Iterative Detection and Decoding of Finite-Length Polar Codes in Gaussian Multiple Access Channels

Moustafa Ebada, Sebastian Cammerer, Ahmed Elkelesh, Marvin Geiselhart and Stephan ten Brink
Institute of Telecommunications, Pfaffenwaldring 47, University of Stuttgart, 70569 Stuttgart, Germany
{ebada,cammerer,elkelesh,geiselhart,tenbrink}@infe.uni-stuttgart.de

Abstract—We consider the usage of finite-length polar codes for the Gaussian multiple access channel (GMAC) with a finite number of users. Based on the interleave-division multiple-access (IDMA) concept, we implement an iterative detection and decoding non-orthogonal multiple access (NOMA) receiver that benefits from a low complexity, while scaling (almost) linearly with the amount of active users. We further show the conceptual simplicity of the belief propagation (BP)-based decoder in a step-by-step illustration of its construction. Beyond its conceptual simplicity, this approach benefits from an improved performance when compared to some recent work tackling the same problem, namely the setup of finite-length forward error-correction (FEC) codes for finite-number of users. We consider the fifth generation mobile communication (5G) polar code with a block length $N = 512$ applied to both a two-user and a four-user GMAC scenario with a sum-rate of $R_{\text{sum}} = 0.5$ and $R_{\text{sum}} = 1$, respectively. Simulation results show that a BP-based soft interference cancellation (SoIC) receiver outperforms a joint successive cancellation (JSC) scheme. Finally, we investigate the effect of a concatenated repetition code which suggests that alternative polar code design rules are required in multi-user scenarios.

I. INTRODUCTION

Polar codes [1] are not only proven to be capacity achieving when the block length tends to infinity, they also have shown to outperform low-density parity-check (LDPC) codes in the short-length regime [2]. A key feature of polar codes is their per-bit-level rate flexibility which can be seen as one of the reasons for their selection for the control channels of the 5G mobile communications standard [3]. Since their introduction, several decoding algorithms for polar codes have been developed. While successive cancellation list (SCL) decoding [4] is widely used, its inherent hard-decision output is limiting its application in iterative receiver architectures, such as iterative detection and demapping (IDD). Moreover, the serial nature of the algorithm puts limits to achievable latency and throughput. In contrast, iterative decoding schemes for polar codes such as BP [5] and its derivative [6] are based on the exchange of soft messages and, thus, appear more promising for iterative multi-user detection.

Driven by the continuous increase of active network devices, NOMA schemes have become a fast-growing field of research, opening up many exciting new research possibilities that require a careful reconsideration and re-evaluation of existing coding schemes [7]. For this, we focus on the application of short length communications motivated by IoT sensor networks or similar industrial applications with strict latency requirements that prevent designers from using longer codewords. Note that, besides its conceptual simplicity in terms of orthogonalization of the channel resources, such a multi-user system may also significantly reduce the medium access latency. However, contrarily to the single-user case, the multi-user setup is characterized by many more code design parameters such as signal-to-noise-ratio (SNR), flexible number of users and, potentially, non-equal transmit powers that need to be considered for the optimal code design [8]. This amount of configurable system parameters – leading to the need of an individual code rate per user – renders a practical deployment difficult. In particular for the short length regime, it urges the need for a multi-user coding scheme with the highest possible rate flexibility while maintaining a powerful error-rate performance at low decoding complexity.

Polar codes were analytically shown to achieve the channel capacity region of the two-user binary input multiple access channel (MAC) [9], [10]. In [11], this was proven for the K-user scenario. However, none of these work provide an efficient decoding scheme for the case of finite code blocklength. The authors of [12] have proposed to use finite-length polar codes in an JSC-based IDD setup for two and four users for the special case of the unsourced MAC.

We adapt the well-known IDMA principle [13] to short

---

Fig. 1: Block diagram of a 2-user BP-based IDMA receiver.

This work has been supported by DFG, Germany, under grant BR 32055-1.
length polar codes with iterative decoding. IDMA is an attractive NOMA superposition coding scheme known to have an exceptionally good performance while maintaining a low decoding complexity using a simple multi-user detector (MUD) based on parallel interference cancellation (PIC). All users can transmit using the same channel code and modulation scheme, only a user-specific interleaver is required. We show that polar codes under iterative decoding can be seamlessly integrated into the conventional IDMA scheme by replacing the LDPC coding scheme. This promises an unseen rate flexibility on a per-bit-level and the full compatibility to the existing 5G polar coding framework. In this work, we focus on (relatively) short-length NOMA schemes that are difficult to realize with LDPC codes [8] due to their inherent random graph structure. Because of its inherent soft-in/soft-out (SISO) property, BP decoding of polar codes appears to be a promising candidate for iterative decoding and detection for multiple users allowing fully parallel decoding.

