An Efficient Construction of Rate-Compatible Punctured Polar (RCPP) Codes Using Hierarchical Puncturing

Song-Nam Hong
Ajou University, Suwon, Korea,
email: snhong@ajou.ac.kr

Abstract

In this paper, we present an efficient method to construct a good rate-compatible punctured polar (RCPP) code. One of the major challenges on the construction of a RCPP code is to design a common information set which is good for all the codes in the family. In the proposed construction, a common information set is simply optimized for the highest-rate punctured polar code in the family and then, this set is updated for each other code by satisfying the condition that information bits are unchanged during retransmissions. This is enabled by presenting a novel hierarchical puncturing and information-copy technique. To be specific, some information bits are copied to frozen-bit channels, which yields an information-dependent frozen vector. Then, the updated information sets are obtained by appropriately combining the common information set and an information-dependent frozen vector. Moreover, the impact of “unknown” frozen bits are resolved using the proposed hierarchical puncturing. Simulation results demonstrate that the proposed RCPP code attains a significant performance gain (about 2dB) over a benchmark RCPP code where both codes use the same puncturing patterns but the latter uses the conventional all-zero frozen vector. Therefore, the proposed method would be crucial to construct a good RCPP code.

Index Terms

Polar code, rate-compatible code, puncturing, incremental redundancy, HARQ-IR.
I. INTRODUCTION

Polar codes, proposed by Arikan [1], achieve the symmetric capacity of binary-input discrete memoryless channels (BI-DMCs) under a low-complexity successive cancellation (SC) decoder. The finite-length performances of polar codes can be enhanced by using list decoder that enables polar codes to approach the performance of optimal maximum-likelihood (ML) decoder [2]. It was further shown in [2] that a polar code concatenated with a simple CRC outperforms well-optimized LDPC and Turbo codes especially for short lengths. Due to their good performance and low-complexity, polar codes are currently considered for possible deployment in future wireless communication systems (i.e., 5G cellular systems).

Wireless broadband systems (e.g., 4G LTE and 5G) operate in the presence of time-varying channels, thus requiring flexible and adaptive transmission techniques. For these systems, hybrid automatic repeat request based on incremental redundancy (HARQ-IR) schemes are widely used where parity bits for re-transmission are chosen in an incremental fashion according to a certain rate requirement. They are enabled by the use of a rate-compatible (RC) code that consists of a family of codes to support various rates. For the RC code, it should be ensured that the set of parity bits of a higher-rate code is a subset of the set of parity bits of a lower-rate code, which is called rate-compatibility constraint. This is able to allow the receiver that fails to decode at a particular rate, to request only additional parity bits from the transmitter. In this reason, there have been extensive researches on the design of RC Turbo and LDPC codes (see [3], [4], [5] and the references therein).

Very recently, a capacity-achieving RC polar code, called parallel concatenated polar (PCP) code, was presented in [6]. The main idea is to employ a capacity-achieving (punctured) polar code for every transmission, by satisfying the rate-compatibility constraint. As an independent work, a similar method was proposed in [7] by the name of incremental freezing. Although they can achieve the optimal performance for extremely large lengths, it does not directly imply that they yield good performances for practical lengths. Especially when incremental rate is small, they may not perform well as the length of constituent (punctured) polar code is too small. An alternative approach to design a RC polar code by performing successively puncturing from a mother polar code, which is called RC punctured polar (RCP) code. On the construction of a RCP code, it is required to jointly optimize rate-compatible puncturing patterns and a common information set that is good for all the codes in the family. Unfortunately, this optimization is not tractable due to its extensive complexity. Instead, numerous heuristic methods were presented to generate a puncturing pattern and an associated information set [8], [9], [10], [11]. In [12], a practical puncturing pattern, named quasi-uniform puncturing (QUP), was presented and shown to yield
an attractive performance but its extension to rate-compatible puncturing is not straightforward due to
the design of a good common information set. Also, an efficient search algorithm to jointly optimize a
puncturing pattern and an information set was developed and shown to outperform LDPC codes \cite{13}. However, it requires a prohibitive complexity to construct rate-compatible puncturing patterns. Therefore, it is still an open problem to construct a good RCPP code with a manageable complexity. This is the motivation of our work.

As noticed before, it is too complex to perform the joint optimization of rate-compatible puncturing
patterns and a common information set. Instead, we in this paper focus on the efficient construction of
a good RCPP code. In fact, there are good rate-compatible puncturing patterns like QUP in \cite{12} as long
as it is possible to employ an optimized information set for each puncturing pattern. Unfortunately, it
is not possible for a RCPP code since information bits should not be changed during retransmissions,
i.e., the corresponding common information set should be used for all the codes. Motivated by this, we
present a novel technique to produce virtual information sets from the common one, each of which is
an optimized information set for the corresponding code in the family.

The contributions of this paper are summarized as follows.

- We present a novel hierarchical puncturing having a special property such that “unknown” frozen
  bits can be allocated to some frozen-bit channels carefully chosen according to a puncturing pattern,
  without affecting the performance of the resulting punctured polar code. Furthermore, we identify
  a class of hierarchical puncturing patterns to satisfy the rate-compatible constraint. For example, it
  includes a successive puncturing, a quasi-uniform puncturing (QUP), and so on.

- We propose an information-copy technique which repeats some part of information bits to frozen-
  bit channels as well as information-bit channels, yielding an information-dependent frozen vector.
Here, the locations of information bits and frozen-bit channels are determined as a function of rate-
compatible puncturing patterns and a common information set. This technique will be used to update
an information set.

- Based on hierarchical puncturing and information-copy technique, we develop an efficient systematic
  method to construct a good RCPP code. In the proposed method, a common information set is simply
  optimized for the highest-rate code in the family and then, this set is updated to produce virtual
information sets suitable for the other codes in the family, by properly combining the common
information set and the information-dependent frozen vector. It is remarkable that the impact of
"unknown" (information-dependent) frozen bits is resolved due to the use of hierarchical puncturing.

- Simulation results demonstrate that the proposed RCPP code attains a significant performance gain
(about 2dB) over a benchmark RCPP code where both codes employ the same QUP but the latter uses the conventional all-zero frozen vector. Therefore, the proposed method be crucial to construct a good RCPP code.

