDD and the JIDD converge when $S = 3$, but the proposed us-JIDD almost converges even after two iterations, whereas the BER under JIDD after two iterations is significantly larger than under the proposed us-JIDD.

B. COMPUTATIONAL COMPLEXITY AND LATENCY COMPARISON

The ultra-reliable and low-latency communication (URLLC) service is one of the major service categories of 5G, and on downlink where the receiver is a small device, computational complexity and latency are more important system design factors, compared to uplink. Thus, in this subsection, we analyze computational complexity and latency in a downlink scenario with a Rayleigh fading channel.
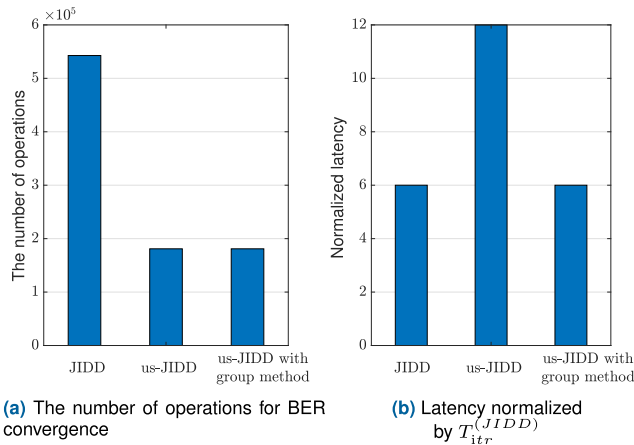
(a) The number of operations for BER convergence

(b) Latency normalized by $T_{itr}^{(JIDD)}$

FIGURE 8. Computational complexity and latency of JIDD, us-JIDD and us-JIDD with the group method in a downlink scenario with a Rayleigh fading channel, K=6, G=3.

Fig. 8a shows the number of operations for BER convergence, O_C in (22), for the three schemes in comparison. Note that O_{itr} in (22) is common to the different schemes, and thus, the relative difference in O_C among the different schemes is determined by S in each scheme. Recall that in Fig. 6, $S = 6$ for the JIDD scheme, whereas $S = 2$ for both the us-JIDD scheme and the us-JIDD scheme with the group method. Hence, the proposed us-JIDD scheme saves 66.7% of the operations, compared to the JIDD scheme. Reducing computational complexity by decreasing S is a remarkable benefit of the us-JIDD scheme, especially on downlink because power consumption is more critical in small receivers such as user terminals, and thus, the low complexity-detection algorithm is preferable on downlink.

Fig. 8b shows the total latency for BER convergence normalized by $T_{itr}^{(JIDD)}$, i.e., $T_D/T_{itr}^{(JIDD)}$ is plotted. Despite the reduced S, the latency of the us-JIDD scheme is longer than the JIDD scheme due to the sequential message update strategy. However, by using a group-wise sequential method, the us-JIDD scheme can reduce latency and achieve the same latency as the JIDD scheme. Summing up the results in Fig. 8a and Fig. 8b, we conclude that the us-JIDD scheme with the group method substantially reduces computational complexity while maintaining the same latency as the conventional JIDD scheme.

VII. CONCLUSION
In this paper, we proposed a novel, user-sequential JIDD scheme for a channel-coded SCMA system. In the proposed scheme, message enhancement from channel decoding immediately propagates to the connected users' messages in the factor graph within each joint MPA iteration. We showed that the proposed scheme significantly reduces computational complexity by dramatically accelerating BER convergence, compared to conventional JIDD schemes. We also confirmed that the latency issue with the sequential message update idea was alleviated by two factors: 1) a fast convergence rate, and 2) the group-wise sequential method. Therefore,

we conclude that the proposed scheme is a competitive solution for channel-coded SCMA systems.

REFERENCES

[1] H. Nikopour, E. Yi, A. Bayesteh, K. Au, M. Hawryluck, H. Baligh, and J. Ma, "SCMA for downlink multiple access of 5G wireless networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 3940–3945.