II. PRELIMINARIES

We provide a brief introduction to the IDMA system model and refer the interested reader to [8] for further details about the system setup.

A. IDMA System Model

Consider the IDMA system model with $K_a$ active users, where each user encodes and decodes its data separately using a fixed polar code of code rate $R_c$ followed by a repetition code of rate $R_r = \frac{1}{d}$, where $R_r = R_c \cdot R_i$, and $R_i$ is the input coding rate per user. After interleaving, the coded bits of the $i$-th user undergo a binary phase shift keying (BPSK) mapping to $x_i$. Outputs from $K_a$ active users are superimposed and transmitted over an additive white Gaussian noise (AWGN) channel which can be expressed as

$$y = \sum_{i=1}^{K_a} \sqrt{P_i} \cdot \tilde{x}_i + n$$

where $P_i$ is the received signal power of the $i$-th user, $\tilde{x}_i = x_i \cdot e^{j\phi_i}$, with $\phi_i$ being random phase shifts which are independently and uniformly distributed in $[0, \pi)$ and user-specific. This phase scrambling was shown to improve the performance of superimposed multi-user scenarios [14], especially for AWGN channels.

B. Polar Codes

Polar codes are the first provably capacity-achieving channel codes over any arbitrary binary-input discrete memoryless channel [1]. Channel polarization is the core idea of polar codes, where synthesized bit-channels show a polarization behavior. The synthesized bit-channels polarize such that bit-channels become either noiseless or completely noisy, and, thus, can be used for information/data transmission or frozen/known bit transmission. A polar code of length $N$ and code dimension $k$ (i.e., $\mathcal{P}(k, N)$) can be fully described with the frozen-bits set. Since every bit-channel can be either frozen or non-frozen, we can represent the frozen and non-frozen bits by a logical vector $\hat{a}$ of length $N$. Element $i$ in the $\hat{a}$-vector (i.e., $a_i$) indicates whether the $i$-th bit-channel is frozen or not. Throughout this work, $a_i = 0$ means that the $i$-th bit-channel is a frozen bit, while $a_i = 1$ indicates an information bit-channel. Thus, the polar code rate can be formulated as $R_c = \frac{k}{N} = \frac{\sum_{i=1}^{N} a_i}{N}$.

Originally, the choice of frozen/non-frozen bit-channels was based on the assumption of using a successive cancellation (SC) decoder to decode the polar code [1]. However, for finite length polar codes, the SC decoder performs poorly in terms of error-rate. Thus, other decoding algorithms were proposed in the literature (e.g., BP decoder [5] and SCL decoder [4]) outperforming the SC decoder. The BP decoder of polar codes is a soft-in/soft-out decoder, thus, it is very suitable for decoding concatenated coding schemes (e.g., see [15]-[17]) and suitable for iterative detection and decoding. In this work, we mainly focus on BP decoding of polar codes when applied to iterative detection and decoding.

C. BP Decoding and Factor Graphs

Polar codes can be visualized as a code on a graph [18] and, thus, iterative BP decoding (i.e., sum product algorithm (SPA)) can be used to decode polar codes [5]. The Forney-style factor graph of polar codes (shown in Fig. 2) consists of $\log_2(N)$ stages. Each stage contains $\frac{N}{2^k}$ processing elements (PEs) (shown in Fig. 2b). The channel output in terms of log-likelihood ratio (LLR) $L_{ch}$ is fed into the right-most stage of the factor graph. The code structure is defined by the left-most stage of the factor graph or, in other words, the frozen and non-frozen bit-channel positions. Then, LLR-based messages propagate over the edges of the factor graph from right-to-left (L-messages) and then from left-to-right (R-messages), until a maximum number of iterations $N_{it,max}$ is reached or until an early stopping condition is satisfied [19]. Throughout this work, we use a $G$-matrix-based early stopping condition which terminates the iterations when $\hat{x}$ is equal to $\hat{u} \cdot G$ where $G$ is the $k \times N$ polar code generator matrix. A codeword estimate $\hat{x}$ and the information bits estimate $\hat{u}$ are the outputs from the BP decoder after performing a hard-decision operation on the LLR messages in the right-most and the left-most stages, respectively.