The outline of this paper is as follows. In Section II we provide some useful notations and definitions to be used throughout the paper. In Section III we present a novel hierarchical puncturing and derive their special properties. In Section IV we develop an efficient method to construct a good RCPP code based on hierarchical puncturing and information-copy technique. Section V concludes the paper.

II. PRELIMINARIES

In this section we provide some useful notations and definitions that will be used in the sequel.

A. Notation

A polar code of length \( N = 2^n \) is considered, in which the synthesized (or polarized) channels are indexed by \( 0, 1, \ldots, N-1 \). Let \( \mathcal{A} \subseteq \{0, \ldots, N-1\} \) be the information-bit set which contains all the indices of the synthesized channels to carry information bits. Accordingly, its complement set \( \mathcal{A}^c \) represents the frozen-bit set that contains all the indices of frozen-bit channels. Let \( \mathbf{G}_N = \mathbf{G}_{2^n} \) be the rate-one generator matrix of all length-\( N \) polar codes, where \( \mathbf{G}_2 \) denotes the 2-by-2 Arikan’s Kernel \(^1\) as

\[
\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.
\]

Then, a length-\( N \) polar code is specified by its information-bit set \( \mathcal{A} \).

Given index subsets \( \mathcal{B}, \mathcal{D} \subseteq \{1, \ldots, N\} \), let \( \mathbf{G}_N(\mathcal{B}, \mathcal{D}) \) denote the submatrix of \( \mathbf{G}_N \) obtained by selecting the rows and columns whose indices belong to \( \mathcal{B} \) and \( \mathcal{D} \), respectively. Define a function \( g(\ell) : \{0, \ldots, N-1\} \rightarrow \{0,1\}^n \) which maps \( \ell \) onto a binary expansion as

\[
g(\ell) = (b_\ell^0, \ldots, b_\ell^n),
\]

such that \( \ell = \sum_{i=1}^n b_\ell^i 2^i \). We let \( w_\mathcal{H}(\mathbf{b}) \) denote the number of non-zero elements in a vector \( \mathbf{b} \) (called Hamming weight). Given \( \mathbf{u}_N = (u_0, \ldots, u_{N-1}) \) and \( \mathcal{A} \subset \{0, \ldots, N-1\} \), we write \( \mathbf{u}_\mathcal{A} \) to represent the subvector \( (a_i : i \in \mathcal{A}) \).

B. Punctured Polar Codes

We define a punctured polar code and its synthesized channels. Let \( \mathbf{u}_N = (u_0, \ldots, u_{N-1}) \) and \( \mathbf{x}_N = (x_0, \ldots, x_{N-1}) \) be respectively the input and output vectors of a length-\( N \) polar code. As shown in Fig. I

---

\(^1\) Arikan's Kernel

---
the encoding of the polar code is given by

\[ u_N G_N = x_N. \]

Note that for the simplicity of explanation, we assume that the bit-reverse permutation, denoted by \( \psi(\cdot) \) is applied to the decoding part, instead of the encoding part as in [1]. Namely, the SC decoding successively decodes the \( \hat{u}_{\psi(i)} \) for \( i = 0, ..., N - 1 \) in that order. In the example of \( N = 8 \), the SC decoding order is given as

\[ \hat{u}_0 \rightarrow \hat{u}_4 \rightarrow \hat{u}_2 \rightarrow \hat{u}_6 \rightarrow \hat{u}_1 \rightarrow \hat{u}_3 \rightarrow \hat{u}_7. \]

A punctured polar code of a length \( N_p < N \) is constructed by eliminating the \( N - N_p \) coded bits. Formally, it is described by the “mother” polar code of the length \( N \) and a binary vector (called puncturing pattern) \( p_N = (p_0, \ldots, p_{N-1}) \in \{0, 1\}^N \) such that \( w_H(p_N) = N_p \). Here, \( p_i = 0 \) indicates that the \( i \)-th coded bit (e.g., \( x_i \)) is punctured and thus not transmitted. For the simplicity, we will drop the index \( N \) in \( p_N \) as long as it is identified in the context. Given \( p \), we define the zero-location set which contains the locations of punctured bits as

\[ B_p \triangleq \{ i \in \{0, \ldots, N-1\} : p_i = 0 \}. \]
Fig. 2. The punctured polar code of $N_p = 4$ where the information set $A = \{5, 7\}$ and the puncturing pattern $\mathbf{p}^{(3,2,1)}_4 = (0, 1, 0, 1, 0, 1, 0, 1)$ (called QUP). Also, the input and output vectors are respectively $\mathbf{u}_4 = (u_1, u_3, u_5, u_7)$ and $\mathbf{x}_4 = (x_1, x_3, x_5, x_7)$.

Note that a puncturing pattern can be specified by either a binary vector or a zero-location set. The corresponding unpunctured coded bits are denoted by $\mathbf{x}_{N_p} = (x_i : i \in B_p^c)$ and accordingly, the $N_p$ channel observations are denoted by $\mathbf{y}_{N_p} = (y_i : i \in B_p^c)$. The notion of synthesized channels in polar codes can be extended into punctured polar codes in a straightforward manner as follows. Given a channel $W$, a length-$N$ polar code, and an associated puncturing pattern $\mathbf{p}$, the transition probability of the $i$-th synthesized channel of the resulting punctured polar code is defined as

$$W^{(i)}(y_{N_p}, \mathbf{u}_{i-1}, \mathbf{p}|u_i) = \frac{1}{2^{N-1}} \sum_{u_{i+1}, \ldots, u_N} \sum_{y_N \in \pi_{\mathbf{p}}(\{y_{N_p}\})} W^N(y_N|u_N \mathbf{G_N}),$$

(6)

where $\pi_{\mathbf{p}}(S) \triangleq \{y_N \in \mathcal{Y}^N : (y_i : i \in B_p^c) \in S\}$ and the channel transition probabilities are

$$W^N(y_N|\mathbf{x}_N) = \prod_{i \in [1:N]} W(y_i|x_i),$$

(7)

where $W(\cdot|\cdot)$ denotes the channel transition probability of the underlying BI-DMC. Throughout the paper, we let $W^{(i)}_p$ denote the $i$-th synthesized channel with the transition probability in (6), and let $I(W^{(i)}_p)$ denote the corresponding symmetric capacity. Also, a punctured polar code is defined as $\mathcal{C}(\mathbf{G_N}, \mathbf{u}_{A^c}, A, \mathbf{p}^{(i)})$ where $\mathbf{u}_{A^c}$ represents a frozen-bit vector. The encoder structure of a punctured polar code is illustrated in Fig. 2.