[2] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2181–2195, Oct. 2017.

[3] L. Yang, Y. Liu, and Y. Siu, "Low complexity message passing algorithm for SCMA system," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2466–2469, Dec. 2016.

[4] L. Lei, C. Yan, G. Wenting, Y. Huilian, W. Yiqun, and X. Shuangshuang, "Prototype for 5G new air interface technology SCMA and performance evaluation," *China commun.*, vol. 12, no. 9, pp. 38–48, Dec. 2015.

[5] Y. Wu, S. Zhang, and Y. Chen, "Iterative multiuser receiver in sparse code multiple access systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 2918–2923.

[6] K. Han, Z. Zhang, J. Hu, and J. Chen, "A high performance joint detection and decoding scheme for LDPC coded SCMA system," in *Proc. IEEE Globecom Workshops*, Feb. 2016, pp. 1–6.

[7] Z. Zhang, K. Han, J. Hu, and J. Chen, "Joint detection and decoding schemes for turbo coded SCMA systems," in *Proc. IEEE Globecom Workshops*, Feb. 2016, pp. 1–6.

[8] Z. Pan, E. Li, L. Zhang, J. Lei, and C. Tang, "Design and optimization of joint iterative detection and decoding receiver for uplink polar coded SCMA system," *IEEE Access*, vol. 6, pp. 52014–52026, 2018.

[9] B. Tahir, S. Schwarz, and M. Rupp, "BER comparison between convolutional, Turbo, LDPC, and Polar codes," in *Proc. 24th Int. Conf. Telecommun. (ICT)*, May 2017, pp. 1–7.

[10] S. Shao, P. Hailes, T.-Y. Wang, J.-Y. Wu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Survey of turbo, LDPC, and polar decoder ASIC implementations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2309–2333, 3rd Quart., 2019.

[11] U. U. Fayyaz and J. R. Barry, "Low-complexity soft-output decoding of polar codes," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 958–966, May 2014.

[12] Y. Du, B. H. Dong, Z. Chen, J. Fang, and L. Yang, "Shuffled multiuser detection schemes for uplink sparse code multiple access systems," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1231–1234, Jun. 2016.

[13] D. Na and K. Choi, *A Very Fast Joint Detection for Polar-Coded SCMA*. Accessed: Feb. 27, 2022. [Online]. Available: https://bwchome2021.wixsite.com/bwcyu/downloadsubmit

[14] Y. Du, B. Dong, Z. Chen, J. Fang, and X. Wang, "A fast convergence multiuser detection scheme for uplink SCMA systems," *IEEE Wireless Commun. Lett.*, vol. 5, no. 4, pp. 388–391, Aug. 2016.

[15] H. Mu, Z. Ma, M. Alhaji, P. Fan, and D. Chen, "A fixed low complexity message pass algorithm detector for up-link SCMA system," *IEEE Wireless Commun. Lett.*, vol. 4, no. 6, pp. 585–588, Dec. 2015.

[16] X. Ma, L. Yang, Z. Chen, and Y. Siu, "Low complexity detection based on dynamic factor graph for SCMA systems," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2666–2669, Dec. 2017.

[17] C. Zhang, Y. Luo, and Y. Chen, "A low-complexity SCMA detector based on discretization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2333–2345, Apr. 2018.

[18] J. Miao, X. Hu, and Z. Zhao, "A low complexity multiuser detection scheme with dynamic factor graph for uplink SCMA systems," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Changchun, China, Aug. 2019, pp. 846–851.

[19] A. Bayesteh, H. Nikopour, M. Taherzadeh, H. Baligh, and J. Ma, "Low complexity techniques for SCMA detection," in *Proc. IEEE Globecom Workshops*, Dec. 2015, pp. 1–6.