The $L$- and $R$-messages are updated in each PE as follows:
\[ R_{\text{out},1} = f(R_{\text{in},1}, L_{\text{in},2} + R_{\text{in},2}) \]
\[ R_{\text{out},2} = f(R_{\text{in},1}, L_{\text{in},1} + R_{\text{in},2}) \]
\[ L_{\text{out},1} = f(L_{\text{in},1}, L_{\text{in},2} + R_{\text{in},2}) \]
\[ L_{\text{out},2} = f(L_{\text{in},1}, L_{\text{in},1} + L_{\text{in},2}) \]

where \( f(L_1, L_2) = L_1 \boxplus L_2 \) is commonly referred to as boxplus operator, which can be expressed as
\[
f(x, y) = x \boxplus y = \log \frac{1 + e^{x+y}}{e^x + e^y}.
\]

III. BP-BASED MULTI-USER DECODING AND DETECTION OF POLAR CODES

Similar to [8], the proposed decoder setup is composed of several blocks, namely:

1) a sub-optimal soft interference cancellation (SoIC)-based MUD block as mentioned before,
2) a repetition (REP) decoder,
3) a polar iterative decoder (i.e., BP decoder).

A. General scheme

Assume that \( y \) is the received noisy channel observation of superimposed symbols \( x_i \). As depicted in Fig. 1, the proposed decoder setup processes \( y \) as follows:

1) SoIC-MUD
   - Optimally: MUD would compute a posteriori probability (APP) per bit with complexity \( \mathcal{O}(M^{K_a}) \), where \( M \) is the number of constellation symbols per user
   - Here: a sub-optimal soft interference cancellation based low complexity MUD is implemented with complexity \( \mathcal{O}(M \cdot K_a) \)
     - \( \hat{x}_i \): feedback from single user decoder
     - \( y_j \): forward message after SoIC
     \[
y_j = y - \sum_{i \neq j} \hat{x}_i
\]

For more details, we refer the interested reader to [8].

2) Demapping and De-scrambling: where log-likelihood-ratio (LLR) of each bit is computed by the (soft) demapper while treating the residual interference as noise; followed by user-specific phase scrambling.

3) Deinterleaving: according to each user’s specific interleave states. Accordingly, the a priori knowledge of the REP decoder comes from the extrinsic message from MUD.

4) REP decoding: of code rate \( R_e = \frac{1}{d_t} \) in case of repetition (discussed later), which enhances the signal-to-interference-plus-noise ratio (SINR) for further outer decoding. The extrinsic information of the \( j \)th information symbol of the repetition code is the summation of the LLRs of all corresponding repeated symbols.

5) BP decoding: run a few \( N_{\text{BP}} \) decoder-internal BP iterations (e.g., 1 or 2 iterations) to get an enhanced soft estimate of a codeword. As discussed later, more (sufficient) BP iterations are required in case of resetting the BP graph’s internal memory after each BP decoding step.

6) Repetition
7) Interleaving
8) Scrambling and remapping
9) Repeating previous steps for \( N_{\text{MUD}} \) times, and, thus, denoted as: \( (N_{\text{BP}} \times N_{\text{MUD}}) \).

Each individual BP decoder for a single user generates an improved soft-information output, which is then used by other user detectors to further de-noise the soft input to their BP decoders. The decoder setup is illustrated in Fig. 1.

B. Internal memory effects of polar BP factor graph

Internal LLR values of a BP factor graph instance, namely \( L_{i,j} \) and \( R_{i,j} \), are equivalently noted as factor graph internal states. These states form the so-called “internal memory” of a graph which affects the decoding convergence to a specific codeword. To further illustrate that, two BP decoding setups are compared in Fig. 3:

1) A BP decoding graph that clears the graph internal states before starting to decode a new codeword (i.e., resets the factor graph).
2) BP decoding graph where the internal memory of it persists (i.e., internal LLR values from previous codeword are not removed while decoding a new codeword).