Given a puncturing pattern $\mathbf{p}$, we define the zero-capacity set which contains the zero-capacity syn-
thesized channels as
\[
D^W_p \triangleq \left\{ i \in \{0, ..., N - 1 \} : I(W^{(i)}_p) = 0 \right\}
\] (8)

In particular when \( W \) is a perfect channel (i.e., noiseless deterministic channel), the above set is denoted by \( D_p \), which satisfies that for any channel \( W \),
\[
D_p \subseteq D^W_p,
\] (9)
due to the nesting property of synthesized channels [14]. Furthermore, it was shown in [15] that
\[
|B_p| = |D_p| = N - w_H(p),
\] (10)
namely, the number of zero-capacity synthesized channels is equal to that of punctured coded bits.
Thus, all the synthesized channels whose indices belong to \( D_p \) should be frozen-bit channels. Given the symmetric capacities of the synthesized channels \( \{I(W^{(i)}_p) : i = 0, ..., N - 1\} \), we let
\[
\mathcal{L} = \max\text{-}\text{ind}^{(t)} \left\{ I(W^{(i)}_p) : i = 0, 1, ..., N - 1 \right\}
\] (11)
\[
\mathcal{S} = \min\text{-}\text{ind}^{(t)} \left\{ I(W^{(i)}_p) : i = 0, 1, ..., N - 1 \right\},
\] (12)
where \( \mathcal{L} \) and \( \mathcal{S} \) contain the indices corresponding to the \( t \) largest and smallest values in \( \{I(W^{(i)}_p)\} \), respectively.

III. A HIERARCHICAL PUNCTURING

We present a novel hierarchical puncturing, which will be used as a key technology on the construction of the proposed RCPP codes in Section IV. As noticed in Section II-B, all the synthesized channels associated with the zero-capacity set \( D_p \) should be frozen-bit channels, i.e.,
\[
u_i = 0 \text{ for } i \in D_p,
\] (13)
Equivalently, we have that \( A \cap D_p = \phi \). Otherwise, the corresponding punctured polar code surely leads to a frame (or block) error. Thus, \( D_p \) should be identified to design a good information set \( A \). In [15], it was shown that there exist a class of puncturing patterns to satisfy \( D_p = B_p \), which are referred to as reciprocal puncturing patterns. Also, their sufficient and necessary conditions are provided as follows:

**Theorem 1 ([15]):** A puncturing pattern \( \mathbf{p} \) is reciprocal if and only if the following properties are satisfied:

- **zero-inclusion:** \( 0 \in B_p \)
- **one-covering:** if \( i \in B_p \) and \( i \geq 1 \) \( j \), then \( j \in B_p \),
where \( i \succeq j \) means that for every digit of ‘1’ in the binary representation of index \( j \), the corresponding digit in the index \( i \) should also be ‘1’ and \( i \succeq 0 \) for every \( i > 0 \).

It is remarkable that for such puncturing patterns, we can straightforwardly identify the zero-capacity set \( \mathcal{D}_p \) from the puncturing pattern \( p \) (equivalently, \( B_p \)), which makes it much easier to design a good information set \( A \) for the punctured polar code.

**Example 1:** For a reciprocal puncturing pattern \( p = (p_0, \ldots, p_{15}) \), suppose that the coded bit \( x_7 \) is punctured (i.e., \( p_7 = 0 \)). Since \( g(7) = (0, 1, 1, 1) \), the one-covering property in Theorem 1 implies that the coded bits corresponding to the following locations should be also punctured:

- **Weight-2 locations:** \( g^{-1}((0, 0, 1, 1)), g^{-1}((0, 1, 0, 1)), g^{-1}((0, 1, 1, 0)) \)
- **Weight-1 locations:** \( g^{-1}((0, 0, 0, 1)), g^{-1}((0, 0, 1, 0)), g^{-1}((0, 1, 0, 0)) \)

Namely, we know that \( p_1 = p_2 = p_3 = p_4 = p_5 = p_6 = 0 \).

**A. Properties of reciprocal puncturing**

We derive some important properties of reciprocal puncturing patterns. Let \( \Pi_n \) denote the set of all permutations of \( (1, 2, \ldots, n) \) with \( |\Pi_n| = n! \). In the example of \( n = 3 \), we have:

\[
\Pi_3 = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}. \quad (14)
\]

Given \( p \) and \( \sigma \in \Pi_n \), we define a permuted puncturing pattern \( p^\sigma \) by describing its zero-location set as

\[
\mathcal{B}_{p^\sigma} = \left\{ g^{-1}\left( (b_{\sigma(n)}^i, \ldots, b_{\sigma(1)}^i) \right) : i \in \mathcal{B}_p \right\}, \quad (15)
\]
where \( g(i) = (b^i_1, ..., b^i_1) \). For example, if \( p = (0, 0, 0, 1, 1, 1) \) and \( \sigma = (3, 2, 1) \), then from Fig. 3 we have:

\[
p^{(3,2,1)} = (0, 1, 0, 1, 0, 1, 0, 1).
\] (16)

Using the above definition, we can get:

**Proposition 1:** If \( p \) is reciprocal, then \( p^\sigma \) is also reciprocal for any permutation \( \sigma \in \Pi_n \).

**Proof:** The proof follows the fact that a permutation \( \sigma \) in (15) definitely preserves the both zero-inclusion and one-covering properties in Theorem 1.