[20] L. Yang, X. Ma, and Y. Siu, "Low complexity MPA detector based on sphere decoding for SCMA," *IEEE Commun. Lett.*, vol. 21, no. 8, pp. 1855–1858, Aug. 2017.

[21] A. B. Sergienko and V. P. Klimentyev, "SCMA detection with channel estimation error and resource block diversity," in *Proc. Int. Siberian Conf. Control Commun. (SIBCON)*, May 2016, pp. 1–5.

[22] U. U. Fayyaz and J. R. Barry, "Polar codes for partial response channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 4337–4341.

[23] A. Bayesteh, H. Nikopour, M. Taherzadeh, H. Baligh, and J. Ma, "Low complexity techniques for SCMA detection," in *Proc. IEEE Globecom Workshops*, Dec. 2015, pp. 1–6.

[24] U. U. Fayyaz and J. R. Barry, "Polar code design for intersymbol interference channels," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2357–2362.

[25] H. Vangala and E. Viterbo. *Polar Coding Algorithms in MATLAB*. Accessed: Feb. 23, 2016. [Online]. Available: https://ecse.monash.edu/staff/eviterbo/polarcodes.html

[26] Altera Innovate Asia. Presentation. *1st 5G Algorithm Innovation Competition-ENV1.0-SCMA*. Accessed: Feb. 1, 2022. [Online]. Available: http://www.innovateasia.com/5g/en/gp2.html

[27] C.-P. Wei, H. Yang, C.-P. Li, and Y.-M. Chen, "SCMA decoding via deep learning," *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 878–881, Apr. 2021.

[28] J. Bao, Z. Ma, Z. Ding, G. K. Karagiannidis, and Z. Zhu, "On the design of multiuser codebooks for uplink SCMA systems," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 1920–1923, Oct. 2016.

DONGJUN NA received the B.S., M.S., and Ph.D. degrees from Yeungnam University, Gyeongsan, South Korea, in 2014, 2016, and 2020, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information and Communication Engineering. His current research interests include physical layer for fifth-generation (5G) and beyond communication systems.

WON-GI SEO received the B.S., M.S., and Ph.D. degrees in electronic and electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 1993, 1995, and 1999, respectively. Since 2007, he has been the CEO of Nextwill. His research interests include nonlinear compensation receiver and transmitter.

HONGSUK SHIM received the B.S. degree in computer engineering from Chungbuk National University, Cheongju, in 2005, and the M.S. degree in computer engineering from Chungnam National University, Daejeon, in 2008. He is currently working as a Senior Researcher with Hanwha Systems, Pangyo, South Korea. His research interests include passive radar (PET and PCL), the electronic-warfare (EW) systems, electronic support (ES), electronic attack (EA) and ELINT/COMINT signal analysis, and signal processing.

KWANWOONG RYU received the B.S. and M.S. degrees in electronics engineering and the Ph.D. degree in information and communication engineering from Yeungnam University, South Korea, in 1997, 1999, and 2006, respectively. From 2004 to 2005, he was an Internship Student with the IP Radio Network Development Department, NTT DoCoMo Inc., Japan. From 2006 to 2009, he was a Senior Staff Engineer with XRONet Corporation. Since July 2009, he has been with the Electronics and Telecommunication Research Institute (ETRI), where he is currently a Principal Member of Research Staff. His research interests include digital broadcasting and communication systems.

KWONHUE CHOI (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic and electrical engineering from the Pohang University of Science and Technology, Pohang, South Korea, in 1994, 1996, and 2000, respectively. From 2000 to 2003, he was with the Electronics and Telecommunications Research Institute, Daejeon, South Korea, as a Senior Research Staff Member. In 2003, he joined the Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, South Korea, where he is currently a Professor. He has authored a textbook: *Problem-Based Learning in Communication Systems Using MATLAB and Simulink* (Wiley, 2016). His research interests include signal design for the communication systems, multiple access schemes, diversity schemes for wireless fading channels, multiple antenna systems, and in-band full duplex systems.

● ● ●