As can be seen in Fig. 3, the graph internal memory initialization plays a significant role in the convergence behavior of a specific decoding instance. This has a strong impact on the multi user turbo decoder setup. After each MUD iteration, the BP factor graph is already at some state and, thus, the soft information is strongly affected by the previously acquired graph state. Thus, one solution might be as follows:

1) run BP for a sufficient number of iterations, instead of only a few iterations
2) pass information to MUD
3) reset BP factor graph (resetFG), i.e., “start over” whenever MUD output is updated.

The resultant new decoding setup enjoys an improved error-rate performance when compared to the previously mentioned setups, as depicted in Fig. 4 and Fig. 5.
IV. RESULTS

Throughout this work, all polar BP curves correspond to a polar code of length $N = 512$ designed with the 5G bit-channel ordering [3], whereas the curves from [12] are optimized for their respective scheme (i.e., different polar code or $A_r$-vector). We show results for two specific scenarios while benchmarking against the iterative SCL and JSC results of [12].

A. 2-User GMAC

Here, codewords from two active users are superimposed, each with a rate of $R_e = \frac{1}{2}$ and $d_e = 1$ (no repetition), thus, overall information rate $R_{sum} = \frac{1}{2}$. The error-rate simulation results for this setup are shown in Fig. 4.

B. 4-User GMAC

Here, codewords from four active users are superimposed with $R_{sum} = 1$ fixed. For the case of $R_e = \frac{1}{4}$ with $d_e = 1$ (no repetition) and $R_e = \frac{1}{2}$ with $d_e = 2$ (two-fold repetition). The error-rate simulation results for this setup are shown in Fig. 5.

Combined with repetition coding, the resultant scheme can be of significantly superior performance under joint polar BP-REP-SoIC decoding compared to that under JSC decoding introduced in [12]. It is worth-mentioning that scaling up the setup for (much) more users is straightforward.

V. THE REPETITION CODING DILEMMA

It is a well-known fact that repetition coding has no coding gain for the single-user case. However, it seems as if repetition codes are very efficient for interference cancellation as pointed out in [8]. As depicted in Fig. 5, a joint BP-based IDD receiver can benefit from an inner repetition code of repetition factor $d_e = 2$ when compared to a setup that does not involve a repetition code. The resultant code/receiver setup outperforms all of the previously mentioned coding/joint-IDD setups, as can be observed in Fig. 5. Here, we further explore this for the case of $K_a = 4$ active users.

A polar code of length $N$ concatenated with an inner repetition code of repetition factor $d_e$ can be viewed as another equivalent polar code of length $N \cdot d_e$, as depicted in Fig. 6. As can be seen, the equivalent polar code in Fig. 6b is constructed by appending $N \cdot (d_e - 1)$ frozen bits to the original polar code in Fig. 6a. The equivalent polar code results in the same error-rate performance when compared to the original code setup, as depicted in Fig. 8 (i.e., compare - and - ).

Based on that, one might deduce that the effect of concatenating an inner repetition code, with the polar code is equivalent to a design enhancement for polar codes for the multi user setup (i.e., a better $A_r$-vector tailored to the multi-user scenario). The performance gain with repetition is not merely a result of increasing the codelength by repetition, as can be observed from Fig. 8 (i.e., compare - and - ).

Fig. 4: BLER performance for codelength $N = 512$ and $R_{sum} = 0.5$ bits per channel use with $K_a = 2$ users over an AWGN channel.

Fig. 5: BLER performance for codelength $N = 512$ and $R_{sum} = 1$ bits per channel use with $K_a = 4$ users over an AWGN channel.
VI. CONCLUSION

In this paper, we proposed the usage of BP decoders for iterative detection and decoding of polar codes in a multi-user scenario. Combined with repetition coding, the resulting scheme can compete with, and even outperform, JSC based IDD [12] for two and four users, while the inherent parallel nature of polar BP decoding may lead to more desirable hardware implementations. Further analysis is required to investigate the case for a massive number of users.

We also shed some light on the gains by the usage of a concatenated repetition code and verified its effectiveness in Monte-Carlo simulations, a theoretical explanation remains, however, an open problem. We concluded that this repetition code corresponds to a different selection of frozen bit-positions (different A-vector) in the polar code design, suggesting that the multi-user detection can benefit from a different polar code design than the single-user case. Finding such an optimized polar code construction is an interesting open future work and an important step towards efficient deployment of multi-user polar coding with iterative detection and decoding.