**Proposition 2:** For any reciprocal puncturing pattern \( p \), we have:

\[
G_N(B_p, B_c^p) = 0.
\] (17)

**Proof:** Recall that \( G_N(B_p, B_c^p) \) is the submatrix of \( G_N \) by taking the columns and rows whose indices respectively belong to \( B_p \) and \( B_c^p \). Suppose there exists a non-zero element in \( G_N(B_p, B_c^p) \), i.e., \( G_N(i, j) = 1 \) for \( i \in B_p \) and \( j \in B_c^p \). That is, \( u_i \) is added to \( u_j \) to generate the \( j \)-th coded bit \( x_j \), which shows that \( i \geq 1 \). Since \( p \) is reciprocal, it should be hold from Theorem 1 that any index \( t \) with \( i \geq 1 \) should be the element of \( B_p \). Accordingly, \( j \) should be the element of \( B_p \), which is the contradiction that \( j \in B_c^p \). Therefore, there should be no 1’s in \( G_N(B_p, B_c^p) \), i.e., \( G_N(B_p, B_c^p) = 0 \). This completes the proof.

We also provide a class of reciprocal puncturing patterns obtained from a successive puncturing and a permutation as follows:

**Definition 1:** Let \( \hat{P}_{N_p} \) denote the successive puncturing pattern which punctures the first \( N - N_p \) coded bits, namely, its zero-location set is given by

\[
B_{\hat{P}_{N_p}} = \{0, 1, ..., N - N_p - 1\}.
\] (18)

It is obviously reciprocal. Also, from Proposition 1 its permuted puncturing pattern \( \hat{P}_{N_p}^\sigma \) is also reciprocal for every \( \sigma \in \Pi_n \). Especially, \( \hat{P}_{N_p}^{(n,n-1,...,1)} \) is known as quasi-uniform puncturing (QUP) in [12] (see Fig 4).

**B. A hierarchical puncturing**

We define a hierarchical puncturing and derive its special properties.

**Definition 2:** A reciprocal puncturing pattern \( p \) with \( w_H(p) = 2^{\bar{n}} = \bar{N} \) for some \( \bar{n} < n \) is said to be hierarchical if

\[
u_N G_N(\{0, ..., N - 1\}, B_c^p) = (u_i, i \in B_c^p) G_N.
\] (19)

DRAFT
From Proposition 2, the condition (19) is equivalent to

\[ G_N(B_p^c, B_p^c) = G_{\bar{N}}. \]  

(20)

From now on, we will explain the two properties of hierarchical puncturing with a simple example. Consider the \( N = 8 \), \( \bar{N} = 5 \) and the reciprocal puncturing pattern \( p_4^{(3,2,1)} \) (e.g., QUP). In this case, we have the input-output relationship of the punctured polar code as

\[
(x_i, i \in B_{p_4}^{c(3,2,1)}) = u_N G_N \left( \{0, ..., 7\}, B_{p_4}^{c(3,2,1)} \right)
= (u_i, i \in B_{p_4}^{c(3,2,1)}) G_N(B_{p_4}^{c(3,2,1)}, B_{p_4}^{c(3,2,1)}) + (u_i, i \in B_{p_4}^{c(3,2,1)}) G_N(B_{p_4}^{c(3,2,1)}, B_{p_4}^{c(3,2,1)})
= (u_i, i \in B_{p_4}^{c(3,2,1)}) G_{\bar{N}},
\]

(21)

where the last equality follows the fact that \( G_N(B_{p_4}^{c(3,2,1)}, B_{p_4}^{c(3,2,1)}) = G_{\bar{N}} \) and \( G_N(B_{p_4}^{c(3,2,1)}, B_{p_4}^{c(3,2,1)}) = 0 \). Thus, from (20) in Definition 2, \( p_4^{(3,2,1)} \) is hierarchical. Also, from (13), we know that \((u_i, i \in B_{p_4}^{c(3,2,1)}) \) only carry the information bits. Let \( u_{\bar{N}} = (u_i, i \in B_{p_4}^{c(3,2,1)}) \) and \( x_{\bar{N}} = (x_i, i \in B_{p_4}^{c(3,2,1)}) \) (i.e., unpunctured coded bits). Then, from (21), we can create a length-\( \bar{N} \) polar coding system:

\[ x_{\bar{N}} = u_{\bar{N}} G_{\bar{N}}. \]

(22)

From the (21) and (22) in the above example, we have the two important observations as

- **Observation 1**: Information bits can be decoded using the length-\( \bar{N} \) polar decoder with the (unpunctured) observation \( y_{\bar{N}} \) (see Fig. 4), instead of using the original polar decoder;
• **Observation 2:** Assigning unknown values to the synthesized channels corresponding to $B_{\hat{p}_4^{(3,2,1)}}$ does not impact on the performance of the length-$\bar{N}$ punctured polar code, since $G_N(B_{\hat{p}_4^{(3,2,1)}}, B_{\hat{c}_4^{(3,2,1)}}) = 0$ (see Proposition 2).

These observations will be exploited later on in Section IV.

For the rest of this section, we provide a class of hierarchical puncturing patterns. To explain them clearly, we start with the simple case of $N = 8$ and $\bar{N} = 4$, i.e., the half coded bits are punctured. From Fig. 1, we can see that the polar encoding consists of $\log N = 3$ levels. Also, we can easily see that $\hat{p}_4 = (0, 0, 0, 1, 1, 1, 1)$ (called successive puncturing) is hierarchical, since in this case we have

$$G_8(B_{\hat{c}_4^{c}}, B_{\hat{c}_4}) = G_4. \quad (23)$$

Also, we observe that its permuted puncturing pattern $\hat{p}_4^{(3,2,1)} = (0, 1, 0, 1, 0, 0, 0)$ is also hierarchical as illustrated in Fig. 2. Observing these two hierarchical puncturing patterns, we can identify that

- in the case of $\hat{p}_4$, $b_i^j = 0$ for all $i \in B_{\hat{p}_4}$
- in the case of $\hat{p}_4^{(3,2,1)}$, $b_i^j = 0$ for all $i \in B_{\hat{p}_4^{(3,2,1)}}$

where $g(i) = (b_i^3, b_i^2, b_i^1)$. To be specific, the 3rd and 1st levels in Fig. 1 are completely eliminated, respectively. Furthermore, the remaining parts in the both cases produce the structure of length-4 polar encoding. In other words, they satisfy the condition (20). By generalizing the above arguments, we have that when the half coded bits are punctured (e.g., $\bar{N} = N/2$), if $b_j^i = 0$ for all $i \in B_{\hat{p}_{\bar{N}/2}}$, then the $j$-th level in polar encoding structure is completely eliminated. Thus, it can satisfy the condition (20). Based on this, $\hat{p}_{\bar{N}/2}^{c}$ is hierarchical for any $\sigma \in \Pi_n$. This is generalized for any $\bar{N} = 2^n$ below:

**Theorem 2:** For any $\bar{N} = 2^n$ with $1 \leq \bar{n} < n$, a puncturing pattern $\hat{p}_{\bar{N}}^{c}$ in Definition 1 is hierarchical for any $\sigma \in \Pi_n$.

**Proof:** As explained in the above, the statement holds for $\bar{n} = n - 1$. Next, focusing on $\bar{n} = n - 2$, we consider the puncturing pattern $\hat{p}_{N/4}^{c}$ for some $\sigma \in \Pi_n$. In order to use the result of $\bar{n} = n - 1$, we first decompose the $B_{\hat{p}_{N/4}^{c}}$ into two disjoint subsets:

$$B_{\hat{p}_{N/4}^{c}} = B_{\hat{p}_{N/2}^{c}} \cup \left( B_{\hat{p}_{N/4}^{c}} \setminus B_{\hat{p}_{N/2}^{c}} \right). \quad (24)$$

Given $B_{\hat{p}_{N/2}^{c}}$, we can create the length-$N/2$ polarization structure with input vector $(u_i : i \in B_{\hat{p}_{N/2}^{c}})$ and output vector $(x_i : i \in B_{\hat{p}_{N/2}^{c}})$ (see Fig. 2) since $\hat{p}_{N/2}^{c}$ is hierarchical. Letting $N' = N/2$ and by re-indexing the input and output vectors, we can yield the length-$N'$ polar code with puncturing pattern $\hat{p}_{N'/2}^{\sigma'}$ where $\sigma'_i = \sigma_{i+1}$ for $i = 1, ..., n - 1$ (see Fig. 4). Then, $\hat{p}_{N'/2}^{\sigma'}$ is hierarchical with respect to the resulting length-$N'$ polar code. By combining the two stages, it is clear that $\hat{p}_{N/4}^{c}$ is hierarchical with
respect to the original length-$N$ polar code. In general for $n = n - \ell$ with an arbitrary $1 \leq \ell \leq n - 1$, we can prove the statement exactly following the above procedures with $\ell$ stages. This completes the proof.

IV. THE PROPOSED RCPP CODE

A RCPP code consists of a family of (punctured) polar codes for which the corresponding puncturing patterns satisfy the rate-compatible constraint. Also, since information bits cannot be changed during retransmissions in HARQ-IR, all the codes in the family should employ a common information set $A$. In general, $A$ is optimized by targeting at one of the codes in the family (e.g., mother polar code). Thus, this set cannot be good for the other codes in the family, thus being able to yield a performance loss especially when a rate-change is large.

In this section, we address the above problem by introducing the so-called information-copy technique based on hierarchical puncturing. The key idea of the proposed method is that some information bits, carefully chosen according to the rate-compatible puncturing patterns, are repeated (or copied) to frozen-bit channels as well as information-bit channels. That is, the proposed RCPP code employs an information-dependent (non-deterministic) frozen vector. Leveraging this, we produce virtual information sets which are good for the other codes in the family. In general, this approach is not available because ‘unknown” frozen bits can significantly degrade the performance. We completely resolve this problem using hierarchical puncturing (see Observation 2 in Section III-B). The detailed procedure to construct the proposed RCPP codes are provided as follows.

Suppose we construct a RCPP code to send $k$ information bits with various rates

$$r_1 = \frac{k}{N_1} < r_2 = \frac{k}{N_2} < \cdots < r_m = \frac{k}{N_m}. \quad (25)$$

First of all, we choose the “mother” code to support the above rates via rate-compatible puncturing as the polar code of the length $\tilde{N}_1 = 2^{n_1}$, where $n_1 = \lceil \log N_1 \rceil$. Also, a family of rate-compatible puncturing patterns are denoted by the length-$\tilde{N}_1$ binary vectors

$$p^{(1)}, p^{(2)}, p^{(3)}, \ldots, \text{and } p^{(m)}, \quad (26)$$

such that $w_H(p^{(i)}) = N_i$ for $i \in \{1, \ldots, m\}$, where each puncturing pattern $p^{(i)}$ generates the punctured polar code of rate $r_i$. Due to the rate-compatible constraint, they should satisfy the

\textbf{RC Condition:} $B_{p^{(1)}} \subset B_{p^{(2)}} \subset B_{p^{(3)}} \subset \cdots \subset B_{p^{(m)}}. \quad (27)$
In the proposed RCPP code, we employ the reciprocal puncturing in Definition 1 such as
\[ p(i) \triangleq p_{N_i}, \]  
(28)
for a fixed \( \sigma \in \Pi_{N_i} \) and for \( i \in \{1, ..., m\} \). It is noticeable that they are hierarchical when \( N_i \) has the form of power of 2 and also satisfy the rate-compatible condition (27). Also, we can describe such puncturing patterns in (28) using the seed sequence as follows:

**Definition 3:** Given \( \bar{N}_1 \) and \( \sigma \), we define the seed sequence by the following length-\( \bar{N}_1 \) binary vector:
\[ s_{\bar{N}_1}^\sigma \triangleq \left( g^{-1}(b_{\sigma(3)}^0, b_{\sigma(2)}^0, b_{\sigma(1)}^0), ..., g^{-1}(b_{\bar{N}_1-1}^{N_i-1}, b_{\bar{N}_1-1}^{N_i-1}) \right), \]
(29)
where \( (b_{i_1}, ..., b_{i_j}) \) represents the binary representation of ‘\( i \)’ for \( i = 0, ..., \bar{N}_1-1 \). From this, the puncturing pattern \( p(i) = p_{N_i}^\sigma \) in (26) is defined with its zero-location set as
\[ B_{p(i)} = \left\{ s_{\bar{N}_1}^\sigma (i) : i = 0, 1, ..., N_i \right\}. \]
(30)

**Example 2:** Consider the case of \( \bar{N}_1 = 8 \) and \( \sigma = (3, 2, 1) \) (e.g., QUP). From (29), we can obtain the length-\( \bar{N}_1 \) seed sequence as
\[ s_{8}^{(3,2,1)} = (0, 4, 2, 6, 1, 5, 3, 7). \]
(31)
Suppose \( N_1 = 7, N_2 = 5, \) and \( N_3 = 3 \). Then, from (30), we can easily obtain the corresponding puncturing patterns as
\[ B_{p(1)} = \{0\} \]
(32)
\[ B_{p(2)} = \{0, 4, 2\} \]
(33)
\[ B_{p(3)} = \{0, 4, 2, 6, 1\}. \]
(34)
Also, we can verify that they satisfy the RC condition (27).