REFERENCES

[1] E. Arıkan, “Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels,” IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
[2] M. C. Coskun, G. Durisi, T. Jerkovits, G. Liva, W. E. Ryan, B. Stein, and F. Steiner, “Efficient Error-Correcting Codes in the Short Blocklength Regime,” Physical Communication, vol. 34, pp. 66–79, Jun. 2019.
[3] “Technical Specification Group Radio Access Network,” 3GPP, 2018, TS 38.212 V15.1.1. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/38_series/38.212/
[4] I. Tal and A. Vardy, “List Decoding of Polar Codes,” IEEE Trans. Inf.
Theory, vol. 61, no. 5, pp. 2213–2226, May 2015.
[5] E. Arik, “Polar Codes: A Pipelined Implementation,” Proc. 4th ISBC,
p. 11–14, 2010.
[6] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, “Belief
Propagation List Decoding of Polar Codes,” IEEE Commun.
Lett., vol. 22, no. 8, pp. 1536–1539, Aug. 2018.
[7] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, “A Survey
of Non-Orthogonal Multiple Access for 5G,” IEEE Commun. Surveys
& Tutorials, vol. 20, no. 3, pp. 2294–2323, 2018.
[8] X. Wang, S. Cammerer, and S. ten Brink, “Near-Capacity Detection and
Decoding: Code Design for Dynamic User Loads in Gaussian Multiple
Access Channels,” IEEE Trans. Commun., vol. 67, no. 11, pp. 7417–
7430, 2019.
[9] E. Şaşoğlu, E. Telatar, and E. M. Yeh, “Polar Codes for the Two-User
Multiple-Access Channel,” IEEE Trans. Inf. Theory, vol. 59, no. 10, pp.
6583–6592, 2013.
[10] S. Onay, “Successive Cancellation Decoding of Polar Codes for the
Two-User Binary-Input MAC,” in IEEE Inter. Symp. Inf. Theory (ISIT),
2013, pp. 1122–1126.
[11] E. Abbe and E. Telatar, “Polar Codes for the m-User Multiple Access
Channel,” IEEE Trans. Inf. Theory, vol. 58, no. 8, pp. 5437–5448, 2012.
[12] E. Marshakov, G. Balitskiy, K. Andreev, and A. Frolov, “A Polar Code
Based Unsourced Random Access for the Gaussian MAC,” in IEEE 90th
Vehicular Technology Conference (VTC2019-Fall), 2019, pp. 1–5.
[13] Li Ping, Lihai Liu, Keying Wu, and W. K. Leung, “Interleave Division
Multiple-Access,” IEEE Trans. on Wireless Commun., vol. 5, no. 4, pp.
938–947, 2006.
[14] G. Song and J. Cheng, “Distance Enumerator Analysis for Interleave-
Division Multi-User Codes,” IEEE Trans. Inf. Theory, vol. 62, no. 7, pp.
4039–4053, 2016.
[15] J. Guo, M. Qin, A. G. i Fabregas, and P. H. Siegel, “Enhanced Belief
Propagation Decoding of Polar Codes through Concatenation,” in IEEE
Inter. Symp. Inf. Theory (ISIT), Jun. 2014, pp. 2987–2991.
[16] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, “Improving
Belief Propagation Decoding of Polar Codes Using Scattered EXIT
Charts,” in IEEE Inf. Theory Workshop (ITW), Sep. 2016, pp. 91–95.
[17] ——, “Flexible Length Polar Codes through Graph Based Augmenta-
tion,” in IEEE Inter. ITG Conf. on Syst., Commun. and Coding (SCC),
Feb. 2017.
[18] G. D. Forney, “Codes on Graphs: Normal Realizations,” IEEE Trans.
Inf. Theory, vol. 47, no. 2, pp. 520–548, Feb. 2001.
[19] B. Yuan and K. K. Parhi, “Early Stopping Criteria for Energy-Efficient
Low-Latency Belief-Propagation Polar Code Decoders,” IEEE Trans.
Sig. Process., vol. 62, no. 24, pp. 6496–6506, Dec. 2014.
[20] Y. Polyanskiy, “A Perspective on Massive Random-Access,” in IEEE
Inter. Symp. Inf. Theory (ISIT), 2017, pp. 2523–2527.
[21] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, “Decoder-
Tailored Polar Code Design Using the Genetic Algorithm,” IEEE Trans.
Commun., vol. 67, no. 7, pp. 4521–4534, Jul. 2019.