In the proposed RCPP code, we optimize the common information set \( A \) by taking into account the puncturing pattern \( p^{(m)} \) (i.e., the highest punctured polar code in the family). Since \( p^{(m)} \) is reciprocal, \( A \) should satisfy the
\[ A \cap B_{p^{(m)}} = \emptyset, \]
(35)
namely, the information-bit channels are only located in \( B_{p^{(m)}}^c \). For the simplicity, the \( m \) number of codes in the family are denoted by \( \{C^{(1)}, ..., C^{(m)}\} \), where each \( C^{(i)} \) is the punctured polar code
\[ C^{(i)} \triangleq C(G_{\bar{N}_1}, u_{A^c}, A, p^{(i)}), \]
(36)
where a frozen-vector \( u_{A^c} \) will be defined later on.
Fig. 5. The punctured polar code of length \( N_p = 2 \), which is constructed using the puncturing pattern \( \hat{p}_p^{(3,2,1)} = (0, 0, 0, 0, 1, 0, 1) \). Recall that the decoding order of SC decoder is as follows: \( u_0 \rightarrow u_4 \rightarrow u_2 \rightarrow u_6 \rightarrow u_1 \rightarrow u_5 \rightarrow u_3 \rightarrow u_7 \).

From now on, we will explain how to produce virtual information sets suitable for the codes \( \mathcal{C}^{(i)} \), \( i = 1, ..., m - 1 \) (equivalently, to construct an information-dependent frozen vector). To explain the main idea clearly, we start with the simple case of \( k = 2 \) and

\[
   r_1 = \frac{2}{8} < r_2 = \frac{2}{5} < r_3 = \frac{2}{3}.
\]

From (28) and (33)-(34), we select the rate-compatible puncturing patterns such as

\[
   \mathbf{p}^{(1)} = 1 = (1, 1, 1, 1, 1, 1, 1, 1)
\]

\[
   \mathbf{p}^{(2)} = \hat{p}_5^{(3,2,1)} = (0, 0, 1, 0, 1, 0, 1, 1)
\]

\[
   \mathbf{p}^{(3)} = \hat{p}_3^{(3,2,1)} = (0, 0, 0, 1, 0, 1, 0, 1).
\]

Also, the common information set (which is optimized for \( \mathcal{C}^{(3)} \)) is obtained as

\[
   \mathcal{A} = \mathcal{A}^{(3)} = \{5, 7\},
\]

where the superscript 3 implies that \( \mathcal{A} \) is optimal with respect to \( \mathcal{C}^{(3)} \). Also, in this example, we noticed that \( \mathcal{A} = \mathcal{A}^{(3)} \) is not optimal for the code \( \mathcal{C}^{(2)} \) as \( \mathcal{A}^{(2)} = \{6, 7\} \neq \mathcal{A}^{(3)} \), i.e., \( I(W_p^{(6)}) > I(W_p^{(5)}) \) (see Fig. 5). Nevertheless, in conventional approaches, \( \mathcal{A} \) is used for all the codes in the family as the information set. In the proposed RCPP code, however, each code in the family can use its own optimized (virtual) information set. We show that in this example, the code \( \mathcal{C}^{(2)} \) can use \( \mathcal{A}^{(2)} = \{6, 7\} \) as its information set. Let \( \bar{n}_i = \lceil \log N_i \rceil \) and \( \bar{N}_i = 2^{\bar{n}_i} \) for \( i = 1, 2, 3 \). From Fig. 4, we observe that \( \mathcal{C}^{(3)} \) can be decoded using the length-4 polar code \( \mathcal{C}(G_4, \{3, 4\}, (0, 0), \mathbf{p} = (0, 1, 1, 1)) \), instead of...
using the mother polar code \( C(G_8, \{5, 7\}, u_{A^c}, 1) \). This is enable because \( p_{3}^{(3,2,1)} \) is hierarchical (see Observation 1 in Section III-B). Also, it is remarkable that the frozen-bit channels \( B_p, \{0, 2, 4, 6\} \) are not associated with this decoding and thus, the allocations of “unknown” frozen bits to those frozen-bit channels do not degrade the performance of \( C^{(3)} \) (see Observation 2 in Section III-B). Thus, we can copy the information-bit \( u_5 \) to the frozen-bit channel 6 (i.e., \( u_6 = u_5 \)), without affecting the performance of \( C^{(3)} \). For the decoding of \( C^{(2)} \), \( u_5 \) can operate as a frozen bit as the copied one \( u_6 \) is decoded in advance and thus, \( A^{(2)} = \{6, 7\} \) is used as the information set. Next focusing on \( C^{(1)} \), we have \( \lceil \log N_1 \rceil = \lceil \log N_2 \rceil \) and hence, both \( C^{(1)} \) and \( C^{(2)} \) should be decoded with the same length polar code (e.g., \( C(G_8, \{5, 7\}, u_{A^c}, 1) \)). In this case, there is no way to eliminate “unknown” frozen bits for the decoding of \( C^{(2)} \) if an information bit is copied to yield an improved information set of \( C^{(1)} \). Namely, the information-copy technique cannot be applied. Consequently, in this example, the information-dependent frozen vector is obtained as \( u_{A^c} = (u_0, u_1, u_2, u_3, u_4, u_6, u_7) = (0, 0, 0, u_5, 0) \). Also, the (punctured) polar codes in the family are defined as

\[
C^{(i)} = C\left(G_8, \mathcal{A} = \{5, 7\}, u_{A^c} = (0, 0, 0, u_5, 0), p^{(i)}\right), i = 1, 2, 3.
\]

(42)

This shows that in the proposed RCPP code, there is a precoding to generate an information-dependent frozen vector of the mother polar code as shown in Fig. 6 where the \( |\mathcal{A}| \times N_1 \) precoding matrix \( P \) is determined as a function of \( \mathcal{A} \) and \( u_{A^c} \) (or \( A^{(3)} \) and \( A^{(2)} \)). In this example, the precoding is given by

\[
(m_0, m_1) \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = (u_0, u_1, \ldots, u_{N_1-1}), \quad \left(\begin{array}{c} m_0 \\ m_1 \end{array}\right) \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = (u_0, u_1, \ldots, u_{N_1-1}), \quad \left(\begin{array}{c} m_0 \\ m_1 \end{array}\right) \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\]

(43)

where \( (m_0, m_1) \) represents the two information bits, since \( P(\{0\}, \{5\}) = P(\{1\}, \{7\}) = 1 \) for \( A^{(3)} = \{5, 7\} \) and \( P(\{0\}, \{6\}) = 1 \) for \( A^{(2)} \setminus A^{(3)} = \{6\} \). The systematic algorithm to determine a precoding matrix \( P \) is provided in Algorithm 2.

From the above example, we can identify that the information-copy technique can be applied only when the following requirements hold:

- A rate-change is sufficiently large, i.e., \( r_j \) is changed into \( r_i \) with \( i < j \) and \( \lceil \log N_i \rceil > \lceil \log N_j \rceil \);  
- The indices of frozen-bit channels should the elements of \( (A^{(2)} \setminus A^{(3)}) \cap B_{p^{(3,2,1)}} \). Here, the intersection with \( B_{p^{(3,2,1)}} \) is required to ensure that the copied bits should not impact on the performance of \( C^{(3)} \);  
- To attain an actual gain of the information-copy technique, the copied information bit (in a frozen-bit channel) should be decoded earlier than the original one in an information-bit channel during the SC decoding.
Taking the above requirements into account, we develop the methods to construct the information sets $\mathcal{A}^{(i)}$s and an information-dependent frozen vector (equivalently, the $|\mathcal{A}| \times \tilde{N}_1$ precoding matrix $\mathbf{P}$) in Algorithms 1 and 2, respectively. Finally, for a given precoding matrix $\mathbf{P}$ and a permutation $\sigma$ (equivalently, seed sequence in Definition 3), the encoding structure of the proposed RCPP code is depicted in Fig. 6.

**Simulation results:** For the simulation, we constructed the proposed RCPP code to send 52 information bits via for different rates

$$r_1 = \frac{52}{256} < r_2 = \frac{52}{192} < r_3 = \frac{52}{128} < r_4 = \frac{52}{64}. \quad (44)$$

Also, we employed the rate-compatible puncturing patterns (e.g., QUP) as

$$\mathbf{p}^{(1)} = \mathbf{1}, \mathbf{p}^{(2)} = \mathbf{p}^{(8,7,...,1)}_{192},$$

$$\mathbf{p}^{(3)} = \mathbf{p}^{(8,7,...,1)}_{128}, \text{ and } \mathbf{p}^{(4)} = \mathbf{p}^{(8,7,...,1)}_{64}. \quad (45)$$

We optimized the common information set $\mathcal{A}$ by taking into account the puncturing pattern $\mathbf{p}^{(4)} = \mathbf{p}^{(8,7,...,1)}_{64}$. Using Algorithm 1, we obtained the $J = 3$ information sets $\mathcal{A}^{(1)}, \mathcal{A}^{(2)},$ and $\mathcal{A}^{(3)} = \mathcal{A}$. Also, using Algorithm 2, the precoding matrix $\mathbf{P}$ is constructed. In the decoding side, both $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(2)}$ are decoded using the length-256 polar decoder and the information set $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}$ is decoded using the length-128 polar decoder and the information set $\mathcal{A}^{(2)}$, and $\mathcal{C}^{(3)}$ is decoded using the length-64 polar decoder and the information set $\mathcal{A}^{(3)}$. As benchmark method, we employed the RCPP code with the same puncturing patterns in (45) while the common information set $\mathcal{A}$ is used for all the codes in the family. The list decoder with list-size 8 and 8-bit CRC are used. From Fig. 7 we observe that the proposed RCPP code significantly outperforms the benchmark code. As expected, the performance gain becomes
a larger as a rate is lower. Therefore, when a RCPP code should support a wide range of rates, the performance gain of the proposed method is much larger.

V. CONCLUSION

We presented novel reciprocal and hierarchical puncturing having special properties. In particular when a hierarchical puncturing is used, it is possible to use “unknown” frozen bits without degrading a performance. Leveraging such property of hierarchical puncturing and the so-called information-copy technique, we showed that each code in the family can employ an improved information set. This is obtained by appropriately combining the common information set and an information-dependent frozen vector. Via simulation results, we demonstrated that the proposed method yields a non-trivial performance gain mainly due to the use of improved information sets. Therefore, it would be a crucial technique to construct a good RCPP code efficiently. A necessary future work is to find a good hierarchical puncturing pattern suitable for the proposed RCPP code.
Algorithm 1 Improved information sets

Input:
- \( r_i = \frac{k}{N_i} \) for \( i \in \{1, \ldots, m\} \) with \( r_i < r_j \) if \( i < j \), and information set \( \mathcal{A} \).
- Bit-permutation parameter \( \sigma \in \Pi_{\bar{n}_i} \).

Output: The optimized information sets \( \mathcal{A}^{(j)} \) for \( j \in \{1, \ldots, J - 1\} \).

Algorithm:
- \( \bar{n}_i = \lceil \log N_i \rceil \) and \( \bar{N}_i = 2^{\bar{n}_i} \) for \( i \in \{1, \ldots, m\} \).
- Let \( J \) be the number of distinct values in \( \{\bar{N}_1, \ldots, \bar{N}_m\} \) and the corresponding values are denoted by \( \bar{L}_1 = \bar{N}_1 > \bar{L}_2 > \cdots > \bar{L}_J = \bar{N}_m \).
- \( \mathcal{A}^{(j)} = \mathcal{A} \) (\( \mathcal{A} \) is optimized for the highest-rate code in the family).
- For \( j = J - 1, \ldots, 1 \)
  1) Find a new information set \( \mathcal{A} \) as
     \[
     \mathcal{A} = \max \text{-} \text{ind}^{(k)} \left\{ I(W_{p^{(i)}}^i) : i = 0, \ldots, \bar{N}_1 - 1 \right\},
     \]
     where \( t^i = \min_{t} \{\bar{N}_t : \bar{N}_t = \bar{L}_j\} \). Also, let
     \[
     \mathcal{I}_1 = (\mathcal{A} \setminus \mathcal{A}^{(j+1)}) \cap B_{\bar{N}_{j+1}} \triangleq \{\ell_1, \ldots, \ell_{|\mathcal{I}|}\}
     \]
     \[
     \mathcal{I}_2 = \min \text{-} \text{ind}^{(|\mathcal{I}_1|)} \left\{ I(W_{p^{(i)}}^i) : i \in \mathcal{A}^{(j+1)} \right\},
     \]
     where \( \psi(\ell_1) < \cdots < \psi(\ell_{|\mathcal{I}|}) \).
  2) Define an information-copy set \( \mathcal{I}_c \) as
     - Initialization: \( \mathcal{I}_c = \emptyset \) and \( \mathcal{I}_d = \emptyset \).
     - For \( i = 1, \ldots, |\mathcal{I}| \)
       a) \( \mathcal{T} = \{q \in \mathcal{I}_2 \setminus \mathcal{I}_d : \psi(q) > \psi(\ell_i)\} \)
       b) If \( \mathcal{T} \neq \emptyset \), then
          \[
          \mathcal{I}_c = \mathcal{I}_c \cup \{\ell_i\} \quad \text{and} \quad \mathcal{I}_d = \mathcal{I}_d \cup \{q^*\},
          \]
          where \( q^* = \min_{q \in \mathcal{T}} \psi(q) \).
  3) Define \( \mathcal{A}^{(j)} \triangleq \mathcal{I}_c \cup (\mathcal{A}^{(j+1)} \setminus \mathcal{I}_d) \).
Algorithm 2 Precoding matrix for information-dependent frozen vector

Input:
- Information set $A^{(j)}$ for $j \in \{1, ..., J\}$ and a length $\bar{N}_1$.

Output: The $|A^{(J)}| \times \bar{N}_1$ precoding matrix $P$ where $P(i, j)$ denotes the $(i, j)$-th element of the $P$ for $i = 0, 1, ..., |A^{(J)}| - 1$ and $j = 0, 1, ..., \bar{N}_1 - 1$.

Initialization:
- $A^{(J)} = \{\ell_0, ..., \ell_{|A^{(J)}| - 1}\}$.
- Define a mapping $h: A^{(j)} \to \{0, 1, ..., |A^{(j)}| - 1\}$, i.e.,
  \[ h(\ell_j) = j \text{ for } j = 0, ..., |A^{(j)}| - 1. \]
- Let $A^{(j)} - A^{(j+1)} = \{\ell_1^{(j)}, ..., \ell_{d_j}^{(j)}\} \subset A^{(j)}$ for $j \in \{1, ..., J - 1\}$.
- $P = 0$.

Algorithm:
- Assign $P(\{h(\ell_j)\}, \{\ell_j\}) = 1$ for $j = 0, ..., |A^{(J)}| - 1$.
- For $j = J - 1, ..., 1$
  - Assign $P(\{h(i_t^{(j)})\}, \{i_t^{(j)}\}) = 1$ for $t \in [1 : d_j]$.

REFERENCES

[1] E. Arikan, “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] I. Tal and A. Vardy, “List decoding of polar codes,” IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2213-2226, May 2015.
[3] D. Rowitch and L. Milstein, “On the performance of hybrid FEC/ARQ systems using rate-compatible punctured turbo (RCPT) codes,” IEEE Trans. Commun., vol. 48, no. 6, pp. 948-959, Jun. 2000.
[4] M. El-Khamy, J. Hou and N. Blushan, “Design of rate-compatible structured LDPC codes for hybrid ARQ applications,” IEEE Jou. Sel. Areas Commun., vol. 27, pp. 965-973, Aug. 2009.
[5] T. 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.
[6] S.-N. Hong, I. Maric and D. Hui, “Capacity-achieving rate-compatible polar codes,” IEEE Trans. Inf. Theory, vol. 63, no. 12, pp. 7620-7632, Dec. 2017.
[7] B. Li, D. Tse, K. Chen and H. Shen, “Capacity-achieving rateless polar codes,” in Proc. IEEE Int. Inf. Theory (ISIT), Barcelona, Spain, Jul. 2016.
[8] A. Eslami and H. Pishro-Nik, “Practical approach to polar codes,” in Proc. IEEE Int. Symp. Inf. Th., pp. 16-20, St. Petersburg, Russia, Aug. 2011.
[9] K. Chen, K. Niu and J. Lin, “A hybrid ARQ scheme based on polar codes,” IEEE Commun. Letters, vol. 17, no.10, pp. 1996-1999, Oct. 2013.

[10] R. Wang and R. Liu, “A novel puncturing scheme for polar codes,” IEEE Commun. Letters, vol. 18, no. 12, pp. 2081-2084, Dec. 2014.

[11] D.-M. Shin, S.-C. Lim and K. Yang, “Design of rate-compatible polar codes based on the reduction of polarizing matrices,” IEEE Trans. Commun., vol. 61, no. 7, pp. 2593-2599, Jul. 2013.

[12] K. Niu, K. Chen and J.-R. Lin, “Beyond turbo codes: rate-compatible punctured polar codes,” in Proc. IEEE Int. Conf. Commun. (ICC), Budapest, Hungary, Jun. 2013.

[13] V. Miloslavskaya, “Shortened Polar Codes,” IEEE Trans. Inf. Theory, vol. 61, no. 9, pp. 4852-4865, Sep. 2015.

[14] S. Korada, “Polar codes for channel and source coding,” Ph.D. thesis, Ecole Polytechnique Federale de Lausanne (EPFL), 2009.

[15] S.-N. Hong, I. Maric and D. Hui, “On the analysis of puncturing for finite-length polar codes: boolean function approach,” to be submitted to IEEE Trans. Commun. Dec. 2017.